

Katrin Franke  
Klaus-Robert Müller  
Bertram Nickolay  
Ralf Schäfer (Eds.)

LNCS 4174

# Pattern Recognition

28th DAGM Symposium  
Berlin, Germany, September 2006  
Proceedings

 Springer

*Commenced Publication in 1973*

Founding and Former Series Editors:

Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

## Editorial Board

David Hutchison

*Lancaster University, UK*

Takeo Kanade

*Carnegie Mellon University, Pittsburgh, PA, USA*

Josef Kittler

*University of Surrey, Guildford, UK*

Jon M. Kleinberg

*Cornell University, Ithaca, NY, USA*

Friedemann Mattern

*ETH Zurich, Switzerland*

John C. Mitchell

*Stanford University, CA, USA*

Moni Naor

*Weizmann Institute of Science, Rehovot, Israel*

Oscar Nierstrasz

*University of Bern, Switzerland*

C. Pandu Rangan

*Indian Institute of Technology, Madras, India*

Bernhard Steffen

*University of Dortmund, Germany*

Madhu Sudan

*Massachusetts Institute of Technology, MA, USA*

Demetri Terzopoulos

*University of California, Los Angeles, CA, USA*

Doug Tygar

*University of California, Berkeley, CA, USA*

Moshe Y. Vardi

*Rice University, Houston, TX, USA*

Gerhard Weikum

*Max-Planck Institute of Computer Science, Saarbruecken, Germany*

Katrin Franke Klaus-Robert Müller  
Bertram Nickolay Ralf Schäfer (Eds.)

# Pattern Recognition

28th DAGM Symposium  
Berlin, Germany, September 12-14, 2006  
Proceedings

## Volume Editors

Katrin Franke

Bertram Nickolay

Fraunhofer Institute for Production Systems and Design Technology (IPK)

Department of Security Technology, Pascalstr. 8-9, 10587 Berlin, Germany

E-mail: {katrin.franke, bertram.nickolay}@ipk.fraunhofer.de

Klaus-Robert Müller

Fraunhofer Institute for Computer Architecture and Software Technology (FIRST)

Department of Intelligent Data Analysis, Kekulestr. 7, 12489 Berlin, Germany

E-mail: klaus-robot.mueller@first.fraunhofer.de

Ralf Schäfer

Fraunhofer Institute for Information and Communication Technology

Heinrich Hertz Institute (HHI), Department of Electronic Imaging Technology

Einsteinufer 37, 10587 Berlin, Germany

E-mail: ralf.schaefer@hhi.fraunhofer.de

Library of Congress Control Number: 2006932037

CR Subject Classification (1998): I.5, I.4, I.3.5, I.2.10, I.2.6, F.2.2

LNCS Sublibrary: SL 6 – Image Processing, Computer Vision, Pattern Recognition, and Graphics

ISSN 0302-9743

ISBN-10 3-540-44412-2 Springer Berlin Heidelberg New York

ISBN-13 978-3-540-44412-1 Springer Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

Springer is a part of Springer Science+Business Media

springer.com

© Springer-Verlag Berlin Heidelberg 2006

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper SPIN: 11861898 06/3142 5 4 3 2 1 0

# Preface

This LNCS volume contains the papers presented at the 28th Annual Symposium of the German Association for Pattern Recognition, DAGM 2006, held during September 12-14, 2006 at Fraunhofer IPK in Berlin, Germany. This symposium was jointly organized by the three Fraunhofer Institutes HHI, IPK and FIRST, and it was a great honor for the organizers to host such a renowned, scientific event.

In total, 171 papers from 29 countries were submitted, of which 76 (44%) were accepted. We would therefore like to thank all the authors for submitting their work and apologize that not all papers could be accepted. This record number of submissions is an acknowledgement of the high reputation of the DAGM Symposium but at the same time it was a challenge for the Program Committee, as all papers were reviewed by three experts. Therefore we are especially grateful to the 62 members of the Program Committee for their remarkable effort and the high quality as well as the timely delivery of the reviews. Out of the 76 accepted papers, 31 were oral presentations and 45 were posters. However, this selection does not imply any quality ranking but reflects the preference of the authors or the clustering of certain topics.

It was also a special honor to have five very renowned invited speakers at this conference:

- *Gabriel Curio* – Charité, Bernstein Center for Computational Neuroscience, Berlin, Germany
- *Thomas Hofmann* – Technical University Darmstadt, Germany
- *Thomas Huang* – Beckman Institute, University of Illinois, USA
- *Sebastian Thrun* – Artificial Intelligence Lab, Stanford University, USA
- *Patrice Simard* – Document Processing and Understanding (DPU) Group - Microsoft Research, Redmond, USA

These speakers presented their views on the state of the art in pattern recognition and image processing.

One day prior to the symposium, there were four tutorials which gave an in-depth insight into topics of current interest:

- “Elements of Geometric Computer Vision” by Andrea Fusiello – University of Verona, Italy
- “Approximate Probabilistic Inference for Machine Learning” by Manfred Opper – Technical University Berlin, Germany
- “3D Camera Tracking, Reconstruction and View Synthesis at Interactive Frame Rates” by Jan-Michael Frahm – University of North Carolina at Chapel Hill (UNC), USA; Jan-Friso Evers-Senne and Reinhard Koch – Christian-Albrechts-University, Kiel, Germany
- “Level Set Methods in Computer Vision” by Daniel Cremers, Thomas Brox and Kalin Kolev – University of Bonn, Germany

The organization of such an event is not possible without the effort and the enthusiasm of the people involved. We would therefore like to thank all the members of the Local Organizing Committee and the Local Steering Committee. Moreover, special thanks go to Katrin Franke and Mario Köppen for managing the reviewing process, for handling of the papers and for the preparation of the book, to Elnaz Mazandarani for the maintenance of the conference system and to Andrea Semionyk for the complete organization of the event.

We would also like to thank our sponsors Deutsche Telekom Laboratories, Robert Bosch GmbH, Siemens AG, and idalab GmbH for their support of the symposium. We hope that these proceedings, published in Springer's *Lecture Notes in Computer Science* as in previous symposia, will not only impact on the current research of the readers, but will also represent important archival material.

June 2006

Klaus-Robert Müller  
Bertram Nickolay  
Ralf Schäfer

# Awards 2005

## Olympus Prize

The Olympus Prize 2005 was awarded to:

**Michael Felsberg** and  
**Volker Roth**

for their outstanding contributions to the area of image segmentation.

## DAGM Prizes

The main prize for 2005 was awarded to:

**Bodo Rosenhahn, Uwe G. Kersting, Andrew W. Smith,**  
**Jason K. Gurney, Thomas Brox, Reinhard Klette**

A System for Marker-less Human Motion Estimation

Further DAGM prizes for 2005 were awarded to:

**Natalia Slesareva, Andrés Bruhn, Joachim Weickert**

Optic Flow Goes Stereo: A Variational Method for Estimating Discontinuity-  
Preserving Dense Disparity Maps

**Matthias Heiler, Jens Keuchel, Christoph Schnörr**

Semidefinite Clustering for Image Segmentation with A-priori Knowledge

**Olaf Ronneberger, Janis Fehr, Hans Burkhardt**

Voxel-Wise Gray Scale Invariants for Simultaneous Segmentation and  
Classification

**Christian Perwass, Christian Gebken, Gerald Sommer**

Estimation of Geometric Entities and Operators from Uncertain Data

# DAGM 2006 Organization

DAGM 2006 was organized by the Fraunhofer Institutes FIRST, HHI and IPK. This 28th symposium of the German Association for Pattern Recognition (DAGM e.V.) was intended as a convention of renowned experts in all areas of pattern recognition and image processing to present and discuss recent progress and advances.

## Chairmen and Organizing Committees

Honorary Chair	Hans Burkhardt (University of Freiburg)
General Co-chairs	Klaus-Robert Müller (Fraunhofer FIRST) Bertram Nickolay (Fraunhofer IPK) Ralf Schäfer (Fraunhofer HHI)
Program Chair	Katrin Franke (Fraunhofer IPK)
Tutorial Chair	Oliver Schreer (Fraunhofer HHI)
Award Chair	Peter Eisert (Fraunhofer HHI)
Web Chair	Mario Köppen (Fraunhofer IPK)
Organizing Chair	Andrea Semionyk (Fraunhofer HHI)
Local Steering Committee	Olaf Hellwich (TU Berlin) Jörg Krüger (TU Berlin) Klaus Obermayer (TU Berlin) Thomas Tolxdorff (FU Berlin)
Local Organization	Tomasz Gingold (Fraunhofer IPK) Elnaz Mazandarani (Fraunhofer IPK) Anton Schwaighofer (Fraunhofer FIRST)

## Program Committee

Jürgen Beyerer (Uni. Karlsruhe)	Joaquin Candela (MPI Tübingen)
Niels Birbaumer (Uni. Tübingen)	Joaquin Curio (Charité Berlin)
Horst Bischof (TU Graz)	Peter Eisert (Fraunhofer HHI)
Gilles Blanchard (Fraunhofer FIRST)	Wolfgang Förstner (Uni. Bonn)
Benjamin Blankertz (Fraunhofer FIRST)	Armin Grün (ETH Zürich)
Joachim M. Buhmann (ETH Zürich)	Fred Hamprecht (Uni. Heidelberg)
Hans Burkhardt (Uni. Freiburg)	Heinz Handels (Uni. Hamburg)



Olaf Hellwich (TU Berlin)	Günter Palm (Uni. Ulm)
Gerd Hirzinger (DLR Oberpfaffenhofen)	Bernd Radig (TU München)
Thomas Hofmann (TU Darmstadt)	Gunnar Rättsch (F. Miescher Labs)
Joachim Hornegger (Uni. Erlangen)	Martin Riedmiller (Uni. Osnabrück)
Berndt Jähne (Uni. Heidelberg)	Helge Ritter (Uni. Bielefeld)
Herbert Janssen (Honda Research Inst.)	J. Ruiz del Solar (Universidad de Chile)
Xiaoyi Jiang (Uni. Münster)	Dietmar Sagerer (Uni. Bielefeld)
Thorsten Joachims (Cornell)	Dietmar Saupe (Uni. Konstanz)
Wilhelm Kincses (Daimler-Chrysler)	Tobias Scheffer (HU Berlin)
Reinhard Koch (Uni. Kiel)	Bernt Schiele (ETH Zürich)
Mario Köppen (Fraunhofer IPK)	Bernhard Schölkopf (MPI Tübingen)
Edgar Körner (Honda Research Inst.)	Christoph Schnörr (Uni. Mannheim)
Ulrich Kressel (Daimler-Chrysler)	Hans-Peter Seidel (MPI Saarbrücken)
Walter Kropatsch (TU Wien)	Gerald Sommer (Uni. Kiel)
Jürgen Kurths (Uni. Potsdam)	John Shawe-Taylor (Uni. of Southampton)
Thomas Martinetz (Uni. Lübeck)	Alex Smola (ANU)
Helmut Mayer (Uni.-BW München)	Winfried Teiwes (SMI Teltow)
Rudolf Mester (Uni. Frankfurt)	Thomas Tolxdorff (FU Berlin)
Bernd Michaelis (Uni. Magdeburg)	Thomas Vetter (Uni. Basel)
Hans-Hellmut Nagel (Uni. Karlsruhe)	Friedrich M. Wahl (Uni. Braunschweig)
Hermann Ney (RWTH Aachen)	Joachim Weickert (Uni. Saarland)
Heinrich Niemann (Uni. Erlangen)	Heinz Wörn (Uni. Karlsruhe)
Klaus Obermayer (TU Berlin)	Stefan Wrobel (Fraunhofer AIS)
Manfred Opper (TU Berlin)	
Manfred Paeschke (Bundesdruckerei)	

## Additional Referees

Devrim Akca	Charanpal Dhanjal	Jürgen Gall
Klaus Arbter	Stephan Didas	Christian Gebken
Martin Böhme	Gyuri Dorko	Stefan Gehrig
Claus Bahlmann	Guido Dornhege	Christian Gollan
Matthew Blaschko	Michael Dorr	Markus Grabner
Fabio de Bona	Philippe Dreuw	Horia Greco
Michael Brückner	Jan Ehrhardt	Allan Hanbury
Mikio L. Braun	MarkusENZweiler	Marc Hanheide
Ulf Brefeld	Kai Essig	Stefan Harmeling
Michael Breuss	Jörg Finger	Sasa Hasan
Andres Bruhn	Thomas Finley	Juergen Hesser
Jan Bungeroth	M.O. Franz	N. Jeremy Hill
Nikos Canterakis	Mario Fritz	Ulrich Hillenbrand
Marco A. Chavarria	Linus Görlitz	Heiko Hirschmüller
Alexandru Condurache	Michael Götting	Florian Hoppe
Thomas Deselaers	Dilan Gürur	Zakria Hussain

Adrian Ion	Petra Philips	Daniel Stein
Brijnesh J. Jain	Luis Pizarro	Klaus Strobl
Martin Kampel	Thomas Pock	Thorsten Sy
Jens Kohlmorgen	F.C. Popescu	Ryota Tamioka
Istvan Kokai	Herward Prehn	Alexandra Teynor
Reinhard Konthe	Matthias Raetsch	Marko Tscherepanow
Franz Kummert	Marco Reisert	Christian Veenhuis
Marcel Lüthi	Konrad Rieck	Rodrigo Verschae
Kai Labusch	Markus Rilk	Raul Vicente Garcia
Julian Laub	Sami Romdhani	Sven Wachsmuth
Steven Lemm	Bodo Rosenhahn	Christian James Walder
Ingo Luetkebohle	Volker Roth	Zhuoran Wang
Karsten Müller	Christian Schmaltz	Martin Welk
Nikodem Majer	Jan Schneider	Ralf Westphal
Arne Mauser	Oliver Schreer	Wilhelm Wilke
Frank C. Meinecke	Gabriele Schweikert	Simon Winkelbach
Zaharya Menevidis	Matthias Seeger	Sebastian Wrede
Sven Molkenstruck	Edgar Seemann	Mingrui Wu
Ugur Oezdemir	Wolfgang Sepp	Ulas Yilmaz
Björn Ommer	Lokesh Setia	Georg Zeller
Cheng Soon Ong	Benyah Shaparenko	Andreas Ziehe
Peter Orbanz	Nils T. Siebel	Alexander Zien
Pavel Pavlov	Natalia Slesareva	

## Sponsoring Institutions

Deutsche Telekom Laboratories, Berlin, Germany  
 Robert Bosch GmbH, Stuttgart, Germany  
 Siemens AG, Munich, Germany  
 idalab GmbH, Berlin, Germany

# Table of Contents

## Image Filtering, Restoration and Segmentation

Ultrasound Image Denoising by Spatially Varying Frequency Compounding . . . . .	1
<i>Yael Erez, Yoav Y. Schechner, Dan Adam</i>	
Exploiting Low-Level Image Segmentation for Object Recognition . . . . .	11
<i>Volker Roth, Björn Ommer</i>	
Wavelet Based Noise Reduction by Identification of Correlations . . . . .	21
<i>Anja Borsdorf, Rainer Raupach, Joachim Hornegger</i>	
Template Based Gibbs Probability Distributions for Texture Modeling and Segmentation . . . . .	31
<i>Dmitrij Schlesinger</i>	
Efficient Combination of Probabilistic Sampling Approximations for Robust Image Segmentation . . . . .	41
<i>Jens Keuchel, Daniel Küttel</i>	
Diffusion-Like Reconstruction Schemes from Linear Data Models . . . . .	51
<i>Hanno Scharr</i>	
Reduction of Ring Artifacts in High Resolution X-Ray Microtomography Images . . . . .	61
<i>Maria Axelsson, Stina Svensson, Gunilla Borgefors</i>	
A Probabilistic Multi-phase Model for Variational Image Segmentation . . . . .	71
<i>Thomas Pock, Horst Bischof</i>	
Provably Correct Edgel Linking and Subpixel Boundary Reconstruction . . . . .	81
<i>Ulrich Köthe, Peer Stelldinger, Hans Meine</i>	
The Edge Preserving Wiener Filter for Scalar and Tensor Valued Images . . . . .	91
<i>Kai Krajssek, Rudolf Mester</i>	

From Adaptive Averaging to Accelerated Nonlinear  
Diffusion Filtering ..... 101  
*Stephan Didas, Joachim Weickert*

Introducing Dynamic Prior Knowledge to Partially-Blurred Image  
Restoration ..... 111  
*Hongwei Zheng, Olaf Hellwich*

## Shape Analysis and Representation

On-Line, Incremental Learning of a Robust Active  
Shape Model ..... 122  
*Michael Fussenegger, Peter M. Roth, Horst Bischof, Axel Pinz*

Using Irreducible Group Representations for Invariant 3D Shape  
Description ..... 132  
*Marco Reisert, Hans Burkhardt*

Shape Matching by Variational Computation of Geodesics  
on a Manifold ..... 142  
*Frank R. Schmidt, Michael Clausen, Daniel Cremers*

A Modification of the Level Set Speed Function  
to Bridge Gaps in Data ..... 152  
*Karsten Rink, Klaus Tönnies*

Generation and Initialization of Stable 3D Mass-Spring Models  
for the Segmentation of the Thyroid Cartilage ..... 162  
*Jana Dornheim, Lars Dornheim, Bernhard Preim, Ilka Hertel,  
Gero Strauss*

Preserving Topological Information in the Windowed Hough Transform  
for Rectangle Extraction ..... 172  
*Dan Cireşan, Dana Damian*

## Recognition, Categorization and Detection

Fast Scalar and Vectorial Grayscale Based Invariant Features for 3D  
Cell Nuclei Localization and Classification ..... 182  
*Janina Schulz, Thorsten Schmidt, Olaf Ronneberger,  
Hans Burkhardt, Taras Pasternak, Alexander Dovzhenko,  
Klaus Palme*

Integrating Recognition and Reconstruction for Cognitive Traffic Scene Analysis from a Moving Vehicle . . . . .	192
<i>Bastian Leibe, Nico Cornelis, Kurt Cornelis, Luc Van Gool</i>	
Sparse Patch-Histograms for Object Classification in Cluttered Images . . . . .	202
<i>Thomas Deselaers, Andre Hegerath, Daniel Keysers, Hermann Ney</i>	
An Object-Oriented Approach Using a Top-Down and Bottom-Up Process for Manipulative Action Recognition . . . . .	212
<i>Zhe Li, Jannik Fritsch, Sven Wachsmuth, Gerhard Sagerer</i>	
Detecting Intrinsically Two-Dimensional Image Structures Using Local Phase . . . . .	222
<i>Di Zang, Gerald Sommer</i>	
Towards Unsupervised Discovery of Visual Categories . . . . .	232
<i>Mario Fritz, Bernt Schiele</i>	
Cross-Articulation Learning for Robust Detection of Pedestrians . . . . .	242
<i>Edgar Seemann, Bernt Schiele</i>	
Analysis on a Local Approach to 3D Object Recognition . . . . .	253
<i>Elisabetta Delponete, Elise Arnaud, Francesca Odone, Alessandro Verri</i>	
Phase Based 3D Texture Features . . . . .	263
<i>Janis Fehr, Hans Burkhardt</i>	
Learning of Graphical Models and Efficient Inference for Object Class Recognition . . . . .	273
<i>Martin Bergtholdt, Jörg H. Kappes, Christoph Schnörr</i>	
Properties of Patch Based Approaches for the Recognition of Visual Object Classes . . . . .	284
<i>Alexandra Teynor, Esa Rahtu, Lokesh Setia, Hans Burkhardt</i>	
<b>Computer Vision and Image Retrieval</b>	
Feature Selection for Automatic Image Annotation . . . . .	294
<i>Lokesh Setia, Hans Burkhardt</i>	

Image Database Navigation on a Hierarchical MDS Grid . . . . . 304  
*Gerald Schaefer, Simon Ruszala*

Linear vs. Nonlinear Feature Combination for Saliency Computation:  
 A Comparison with Human Vision . . . . . 314  
*Nabil Ouerhani, Alexandre Bur, Heinz Hügli*

Facial Expression Modelling from Still Images Using a Single Generic  
 3D Head Model . . . . . 324  
*Michael Hähnel, Andreas Wiratanaya, Karl-Friedrich Kraiss*

Extraction of Haar Integral Features on Omnidirectional Images:  
 Application to Local and Global Localization . . . . . 334  
*Ouiddad Labbani-Igbida, Cyril Charron,  
 El Mustapha Mouaddib*

## Machine Learning and Statistical Data Analysis

Model Selection in Kernel Methods Based on a Spectral Analysis  
 of Label Information . . . . . 344  
*Mikio L. Braun, Tilman Lange, Joachim M. Buhmann*

Importance-Weighted Cross-Validation for Covariate Shift . . . . . 354  
*Masashi Sugiyama, Benjamin Blankertz, Matthias Krauledat,  
 Guido Dornhege, Klaus-Robert Müller*

Parameterless Isomap with Adaptive Neighborhood Selection . . . . . 364  
*Nathan Mekuz, John K. Tsotsos*

Efficient Algorithms for Similarity Measures over Sequential Data:  
 A Look Beyond Kernels . . . . . 374  
*Konrad Rieck, Pavel Laskov, Klaus-Robert Müller*

Multi-scale Bayesian Based Horizon Matchings Across Faults in 3d  
 Seismic Data . . . . . 384  
*Fitsum Admasu, Klaus Tönnies*

## Biomedical Data Analysis

Robust MEG Source Localization of Event Related Potentials:  
 Identifying Relevant Sources by Non-Gaussianity . . . . . 394  
*Peter Breun, Moritz Grosse-Wentrup, Wolfgang Utschick,  
 Martin Buss*

Classifying Event-Related Desynchronization in EEG, ECoG and MEG Signals . . . . .	404
<i>N. Jeremy Hill, Thomas Navin Lal, Michael Schröder, Thilo Hinterberger, Guido Widman, Christian E. Elger, Bernhard Schölkopf, Niels Birbaumer</i>	
Optimizing Spectral Filters for Single Trial EEG Classification . . . . .	414
<i>Ryota Tomioka, Guido Dornhege, Guido Nolte, Kazuyuki Aihara, Klaus-Robert Müller</i>	
Probabilistic De Novo Peptide Sequencing with Doubly Charged Ions . . . .	424
<i>Hansruedi Peter, Bernd Fischer, Joachim M. Buhmann</i>	
<b>Motion Analysis and Tracking</b>	
Direct Estimation of the Wall Shear Rate Using Parametric Motion Models in 3D . . . . .	434
<i>Markus Jehle, Bernd Jähne, Ulrich Kertzscher</i>	
On-Line Variational Estimation of Dynamical Fluid Flows with Physics-Based Spatio-temporal Regularization . . . . .	444
<i>Paul Ruhnau, Annette Stahl, Christoph Schnörr</i>	
Near Real-Time Motion Segmentation Using Graph Cuts . . . . .	455
<i>Thomas Schoenemann, Daniel Cremers</i>	
Segmentation-Based Motion with Occlusions Using Graph-Cut Optimization . . . . .	465
<i>Michael Bleyer, Christoph Rhemann, Margrit Gelautz</i>	
Realtime Depth Estimation and Obstacle Detection from Monocular Video . . . . .	475
<i>Andreas Wedel, Uwe Franke, Jens Klappstein, Thomas Brox, Daniel Cremers</i>	
3D Human Motion Sequences Synchronization Using Dense Matching Algorithm . . . . .	485
<i>Mikhail Mozerov, Ignasi Rius, Xavier Roca, Jordi González</i>	
Cloth X-Ray: MoCap of People Wearing Textiles . . . . .	495
<i>Bodo Rosenhahn, Uwe G. Kersting, Katie Powell, Hans-Peter Seidel</i>	

Unconstrained Multiple-People Tracking . . . . .	505
<i>Daniel Rowe, Ian Reid, Jordi González, Juan Jose Villanueva</i>	
Robust Non-rigid Object Tracking Using Point Distribution Manifolds . . . . .	515
<i>Tom Mathes, Justus H. Piater</i>	
A Variational Approach to Joint Denoising, Edge Detection and Motion Estimation . . . . .	525
<i>Alexandru Telea, Tobias Preusser, Christoph Garbe, Marc Droske, Martin Rumpf</i>	
Multi-step Multi-camera View Planning for Real-Time Visual Object Tracking . . . . .	536
<i>Benjamin Deutsch, Stefan Wenhardt, Heinrich Niemann</i>	
<b>Pose Recognition</b>	
Nonparametric Density Estimation for Human Pose Tracking . . . . .	546
<i>Thomas Brox, Bodo Rosenhahn, Uwe G. Kersting, Daniel Cremers</i>	
Learning to Mimic Motion of Human Arm and Hand Grabbing for Constraint Adaptation . . . . .	556
<i>Stephan Al-Zubi, Gerald Sommer</i>	
Visual Hand Posture Recognition in Monocular Image Sequences . . . . .	566
<i>Thorsten Dick, Jörg Zieren, Karl-Friedrich Kraiss</i>	
Monocular Body Pose Estimation by Color Histograms and Point Tracking . . . . .	576
<i>Daniel Grest, Dennis Herzog, Reinhard Koch</i>	
Pose Estimation from Uncertain Omnidirectional Image Data Using Line-Plane Correspondences . . . . .	587
<i>Christian Gebken, Antti Tolvanen, Gerald Sommer</i>	
Kernel Particle Filter for Visual Quality Inspection from Monocular Intensity Images . . . . .	597
<i>Dirk Stöbel, Gerhard Sagerer</i>	



## Stereo and Structure from Motion

Monocular 3D Scene Reconstruction at Absolute Scales by Combination of Geometric and Real-Aperture Methods . . . . .	607
<i>Annika Kuhl, Christian Wöhler, Lars Krüger, Pablo d'Angelo, Horst-Michael Groß</i>	
An Effective Stereo Matching Algorithm with Optimal Path Cost Aggregation . . . . .	617
<i>Mikhail Mozerov</i>	
Tracking Camera Parameters of an Active Stereo Rig . . . . .	627
<i>Thao Dang, Christian Hoffmann</i>	
Handling Camera Movement Constraints in Reinforcement Learning Based Active Object Recognition . . . . .	637
<i>Christian Derichs, Heinrich Niemann</i>	

## Multi-view Image and Geometric Processing

The Inversion Camera Model . . . . .	647
<i>Christian Perwass, Gerald Sommer</i>	
Determining an Initial Image Pair for Fixing the Scale of a 3D Reconstruction from an Image Sequence . . . . .	657
<i>Christian Beder, Richard Steffen</i>	
Multi-camera Radiometric Surface Modelling for Image-Based Re-lighting . . . . .	667
<i>Oliver Grau</i>	
A Multiple Graph Cut Based Approach for Stereo Analysis . . . . .	677
<i>Ulas Vural, Yusuf Sinan Akgul</i>	
Robust Variational Segmentation of 3D Objects from Multiple Views . . . .	688
<i>Kalin Kolev, Thomas Brox, Daniel Cremers</i>	
Online Calibration of Two Zoom-Pan-Tilt Units for Planar Dynamic Events . . . . .	698
<i>Kurt Cornelis, Nico Cornelis, Maarten Aerts, Egemen Özden, Luc Van Gool</i>	
Dense Stereo by Triangular Meshing and Cross Validation . . . . .	708
<i>Peter Wey, Bernd Fischer, Herbert Bay, Joachim M. Buhmann</i>	

## 3D View Registration and Surface Modelling

Low-Cost Laser Range Scanner and Fast Surface Registration Approach . . . . .	718
<i>Simon Winkelbach, Sven Molkenstruck, Friedrich M. Wahl</i>	
A Point-Based Approach to PDE-Based Surface Reconstruction . . . . .	729
<i>Christian Linz, Bastian Goldlücke, Marcus Magnor</i>	
Robust Feature Representation for Efficient Camera Registration . . . . .	739
<i>Kevin Köser, Volker Härtel, Reinhard Koch</i>	
Reconstruction of Façade Structures Using a Formal Grammar and RjMCMC . . . . .	750
<i>Nora Ripperda, Claus Brenner</i>	
Stable Wave Detector of Blobs in Images . . . . .	760
<i>Jan Dupač, Václav Hlaváč</i>	
<b>Author Index . . . . .</b>	<b>771</b>

# Ultrasound Image Denoising by Spatially Varying Frequency Compounding\*

Yael Erez<sup>1</sup>, Yoav Y. Schechner<sup>1</sup>, and Dan Adam<sup>2</sup>

<sup>1</sup> Dept. Electrical Engineering, Technion – Israel Inst. Tech.,  
Haifa 32000, Israel

{yaele@tx, yoav@ee}.technion.ac.il

<sup>2</sup> Dept. Biomedical Engineering, Technion – Israel Inst. Tech.,  
Haifa 32000, Israel

dan@bm.technion.ac.il

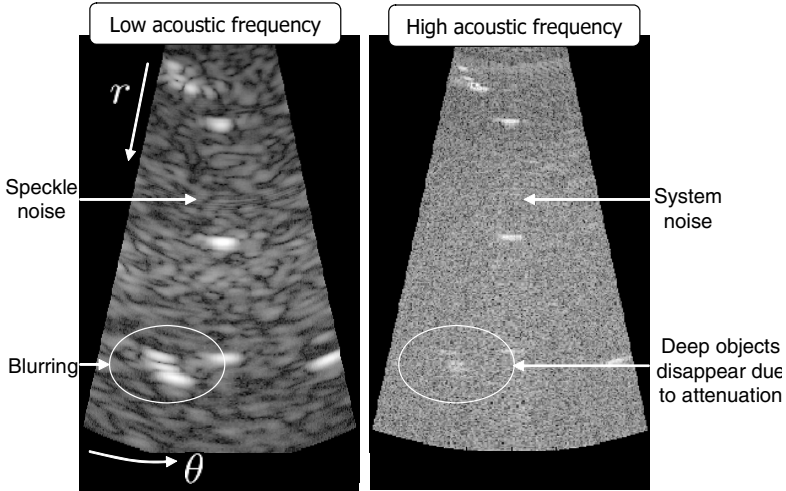
**Abstract.** Ultrasound images are very noisy. Along with system noise, a significant noise source is the speckle phenomenon, caused by interference in the viewed object. Most past approaches for denoising ultrasound images essentially blur the image, and they do not handle attenuation. Our approach, on the contrary, *does not* blur the image and *does* handle attenuation. Our denoising approach is based on frequency compounding, in which images of the same object are acquired in different acoustic frequencies, and then compounded. Existing frequency compounding methods have been based on simple averaging, and have achieved only limited enhancement. The reason is that the statistical and physical characteristics of the signal and noise vary with depth, and the noise is correlated. Hence, we suggest a spatially varying frequency compounding, based on understanding of these characteristics. Our method suppresses the various noise sources and recovers attenuated objects, while maintaining high resolution.

## 1 Introduction

Ultrasound is an imaging technique that uses high frequency acoustic waves. It is safe, suitable for many applications and is relatively cheap. It is used in sonar, medical imaging and material science work. However, there are some problems that interfere with the diagnosis. Fig. 1 illustrates some of these problems. The most prominent problem, which distinguishes ultrasound from most imaging techniques, is strong speckle noise. Speckles appear as grains of different sizes and intensities, that result from the coherent nature of the ultrasound radiation [2]. The speckle image is signal dependent. It is time invariant and thus

---

\* Yoav Schechner is a Landau Fellow-supported by the Taub Foundation, and an Alon Fellow. This research was partly supported by the Israel Science Foundation (Grant No. 315/04), by the Ministry of Industry and Trade, office of the Chief Scientist - "Magnetron" Program, and by the Technion VPR Fund for Research & Development. This continuous assistance is deeply appreciated. The research was carried out in the Ollendorff Minerva Center. Minerva is funded through the BMBF.



**Fig. 1.** Problems that disrupt diagnosis in ultrasound imaging. The depth range is 0 – 12cm.

cannot be suppressed by temporal averaging. A second problem is attenuation. The acoustic signal propagating in the medium is scattered and absorbed [2], and hence attenuated. This phenomenon is more pronounced in high acoustic frequencies. When the attenuated signal is amplified, it is accompanied by amplification of system noise, which is signal independent. Below a certain level of signal to noise ratio (SNR), objects are overwhelmed by system noise, thus amplification in post-processing does not reconstruct these objects.

Most past approaches for denoising ultrasound images have used standard image reconstruction tools, such as weighted median filter [9], wavelet based methods [4] [5], Gaussian non-linear filters [3] and anisotropic diffusion [14]. All these methods essentially blur the image. Moreover, they do not handle spatially varying physical effects, as attenuation. Another approach is *frequency compounding*,<sup>1</sup> in which images of an object are acquired in different acoustic frequencies, and then compounded [10]. Existing compounding methods [1] have used simple processing methods such as pointwise arithmetic averaging, and have achieved only limited enhancement.

In this paper we present a method that does not suffer from the mentioned disadvantages. It is based on frequency compounding, and the images are analyzed in a stochastic manner. The stochastic denoising is spatially varying and it is based on statistical and physical characteristics of the signal and noise as a function of depth and acoustic frequency. The stochastic denoising shows significant speckle reduction, with no resolution loss, while deep objects are reconstructed as well.

<sup>1</sup> *Spatial compounding* is also possible. Yet, it introduces a complex registration problem, and it does not improve detection in deep regions.

## 2 Theoretical Background

Let us first model blur. We assume the ultrasound images to be two-dimensional (2D), given in their polar coordinates  $(r, \theta)$ . The  $r$  coordinate (radial axis) is the axis of wave propagation, and  $\theta$  (lateral axis) represents a serial scan of the direction of the radiating ultrasound beam. The 2D signal measured by the system is the result of a natural filtering of the 2D tissue reflectivity function  $a_0(r, \theta)$  with a 2D point spread function (PSF). This PSF is space variant. In particular, its lateral support changes with the depth  $r$ : the acoustic beam is focused at a certain depth, where the lateral PSF is narrowest, while at other depths this PSF gradually widens. Yet, in small regions we can assume this filter to be space invariant. There, the measured signal is

$$a^{\text{RF}}(r, \theta) = a_0(r, \theta) * h(r, \theta) . \quad (1)$$

Following [12], it is reasonable to assume the PSF to be separable. The PSF also depends on system properties, such as acoustic frequency [2].

Image formation is also affected by attenuation of ultrasound in the medium [2]. A general simple and effective model of the amplitude of the signal is

$$a^{\text{RF}}(r, \theta) = e^{-2\alpha r f_{\text{acoustic}}} a_0(r, \theta) * h(r, \theta) , \quad (2)$$

where  $\alpha$  is the *attenuation coefficient* of the acoustic amplitude, and  $f_{\text{acoustic}}$  is the acoustic frequency. A rule of thumb [2] is: attenuation in tissue is approximately  $1\text{dB}/(\text{cm} \cdot \text{MHz})$ , for a signal going from a probe to the object and then returning. It is clear from Eq. (2) that attenuation depends on the acoustic frequency: high acoustic frequencies suffer from stronger attenuation and thus a lower SNR, particularly at large depths. This is evident in Fig. 1

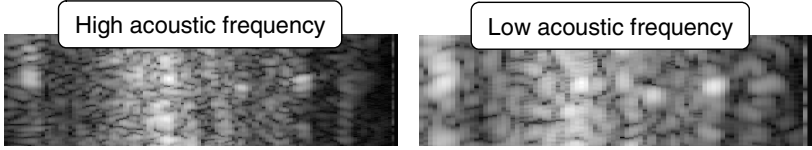
In ultrasound systems, the measured signal  $a^{\text{RF}}$  undergoes several standard conversion steps. First, attenuation is compensated for. Then, the acoustic modulation is extracted: note that  $a^{\text{RF}}$  is a high-frequency ( $\text{MHz}$ ) signal, which is modulated by the tissue reflectivity function. To extract the tissue information, the envelope of the attenuation-compensated  $a^{\text{RF}}$  is detected, yielding

$$a^{\text{magnitude}}(r, \theta) = |\text{envelope}[e^{2\alpha r f_{\text{acoustic}}} \cdot a^{\text{RF}}(r, \theta)]| , \quad (3)$$

where  $\text{envelope}[g(r)]$  is an operator [2] that extracts the envelope of a modulated wave  $g(r)$  (recall that  $r$  is the axis of wave propagation). Note that Eq. (3) derives the *modulus* of the envelope, since the envelope is complex, in general.

### Speckle Noise

Speckle noise has a granular texture, as presented in Fig. 2. Speckles degrade the ability to resolve details and detect objects of size comparable to the speckle size. This noise stems from point scatterers in an homogenous tissue, that cannot be resolved by the ultrasound system. These point scatterers, which are much smaller than the ultrasound wavelength, scatter the wave. Two or more waves



**Fig. 2.** Speckle appearance of the same tissue in different acoustic frequencies. High acoustic frequency speckles are smaller than the low acoustic frequency speckles.

travelling to the probe from such scatterers may interfere with each other, constructively or destructively, creating bright and dark spots, termed speckles. For interference, the backscattered signal from the scatterers should overlap in time and space. This happens when the distance between them is within the PSF (radially and laterally) support. This is an *important* point to remember: the speckle typical size is similar to the PSF support. Since the PSF changes with depth, the statistics of this noise are space (depth)-variant. Furthermore, they change when the acoustic frequency used to acquire the image changes, as shown in Fig. 2, as the PSF does. We exploit these properties in this paper.

Speckle is generally modelled as multiplicative noise [7]. The overall detected magnitude is

$$a^{\text{total}}(r, \theta) = a^{\text{magnitude}}(r, \theta) \cdot s^{\text{magnitude}}(r, \theta) + \eta(r, \theta), \quad (4)$$

where the real number  $s^{\text{magnitude}}$  represents real nonnegative speckle noise at certain coordinates, and  $\eta$  represents system noise there. The system noise increases with depth, due to the attenuation compensation done in Eq. (3). Still, assume for a moment that the additive noise is sufficiently small compared to the multiplicative noise. Then, a log operation on Eq. (4) transforms speckles to additive noise

$$\underbrace{\log [a^{\text{total}}(r, \theta)]}_{a^{\log}} \approx \underbrace{\log [a^{\text{magnitude}}(r, \theta)]}_{\log(a^{\text{magnitude}})} + \underbrace{\log [s^{\text{magnitude}}(r, \theta)]}_{s^{\log}}. \quad (5)$$

The logarithm operation is standard when displaying ultrasound images on a computer screen [2], since the dynamic range of  $a^{\text{total}}$  is very large [2]. Therefore, in the image used for display, the speckle noise is already additive.

### 3 Solution

Our solution is spatially varying frequency compounding, based on the best linear unbiased estimator (BLUE), also known as Gauss-Markov or weighted least squares (WLS) [8]. This stochastic method relies on the following principles:

- The compounding should be space (depth) variant, since the statistics of noise change with the depth  $r$ , as the PSF.

- In speckles, adjacent pixels are correlated [12]. Therefore, it is desirable that compounding would account for this spatial correlation.
- Speckles in different images of the same object, acquired with different acoustic frequencies, are correlated [13]. Therefore, simple averaging is not very efficient for speckle reduction. On the contrary, we should account for the cross-correlation between different acoustic channels.
- The method is not intended for sharpening. Therefore, it does not include deblurring. Nevertheless, we do not want to further blur existing information.
- In general, deep objects are not visible in high acoustic frequency (due to increased attenuation). However, thanks to our use of a low acoustic frequency image in the compounding, we should end up seeing even the deepest objects.
- In general, spatial resolution is low, when using a low acoustic frequency (due to a wider PSF). However, thanks to our use of a high acoustic frequency image in the compounding, we should end up with high spatial resolution, at least in close distance.

In the following we detail our solution.

### 3.1 Speckle Model

We refer to the signals  $\mathbf{a}^{\text{magnitude}}$  and  $\mathbf{a}^{\text{log}}$  as discrete  $N \times 1$  vectors. When acquiring  $K$  images in different acoustic frequencies, then based on Eq. (5),

$$\begin{pmatrix} \mathbf{a}_1^{\text{log}} \\ \mathbf{a}_2^{\text{log}} \\ \vdots \\ \mathbf{a}_K^{\text{log}} \end{pmatrix} = \begin{pmatrix} \log \mathbf{a}_1^{\text{magnitude}} \\ \log \mathbf{a}_2^{\text{magnitude}} \\ \vdots \\ \log \mathbf{a}_K^{\text{magnitude}} \end{pmatrix} + \begin{pmatrix} \mathbf{s}_1^{\text{log}} \\ \mathbf{s}_2^{\text{log}} \\ \vdots \\ \mathbf{s}_K^{\text{log}} \end{pmatrix}. \quad (6)$$

At this point we use the principle mentioned above, of not attempting to invert blur, thus we do not consider the blur  $h$  in the reconstruction. Using a  $\delta$  function for  $h$  in Eq. (2) can estimate  $\hat{a}_0(r, \theta) = e^{2\alpha r f_{\text{acoustic}}} a^{\text{RF}}(r, \theta)$ . Therefore, we set

$$\mathbf{a}_k^{\text{magnitude}} \approx |\text{envelope}(\hat{\mathbf{a}}_0)|, \quad (7)$$

for all  $k$ . Now, the frames  $\mathbf{a}_k^{\text{magnitude}}$  differ in the noise, which is indeed different, especially the speckle noise. All frames include a similar object content, i.e.,

$$\mathbf{a}_1^{\text{magnitude}} \approx \mathbf{a}_2^{\text{magnitude}} \approx \dots \approx \mathbf{a}_K^{\text{magnitude}} = \mathbf{a}^{\text{magnitude}}, \quad (8)$$

Hence, Eq. (6) reduces to

$$\begin{pmatrix} \mathbf{a}_1^{\text{log}} \\ \mathbf{a}_2^{\text{log}} \\ \vdots \\ \mathbf{a}_K^{\text{log}} \end{pmatrix} = \begin{pmatrix} \mathbf{I} \\ \mathbf{I} \\ \vdots \\ \mathbf{I} \end{pmatrix} \log(\mathbf{a}^{\text{magnitude}}) + \begin{pmatrix} \mathbf{s}_1^{\text{log}} \\ \mathbf{s}_2^{\text{log}} \\ \vdots \\ \mathbf{s}_K^{\text{log}} \end{pmatrix}. \quad (9)$$

### 3.2 BLUE

Consider data  $\mathbf{a}^{\text{data}}$  in the general linear model

$$\mathbf{a}^{\text{data}} = \mathbf{H}\mathbf{a} + \mathbf{n} , \quad (10)$$

where  $\mathbf{H}$  is a known  $KN \times N$  matrix (operator),  $\mathbf{a}$  is an  $N \times 1$  vector of variables to be estimated, and  $\mathbf{n}$  is an  $N \times 1$  noise vector with zero mean and covariance  $\mathbf{C}$ . The Gauss-Markov theorem [8] states that the BLUE of  $\mathbf{a}$  is

$$\hat{\mathbf{a}} = (\mathbf{H}^T \mathbf{C}^{-1} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{C}^{-1} \mathbf{a}^{\text{data}} . \quad (11)$$

To apply the BLUE on Eq. (9), we substitute  $\mathbf{a} = \log(\mathbf{a}^{\text{magnitude}})$  as in Eqs. (10,11), while  $\mathbf{a}^{\text{data}}$  represents the vector on the left-hand-side of Eq. (9). Now, the noise covariance matrix  $\mathbf{C}$  used in Eq. (11) has the form

$$\mathbf{C} = \begin{pmatrix} \mathbf{C}_{\mathbf{s}_1^{\log} \mathbf{s}_1^{\log}} & \mathbf{C}_{\mathbf{s}_1^{\log} \mathbf{s}_2^{\log}} & \cdots & \mathbf{C}_{\mathbf{s}_1^{\log} \mathbf{s}_K^{\log}} \\ \mathbf{C}_{\mathbf{s}_2^{\log} \mathbf{s}_1^{\log}} & \mathbf{C}_{\mathbf{s}_2^{\log} \mathbf{s}_2^{\log}} & \cdots & \mathbf{C}_{\mathbf{s}_2^{\log} \mathbf{s}_K^{\log}} \\ \vdots & & \ddots & \\ \mathbf{C}_{\mathbf{s}_K^{\log} \mathbf{s}_1^{\log}} & \mathbf{C}_{\mathbf{s}_K^{\log} \mathbf{s}_2^{\log}} & \cdots & \mathbf{C}_{\mathbf{s}_K^{\log} \mathbf{s}_K^{\log}} \end{pmatrix} , \quad (12)$$

where  $\mathbf{C}_{\mathbf{s}_k^{\log} \mathbf{s}_i^{\log}}$  is the cross-covariance matrix between two speckle images  $\mathbf{s}_k^{\log}$  and  $\mathbf{s}_i^{\log}$  in different acoustic frequencies. Eq. (11) performs a linear combination of all data  $\mathbf{a}^{\text{data}}$  (all pixels in all images) in order to estimate the value in each pixel of  $\hat{\mathbf{a}}$ . Therefore, the BLUE may potentially perform deconvolution, in addition to noise averaging. Nevertheless, in our case

$$\mathbf{H} = (\mathbf{I}, \mathbf{I}, \dots, \mathbf{I})^T , \quad (13)$$

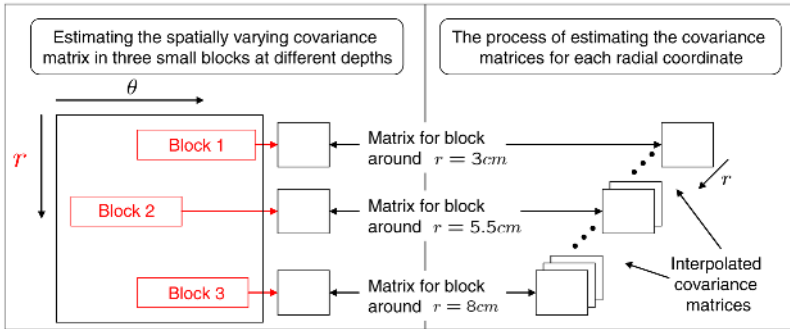
since we do not attempt deblurring. The BLUE exploits the correlation between variables. This enables denoising based on partially correlated variables, in contrary to a simple average, which implicitly assumes uncorrelated variables.

### 3.3 Spatially Varying BLUE

To use the BLUE we need to know the noise mean and covariance (*statistics*), in the set of images we use. When applying the method for noise reduction, we need to consider the noise statistics. We estimate the covariance functions from the data itself. We performed empirical measurements of these functions. This empirical study revealed that the noise is not stationary. This is not surprising, since according to [12], the auto and cross correlations of speckles depend on the system PSF, which (Sec. 2) changes significantly with depth.

Let us first examine a certain block in the image. We can assume stationarity within this block. However, the statistics change in different image regions. Is there a need to divide the whole image to blocks, and measure the statistics





**Fig. 3.** Estimating the spatially varying covariance matrix

within each of them? Practically, the answer is *No*. Since the statistics change gradually, it is possible to examine a few blocks in the field of view (FOV) as illustrated in the left side of Fig. 3, and measure the noise statistics only within them. This processing is applied in the polar coordinate space, as illustrated in Fig. 3. Then the speckle statistics around any point in the FOV can be deduced. The measurement of the statistics in these few selected blocks is described in Sec 3.4, as is the inference from these few blocks to any other region.

The BLUE requires the cross-correlation between different channels. As any cross-correlation function, it depends on the lag between pixels. When taking into account a maximum lag of  $d_{\max}^{\text{radial}}$  in the radial direction and a maximum lag of  $d_{\max}^{\text{lateral}}$  in the lateral direction, the size of the covariance matrix equals  $(d_{\max}^{\text{radial}} \cdot d_{\max}^{\text{lateral}} \cdot K)^2$ . Empirical measurements that we performed in several images showed a fast decrease in the off-diagonal elements of  $\mathbf{C}_{S_k^{\log} S_i^{\log}}$ . We conclude that the lengths of the spatial correlation are short. Hence, small lags are sufficient to reflect the statistics. We are thus allowed to use small regions, for which the radial maximum lag is  $\approx 40$  pixels corresponding to  $\approx 1.5\text{mm}$  in our system.

We now have the statistics in a few blocks. Then, using interpolation, we infer the statistics in any region centered on any pixel in the FOV. Subsequently, we can apply the BLUE around each pixel in the image. In other words, around each pixel, we define a small region, and since the noise statistics in this region has been estimated in the previous steps, we can apply the BLUE for it, and estimate  $\log(\mathbf{a}^{\text{magnitude}})$  at that location.

### 3.4 Measuring Statistics

We have seen in Sec. 3.3 that we use few small blocks in the image, to measure the covariance matrix, which is spatially variant. We chose blocks in which there is no meaningful object detail.<sup>2</sup>

<sup>2</sup> Practically, we would not expect a physician to select such blocks manually in each session. Hence, the typical covariance matrix can be learned using sets of typical speckle images of arbitrary objects. This is a matter we intend for future research.

Covariance depends on both lateral and radial lags. Furthermore, radial and lateral correlations differ. Based on the separability of the PSF [12], the covariance matrix is also separable [12]. For each matrix element  $[d, q]$

$$\hat{\mathbf{C}}[d, q] = \hat{\mathbf{C}}^{\text{radial}}[d, q] \cdot \hat{\mathbf{C}}^{\text{lateral}}[d, q], \quad (14)$$

where  $\hat{\mathbf{C}}^{\text{radial}}$  and  $\hat{\mathbf{C}}^{\text{lateral}}$  are the noise covariance matrices in the radial and lateral directions, respectively. Both matrices are measured in a similar way. For example, the cross covariance in the radial direction between two acoustic frequencies  $k$  and  $i$ , is estimated as

$$\hat{\mathbf{C}}_{\mathbf{s}_k \mathbf{s}_i}^{\text{radial}}[d, d + d^{\text{radial}}] = Z \sum_l \{ \mathbf{s}_k[l] - \hat{\mu}_{\mathbf{s}_k} \} \{ \mathbf{s}_i[l + d^{\text{radial}}] - \hat{\mu}_{\mathbf{s}_i} \}, \quad (15)$$

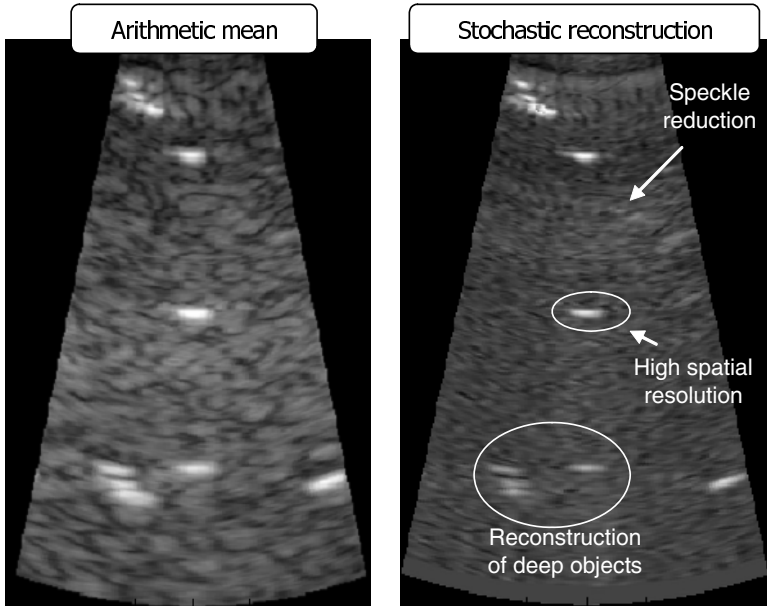
where  $0 \leq d < L$ ,  $l$  is a pixel index,  $d^{\text{radial}}$  is the radial lag between pixels,  $Z$  is a normalization factor,  $L$  is the block length and  $\hat{\mu}_{\mathbf{s}}$  is the estimated noise mean (this mean is estimated using the same data). This estimator of the covariance matrix is unbiased.

The estimated covariance functions of the selected blocks do not apply to the entire radial dimension. We still need to evaluate it in between (see the left side of Fig. 3). For this, we assume that between points in the FOV, the statistics change gradually. Hence, we can fill the missing data by interpolation. One can use interpolation methods of matrix-valued images [6], that preserve the semi-definiteness of the covariance matrix.

## 4 Experiment

In the experiment, we used a commercial medical ultrasonic system, the GE Vivid 3. The electronic signal generated by this system is a square burst with duration of three half periods. The probes used are phased arrays by GE, named **3s** and **5s**. The algorithm was applied on data obtained from a tissue-mimicking phantom, so that controlled and repeatable data can be generated. Fat was placed on top of the phantom to demonstrate an attenuating layer. The acquired images are presented in Fig. 1. One image was acquired with a burst frequency of  $1.5\text{MHz}$  and the **3s** probe (referred to as low acoustic frequency image). The second image was acquired with a burst frequency of  $2.5\text{MHz}$  and the **5s** probe (referred to as high acoustic frequency image). As illustrated in Fig. 2 speckle appearance of the same tissue changes in different acoustic frequencies. Nevertheless, in the high acoustic frequency image, system noise is very significant. We have direct access to  $\mathbf{a}^{\text{RF}}$ , received in MHz from the medium. We then directly apply sampling, attenuation compensation, envelope detection and log operation.

The input for the algorithm is  $\mathbf{a}_k^{\text{log}}$ . The BLUE was applied based on the two images, as illustrated in Fig. 4. The stochastic reconstruction significantly reduces speckle noise, along with high spatial resolution and reconstruction of deep objects. A by-product of the stochastic reconstruction is system noise reduction, due to the weighted averaging of the images. The peak signal to noise



**Fig. 4.** Stochastic reconstruction vs. simple averaging. The stochastic reconstruction produces an image with speckle reduction, along with high spatial resolution and reconstruction of deep objects.

**Table 1.** In all depths, the PSNR obtained by stochastic reconstruction is higher than the PSNR obtained by arithmetic mean

Depth (cm)	Arithmetic mean	Stochastic reconstruction
6	66 : 1 (18dB)	117 : 1 (21dB)
8	48 : 1 (17dB)	73 : 1 (19dB)
10	78 : 1 (19dB)	124 : 1 (21dB)

ratio (PSNR) was calculated. The results are presented in Table 1. The stochastic reconstruction presents a higher PSNR in all depths.

## 5 Discussion

The method reduces noise of ultrasound images. It also exposes deep objects while it maintains high resolution and does not blur the object to achieve denoising. Our approach requires a fast acquisition of two or more acoustic frequencies. There exists enabling technology [15] allowing that.

Future research can focus on the acquisition process as well as on the processing. In particular, it is worth studying which acoustic frequencies are optimal in this paradigm. In addition, more advanced mathematical tools can be used. For

example, diffusion methods [14] and adaptive subdivision coupled to statistical estimation [11] may be useful to this frequency compounding approach.

We wish to thank Zvi Friedman and Yonina Eldar for useful discussions.

## References

1. I. Amir, N. M. Bilgutay, and V. L. Newhouse. Analysis and comparison of some frequency compounding algorithms for the reduction of ultrasonic clutter. *IEEE Trans. on Ultrasonics Ferroelectric and Frequency Control*, 4:402–411, 1986.
2. B. A. J. Angelsen. *Ultrasound Imaging Waves, Signals, and Signal Processing*. Emantec, Norway, 2000.
3. V. Aurich and J. Weule. Non-linear gaussian filters performing edge preserving diffusion. In *Proc. 17th DAGM Symposium*, pages 538–545, 1995.
4. G. Cincotti, G. Loi, , and M. Pappalardo. Frequency decomposition and compounding of ultrasound medical images with wavelet packets. *IEEE Trans. on Medical Imaging*, 20:764–771, 2001.
5. S. Gupta, R. C. Chauhan, and S. C. Sexana. Wavelet-based statistical approach for speckle reduction in medical ultrasound images. *Medical & Biological Engineering & Computing*, 42:189–192, 2004.
6. H. Hagen and J. Weickert. *Visualisation and Processing of Tensor Images*. Springer, Berlin, 2006.
7. A. K. Jain. *Fundamentals of Digital Image Processing*. Prentice-Hall, Englewood Cliffs, N.J., 1989.
8. S. M. Kay. *Fundamentals of Statistical Signal Processing. Estimation Theory*. Prentice-Hall, Englewood Cliffs, N.J., 1993.
9. T. Loupas, W. N. McDicken, and P. L. Allan. An adaptive weighted median filter for speckle suppression in medical ultrasonic images. *IEEE Trans. on Circuits and Systems*, 36:129–135, 1989.
10. P. A. Magnin, O. T. Von Ramm, and F. L. Thurstone. Frequency compounding for speckle contrast reduction in phased array images. *Ultrasonic Imaging*, 4:267–281, 1982.
11. J. Polzehl and V. G. Spokoiny. Adaptive weights smoothing with applications to image restoration. *J. R. Statist. Soc. B.*, 62:335–354, 2000.
12. R. F. Wagner, S. W. Smith, J. M. Sandrik, and H. Lopez. Statistics of speckle in ultrasound B-scans. *IEEE Trans. on Sonics and Ultrasonics*, 30:156–163, 1983.
13. W. F. Walker and G. E. Trahey. The application of K-space in pulse echo ultrasound. *IEEE Trans. on Ultrasonics, Ferroelectrics, and Freq. Control*, 45:541–558, 1998.
14. J. Weickert. A review of nonlinear diffusion filtering. *Lecture Notes in Computer Science*, 1252:3–28, 1997.
15. J. Xuecheng, I. Ladabaum, and T. K. Butrus. The microfabrication of capacitive ultrasonic transducers. *Journal of Microelectromechanical Systems*, 7, 1998.

# Exploiting Low-Level Image Segmentation for Object Recognition

Volker Roth and Björn Ommer

ETH Zurich, Institute of Computational Science  
Universität-Str. 6, CH-8092 Zurich  
{vroth, bjoern.ommer}@inf.ethz.ch

**Abstract.** A method for exploiting the information in low-level image segmentations for the purpose of object recognition is presented. The key idea is to use a whole ensemble of segmentations per image, computed on different random samples of image sites. Along the boundaries of those segmentations that are *stable* under the sampling process we extract strings of vectors that contain local image descriptors like shape, texture and intensities. Pairs of such strings are *aligned*, and based on the alignment scores a mixture model is trained which divides the segments in an image into fore- and background. Given such candidate foreground segments, we show that it is possible to build a state-of-the-art object recognition system that exhibits excellent performance on a standard benchmark database. This result shows that despite the inherent problems of low-level image segmentation in poor data conditions, segmentation can indeed be a valuable tool for object recognition in real-world images.

## 1 Introduction

The goal of image segmentation is the detection of meaningful structures from a cluttered scene. Most current segmentation techniques take a bottom-up approach, where local image properties such as feature similarity (brightness, texture, motion etc) are used to detect coherent units. Unfortunately, image segmentation becomes very difficult in poor data conditions like shadows, occlusions and noise. In such situations, the detected coherent units often do not coincide with our perception of objects in a scene.

The automatic detection and recognition of visual objects in images, on the other hand, has been among the prime objectives of computer vision for several decades. Large intra-category variations of appearances and instantiations within object classes turn learning category models into a key challenge. Therefore, common characteristics of an object class have to be captured while offering invariance with respect to variabilities or absence of these features. In principle, segmentation algorithms might help to solve the *object detection* task by partitioning the image into meaningful parts that might serve as the inputs of a classification system. Many papers on image segmentation contain statements of the form “*segmentation is an important preprocessing step for object recognition*”. Due to the above limitations, however, the practical usefulness of low-level segmentation algorithms for the purpose of object recognition is questionable and, indeed, the currently best approaches to object recognition do *not* employ low-level segmentation, see e.g. [1,2,3,4,5].

In order to circumvent the obvious problems of segmentation, it has been proposed to treat segmentation and recognition in an interleaved manner, e.g.[6]. Approaches of this kind typically mix bottom-up strategies with top-down elements based on back-propagating hypotheses about the objects down to the segmentation level. These methods seem to work well, if the initial segmentation is of “reasonable” quality, which is often the case if one considers *moving* objects in videos where the motion information supports the segmentation process. Good performance can also be achieved, if only a small number of classes is considered for which relatively strong initial object hypotheses can be build by including additional side information. For the task of detecting objects in still images, however, the recognition performance of these methods is still rather limited, particularly if there are many potential object classes.

Despite the generally poor quality of bottom-up segmentations in real-world images, we demonstrate that it is possible to exploit low-level segmentations for building a very powerful object recognition system. The key idea is to use not only one segmentation, but a whole *ensemble of segmentations* which often captures at least *parts* of the objects in a scene. Such partial matches of the objects boundaries can be successfully used for discriminating between foreground/background segments.

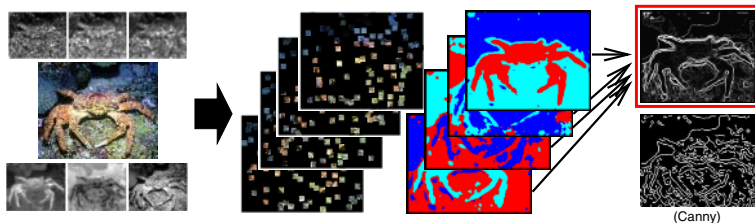
Given the candidate foreground segments, we show that it is often possible to recognize the object class. We propose two different approaches: the **direct approach** exclusively relies on the low-level segmentations by computing majority votes over all stable segments in an image, whereas the **combined approach** uses the predicted foreground segments as input of a hierarchical classification scheme. The latter learns to group parts of the image foreground segments into a hierarchy of category-specific compositions, and binds them together using a probabilistic shape model to recognize objects in scenes. The foundation for this approach is laid by the principle of *compositionality* [7]: As observed in cognition in general and especially in human vision, complex entities are perceived as compositions of comparably few, simple, and widely usable parts. Objects are represented based on their components and the relations between them. Composition models bridge the semantic gap between low level image features and high level scene categorizations [3,4] by establishing intermediate hidden layer representations. Our experiments with the *caltech 101* database [8] show that both the direct and the combined approach allow us to build a highly competitive object recognition system.

## 2 Ensembles of Low-Level Segmentations

For segmenting the images we use an adapted version of the algorithm proposed in [9] which combines both the ideas of *partitioning* and *feature combination/selection*. The latter aspect turns out to be very important for finding good segmentations, since segment-specific information is often spread over different cues like color and texture. The core of this algorithm consists of a Gaussian mixture model with built-in *relevance detection* which automatically selects important features by maximizing a constrained likelihood criterion. In order to find reasonable settings for the free model parameters, we devise a resampling-based model selection strategy which follows largely [10,9]. The key idea is to draw *resamples* of the object set, to train the segmentation model

on the individual resamples and to compare the resulting solutions. Adapted to our image segmentation problem, this strategy translates into sampling different *image sites*, inferring a segmentation on the basis of these sites and identifying *stable* segmentations (i.e. those which can be reproduced on many different random samples of image sites). We repeat this procedure for different numbers of mixture modes, and finally we receive a *stability-ranked* list of prototypical segmentations, see [9] for further details.

In addition to selecting these stable segmentations, we also overlay all individual segmentations to compute a *probabilistic boundary map* that encodes for each pixel its probability of being part of a segment boundary, see figure 1 for a schematic overview. Despite the fact that many individual segmentations are often of rather poor quality, the ensemble approach has two important advantages: (i) Within the subgroup of *stable* segmentations we often find relatively good partitions; (ii) the aggregated boundary map typically captures many details of the object in the image. To highlight the latter issue, we have additionally plotted the response of a Canny edge-detector in the right panel of figure 1. Due to the local character of the edge detection process, the Canny edges are much more noisy than the aggregated segment boundaries.



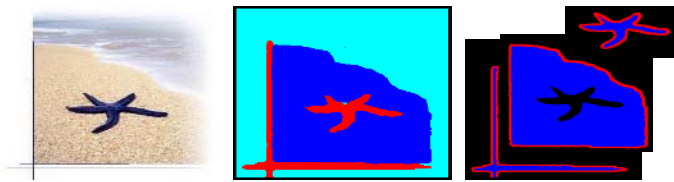
**Fig. 1.** Ensembles of segmentations. Left: input image and extracted features (top: three texture channels, bottom: LUV color channels). Middle: resampled image sites and corresponding segmentations. Top right: probabilistic boundary map found by overlaying all individual segment boundaries. Bottom right: Canny-edges for comparison.

### 3 Foreground/Background Learning

In the following we assume that we are given a set of training images with *category labels*. Additional information about the location of the objects, however, is not available. We further assume that there exists a *background* category with images that belong to none of the categories. Such a situation is e.g. given for the popular *caltech 101 dataset* [8] that contains images from 101 categories. For all experiments we used the images from 20 categories in *caltech 101*: *anchor, umbrella, barrel, trilobite, wrench, windsor\_chair, tick, stapler, electric\_guitar, gramophone, stop\_sign, cup, lobster, crayfish, water\_lilly, crab, starfish, wild\_cat, pyramid, pagoda*. This choice was guided by two criteria: we wanted a subset that is reasonably small ( $\approx 1000$  images) to explore a new method and that is sufficiently difficult to reliably evaluate the performance. The chosen categories are a mixture of artificial and natural object classes and they contain some classes that are very difficult to separate like *lobster* and *crayfish*. From all classes we randomly pick a *training set* of 25 images each. The remaining images are exclusively used for performance evaluation.

Based on ensembles of segmentations, we now introduce a method for identifying foreground segments. This foreground learning takes place in a *pairwise setting*. We first randomly pick two categories. For all training images belonging to these two categories, we consider all segmentations which exceed a certain stability threshold (see section 2) and we extract the boundary of each connected component. On regularly spaced points along these curves, vectors of *local image descriptors* are extracted. Thus, each connected segment is represented as a *string of vectors*. The same procedure is applied to the training images in the background class. Putting all such strings together we obtain a dataset consisting of  $n$  boundary strings from two categories and the background class. We then compute *local string alignments* for all pairs of these  $n$  strings. The final  $(n \times n)$  matrix of alignment scores is transformed into a valid *Mercer kernel*. In order to discriminate between fore- and background segments, we learn a *Gaussian mixture model* with three modes on these data which are represented by the kernel matrix. The estimated membership probabilities in one of the modes are used for identifying foreground segments: those segments that have a high probability for the *correct* image category are treated as foreground areas.

**Boundary extraction and string representation.** After the segmentation process, each pixel in an image has a group label. In a first step, *connected* pixels which share the same group label are extracted. For simplicity, we will refer to such connected groups of pixels as *segments* in the sequel. For each of these segments, we compute a chain-code representation of the segment boundary. We call such a boundary *closed* if the segment is entirely contained in the image, i.e. if it does not hit the image borders. For such closed segments the boundary chain is extended to two full circulations, which guarantees us that the alignment score between two such segments becomes independent of the starting point (note that we use *local* alignments). If a segment is not closed, we start at the image border and continue the chain until the border is hit again. Figure 2 depicts examples of such segment boundaries.

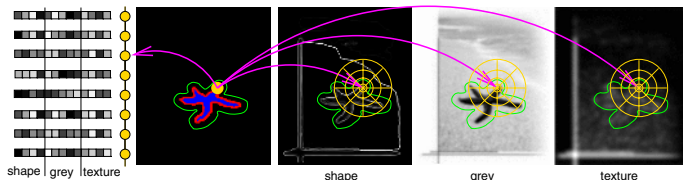


**Fig. 2.** Boundary extraction. Left: original image; middle: most stable segmentation; right: three extracted segments (blue) and their boundaries (red).

On regular intervals along the segment boundaries, we then extract a *vector of image descriptors*. The components of such a vector contain three different descriptor types: a *shape context* histogram, a *texture patch* and a *gray-value patch*. The shape context descriptor [11] consists of a log-polar histogram with 60 bins (10 angles, 6 scales) which is centered at the current position along the boundary. Each bin represents the (weighted) sum of those pixels in the map of aggregated segment boundaries which fall into the bin-specific part of the polar histogram and which are “close” to the segment, i.e. which lie in a close vicinity of the segment, see the green tube around the segment



in figure 3. The texture- and gray-value patches consist of locally averaged values of Gabor filter responses and image intensities respectively. In analogy to the shape context descriptor, a polar grid is used for defining the areas over which the averaging takes place. This polar geometry has the advantage that we can easily incorporate rotation invariance into the alignment process by simply shifting the indices of the descriptors.



**Fig. 3.** String representation of segments. Left to right: chain of vectors containing local image descriptors (schematic), segment boundary (red) and vicinity around the segment (green), polar histogram of the shape context descriptor (yellow), polar gray-value patch, polar texture patch.

**String alignments.** Having extracted a string for each of the  $n$  segments, we then compute the  $n \times n$  matrix of pairwise local alignments by way of the Smith-Waterman algorithm [12]. Contrary to the typical setting in which this algorithm is used, in our case we do not have a fixed alphabet of symbols for which a predefined scoring table for aligning pairs of these symbols is available. We rather have “strings” which are ordered collections of vectors with real-valued entries. Instead of looking up the symbol-wise scores in a table, in each step of the algorithm we evaluate a scoring function for two vectors. The components of these vectors consist of 60 bins of a shape context histogram, 60 locally averaged texture measurements and 60 locally averaged gray-values. Thus, a vector is composed of three subvectors  $v = (v^{\text{shape}}, v^{\text{text}}, v^{\text{gray}})^T$

In the experiments below we use a simple aggregation of these three cues that combines  $\chi^2$  distances between shape context histograms with correlation scores for texture and intensity patches: the scoring function for two vectors  $v_1, v_2$  has the form

$$s(v_1, v_2) = a - b \cdot \left( D_{\chi^2}(v_1^{\text{shape}}, v_2^{\text{shape}}) + D_{cc}(v_1^{\text{text}}, v_2^{\text{text}}) + D_{cc}(v_1^{\text{gray}}, v_2^{\text{gray}}) \right), \quad (1)$$

with the  $\chi^2$  distance  $D_{\chi^2}(v_1, v_2)$  and the cross correlation distance  $D_{cc}(v_1, v_2) = 1 - |cor(v_1, v_2)|$ , with  $cor(v_1, v_2)$  being the correlation between the vectors  $v_1$  and  $v_2$ . Note that distances are transformed into similarities, so that a high score means that two strings are similar. The constants  $a = 1/2, b = 1/3$  were selected empirically.

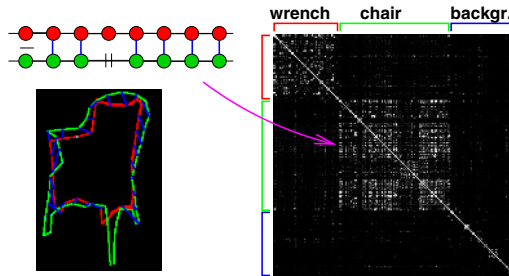
Since the extracted segments often capture only *parts* of the objects, the alignment scores are divided by the length of the alignment. In order to avoid high scores for very short “random” alignments, we consider such length-normalized alignments as “valid” only if the total alignment length exceed a certain threshold. In our experiments we require that two strings must be aligned at more that 15 consecutive positions, otherwise the score is down-weighted by a factor of ten. For a better geometric interpretation, we have depicted such positions which align to each other in figure 4 below as blue lines.

To further decrease the sensitivity to local segmentation errors, we allow *gaps* in the alignments. Such gaps are penalized by a predefined cost value  $g$ . In our experiments we

use  $g = 0.1$  which means that the current alignment score is decreased by 0.1 whenever a position in one string is aligned with a gap in the other. For two strings  $x, y$  with lengths  $l, l'$  the alignment algorithm recursively fills the  $(l \times l')$  matrix  $F$ :

$$F(i, j) = \max\{0, F(i-1, j-1) + s(x_i, y_j), F(i-1, j) - g, F(i, j-1) - g\}. \quad (2)$$

*Backtracking* from the highest value in  $F$  yields the optimal alignment, see [12] for details. Recall that the  $i$ -th position of string  $x$  is a shape/texture/intensity-vector, and that  $s(\cdot, \cdot)$  denotes the scoring function defined in (1). An example alignment matrix for the categories “wrench”, “windsor\_chair” and “background” is depicted in the right panel of figure 4, which shows a distinct block structure.



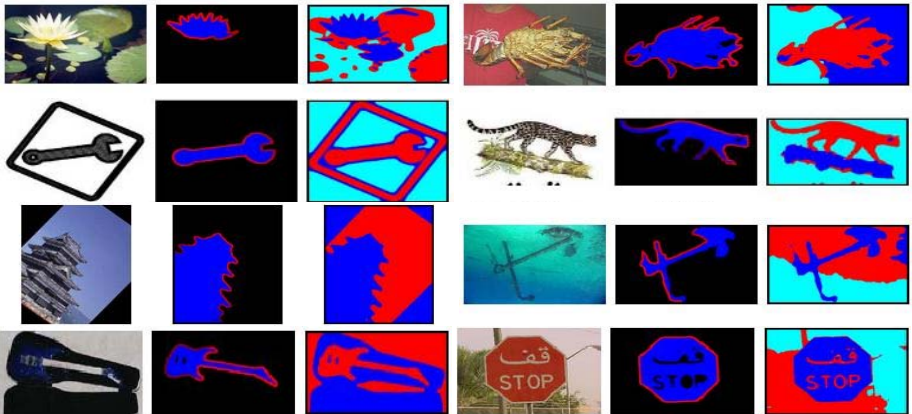
**Fig. 4.** String alignments. Left: alignment of two boundary strings (top: schematic, bottom: 2 segments from the “windsor\_chair” category). The blue lines connect aligned vectors. Right: pairwise alignment matrix for segments from the categories “wrench”, “windsor\_chair” and “background”.

**Detecting foreground segments.** In the final step in our foreground-detection process a *Gaussian mixture model* is learned for the  $n$  segments. These segments are represented in form of a  $(n \times n)$  matrix  $K$  of pairwise alignment scores. If this matrix would be positive semidefinite we could identify it as a *Mercer kernel* and train a mixture model in the kernel-induced space as proposed e.g. in [13]. It is well known that probabilistic alignment models such as *pair hidden Markov models* produce scores which fulfill the requirements of a valid Mercer kernel. For simplicity, however, we used a deterministic alignment model which might violate the positive-semi-definiteness condition. Moreover, the length-normalization of scores can lead to additional negative eigenvalues of  $K$ . In practice, however, we observe that there are typically only very few negative eigenvalues which are all of small magnitude. In order to transform it into a valid Mercer kernel, we use the *kernel PCA* idea [14] to find a decomposition  $K = V\Lambda V^T$  with a diagonal matrix  $\Lambda$  of eigenvalues. Discarding all negative eigenvalues we form a valid kernel  $K' = V_+ \Lambda_+ V_+^T$ .

Based on this kernel matrix  $K$  we now learn a Gaussian mixture model with 3 mixture modes. For initialization we label all segments in an image according to its category label, despite the fact that some segments might belong to the background class. During further iterations of the EM algorithm (see [13] for details), we re-estimate these membership probabilities in one of the three classes (two categories + background) for each segment. It is interesting that the selection of foreground segments does not vary

significantly if *different pairs* of categories or if *more than two* categories are selected. Examples of detected foreground segments are depicted in figure 5.

To predict foreground segments in the *test images* we first reduce the size of the training set by extracting from each of the 25 training images per category only the *two highest scoring foreground segments*. Based on the string representations of these  $2 \cdot 25 \cdot 20$  segments (2 segments/image, 25 images/category, 20 categories), we compute a new pairwise alignment matrix of size  $1000 \times 1000$  which now represents *all* training images. Discarding negative eigenvalues we again arrive at a valid kernel matrix  $K' = V_+ \Lambda_+ V_+^T =: X X^T$  that allows us to form a vectorial representation of the segments as the rows of the matrix  $X$ . Based on this data matrix and the corresponding category labels of the training images we learn a probabilistic 20-class kernel classifier. In the experiments we used a multi-class variant of *nonlinear kernel discriminant analysis* described in [15], which allows us to *predict* foreground segments in test images.



**Fig. 5.** Detecting foreground segments. From left to right in triplets: image, detected foreground segment (the one with the highest probability), corresponding stable segmentation.

## 4 Object Recognition

In order to exploit the results of the foreground identification for the purpose of object recognition, we use the classifier that was learned on the basis of the training images as described above to predict foreground segments in the *test images*. For this purpose we align each segment in a test image with *all*  $n = 1000$  training segments. The resulting 1000-dimensional alignment vector is projected onto the set of eigenvectors  $V_+$  of  $K'$ . Appropriate scaling by the eigenvalues (see [14]) yields a vectorial representation  $x_t$  of the test segment, for which the classifier predicts a set of membership probabilities in each of the 20 image categories. Segments that can be clearly assigned to one of the categories (i.e. which have a high membership probability) are considered as *hypothetical foreground segments* in a test image.

These hypotheses are now used for predicting the category labels of the test images in two different ways: the **direct approach** computes a weighted majority vote over all

segments in a test image. When assigning each image the most probable category label, the average retrieval rate of the direct approach is 58.3%. Among the *two most probable* categories, we find the correct one in  $\approx 71\%$ , and among the *three most probable* in  $\approx 79\%$ . Taking into account that the direct approach only uses low-level segmentations and that for roughly 1/4 of all images it seems to be very difficult to find any good segmentations, these retrieval rates are amazingly high. For comparison: our reference implementation [4] (which currently is one of the best methods on the caltech 101 database) achieves an average retrieval rate of 61.8% when trained exclusively on these 20 categories. For analyzing the effect of using many segmentations per image (we used 100 in the experiments), we repeated the whole processing pipeline with only 5 segmentations per image. In this setting, the average retrieval rate drops down to 26% which effectively demonstrates the advantage of using large ensembles.

The **combined approach** uses the boundaries of the predicted foreground segments as input for a *compositionality-based* recognition system which implements a variant of the model in [4]. The segment boundary contours are first split into shorter subcurves before encoding them using localized feature histograms from [3]. Top-down grouping of segment boundaries yields compositions of curves with increased discriminative power compared to their original constituents. The conceptual idea is to group image parts not based on their similarity but based on the familiarity of their composition. Assume for the moment that groupings which are distinctive for categories have already been learned from the training data. The objective of top-down grouping is then to form a hierarchy of compositions by combining those constituents whose composition has highest category posterior. The goal is now to automatically learn and represent models for top-down grouping in the case of large numbers of object classes. We tackle this problem by first estimating category dependent co-occurrence statistics of foreground curve segments in the training images. Using this distribution, the curves are then grouped. The resulting compositions are used to update the previously estimated category dependent grouping statistics and to learn a *global shape model*. This shape

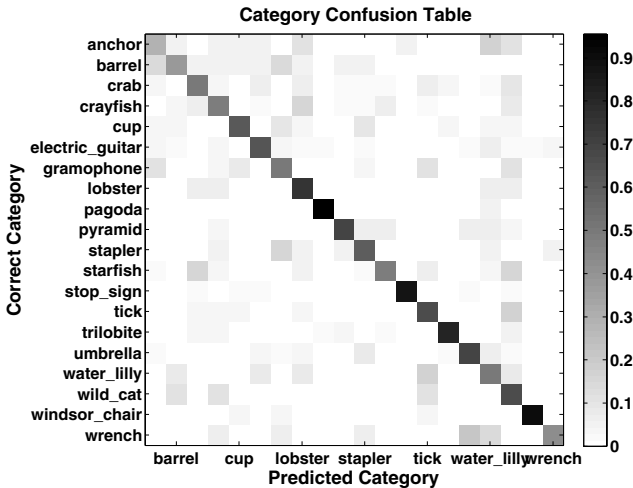


Fig. 6. Category confusion table of the compositional model with 62.3% retrieval rate

model is used for coupling all the established compositions in a final step. Due to limited space we refer the interested reader to [4] for details about the model and its practical implementation in form of a graphical model employing belief propagation.

Our first experiments with this combined approach yielded an average retrieval rate of 62.3% which is at least competitive to the reference model [4] (however, the increase in performance is probably not statistically significant). Figure 6 shows the corresponding *category confusion table*. This result shows that despite the difficulties of low-level segmentation, it is possible to exploit the information contained in ensembles of segmentations for building state-of-the-art recognition systems.

## 5 Discussion

Despite the fact that bottom-up image segmentation is sometimes considered as an important preprocessing step for object recognition, the actual usefulness of such an approach in real-world recognition scenarios is still an ongoing debate. For real-world scenes it is often difficult to find segmentations of sufficiently high quality that would allow to reliably extract object-specific information like shape, color distribution, texture features etc. It is, thus, not surprising that the currently best object recognition systems do *not* use low-level segmentations.

In this work we show that it is indeed possible to build state-of-the-art recognition systems on the basis of low-level image segmentations. The key idea is to use not only one single segmentation per image, but ensembles of many segmentations that are computed on different random samples of image sites. Although most of the individual segmentations might be of rather poor quality, the combination of many segmentations helps to overcome problems induced by poor data conditions in two ways: (i) analyzing the variation of segmentation solutions under the sampling process, we can identify subsets of *stable* segmentations that in many cases are of much higher quality than single-segmentation solutions. (ii) Aggregating all segment boundaries, we build probabilistic boundary maps. Compared with standard edge-detectors, the aggregated segment boundaries often encode at least parts of the “true” objects in the image.

Segmentations that have been identified as *stable* are represented by local image descriptors along their boundaries. These descriptors encode *shape*, *intensities* and *texture* in the form of histograms of segment boundaries, gray-value patches and local Gabor filter responses. Based on this string representation, all stable segments are compared utilizing a string alignment algorithm. From the matrix of alignment scores, a Mercer kernel is derived on which a (kernelized) Gaussian mixture model is trained which is used to build hypotheses about foreground segments. The hypothetical foreground segments are then used for recognizing the objects in test images in two different ways: the *direct approach* exclusively relies on the low-level segmentation information by building weighted majority votes over all segments in an image. In the *combined approach*, the segment boundaries serve as inputs for a compositionality-based recognition system which aggregates curves (or parts thereof) to category-specific compositions.

On a 20-category subset of the *caltech 101* database we compare these two approaches with one of the currently best recognition systems which yields a retrieval rate of 61.8 % on the considered images. We observe that even the direct approach which

“naively” works on the segments without building any compositions achieves a very good performance of 58.2% (a “bag-of features” approach on the hypothetical foreground segments yields only 49%). First experiments with the combined approach even slightly outperform the base-line system (although not in a statistically significant way). As more important than the exact retrieval rates, however, we consider the following: (i) Low-level segmentations can indeed be used for building competitive object recognition systems for real-world images. (ii) The use of large ensembles of segmentations is essential (otherwise the performance drops down significantly). (iii) A comparison with the performance of the method from [4] indicates that the segmentation process concentrates relevant image information in few boundary curves and mainly discards non-discriminative image regions. (iv) We believe that both the direct- and the combined approach can be substantially improved by systematically searching for advanced local image descriptors and improved scoring functions in the alignment process.

**Acknowledgments.** This work was supported in part by the Swiss national fund under contract no. 200021-107636.

## References

1. Agarwal, S., Awan, A., Roth, D.: Learning to detect objects in images via a sparse, part-based representation. *IEEE Trans. Pattern Anal. Machine Intell.* **26**(11) (2004)
2. Leibe, B., Schiele, B.: Scale-invariant object categorization using a scale-adaptive mean-shift search. In: *DAGM-Symposium*. (2004) 145–153
3. Ommer, B., Buhmann, J.M.: Object categorization by compositional graphical models. In: *EMMCVPR*. (2005) 235–250
4. Ommer, B., Buhmann, J.M.: Learning compositional categorization models. In: *ECCV*, Springer (2006) 316–329
5. Berg, A.C., Berg, T.L., Malik, J.: Shape matching and object recognition using low distortion correspondence. In: *CVPR*. (2005) 26–33
6. Yu, S.X., Gross, R., Shi, J.: Concurrent object recognition and segmentation by graph partitioning. In: *NIPS*, MIT Press (2002) 1383–1390
7. Geman, S., Potter, D.F., Chi, Z.: *Composition Systems*. Technical report, Division of Applied Mathematics, Brown University, Providence, RI (1998)
8. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In: *CVPR Workshop GMBV*. (2004)
9. Roth, V., Lange, T.: Adaptive feature selection in image segmentation. In: *Pattern Recognition–DAGM’04*, Springer (2004) 9–17
10. Lange, T., Roth, V., Braun, M.L., Buhmann, J.M.: Stability-based validation of clustering solutions. *Neural Computation* **16**(6) (2004) 1299 – 1323
11. Belongie, S., Malik, J., Puzicha, J.: Matching shapes. In: *ICCV*. (2001) 454–463
12. Smith, T.F., Waterman, M.S.: Identification of common molecular subsequences. *Journal of Molecular Biology* **147** (1981) 195–197
13. Roth, V., Steinhage, V.: Nonlinear discriminant analysis using kernel functions. In Solla, S., Leen, T., Müller, K.R., eds.: *NIPS 12*, MIT Press (1999) 568–574
14. Schölkopf, B., Smola, A., Müller, K.R.: Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation* **10**(5) (1998) 1299–1319
15. Roth, V., Tsuda, K.: Pairwise coupling for machine recognition of hand-printed japanese characters. In: *CVPR*. (2001) 1120–1125

# Wavelet Based Noise Reduction by Identification of Correlations

Anja Borsdorf, Rainer Raupach, and Joachim Hornegger

Institute of Pattern Recognition, Friedrich-Alexander-University Erlangen-Nuremberg  
Siemens Medical Solutions, Forchheim

**Abstract.** In this paper we present a novel wavelet based method for edge preserving noise reduction. In contrast to most common methods, the algorithm introduced here does not work on single input data. It takes two or more spatially identical images, which are both impaired by noise. Assuming the statistical independence of noise in the different images, correlation computations can be used in order to preserve structures while reducing noise. Different methods for correlation analysis have been investigated, on the one hand based directly on the original input images and on the other hand taking into account the wavelet representation of the input data. The presented approach proves to be suited for the application in computed tomography, where high noise reduction rates of approximately 50% can be achieved without loss of structure information.

## 1 Introduction

Particularly in diagnostic imaging, data contains noise predominantly caused by quantum statistics. A common problem in image processing, therefore, is the reduction of this pixel noise. Several approaches for edge-preserving noise reduction are known. The goal of all of these methods is to lower the noise power without averaging across edges. Some popular examples are nonlinear diffusion filtering [1] and bilateral filtering [2], which directly work in the spatial domain. Additional approaches exist that reduce noise based on the frequency representation of the input data, in particular wavelet-domain denoising techniques. Most of these algorithms are based on the observation that information and white noise can be separated using an orthogonal basis in the wavelet domain, as described e.g. in [3]. Structures (such as edges) are located in a small number of dominant coefficients, while white noise, which is invariant to orthogonal transformations and remains white noise in the wavelet domain, is spread across a range of small coefficients. This observation dates back to the work of Donoho and Johnstone [4]. Using this knowledge, thresholding methods were introduced, which erase insignificant coefficients but preserve those with larger values. Several techniques have been developed to further improve the detection of edges and relevant image content, for instance by comparing the detail coefficients at adjacent scales [5,6]. Most denoising methods based on wavelets suffer from the limitation that they are only applicable to white noise. A more robust algorithm which adapts itself to several types of noise is for instance presented in [7].

Nevertheless, most existing methods for noise reduction work on single image data and their ability to distinguish between information and noise, therefore, strongly depends on the size and the contrast of image structures. In contrast, if two or more images

are available, which show the same information but statistically independent noise, the differentiation between signal and noise can be further improved [8]. By comparing the input images either in spatial domain or on the basis of the wavelet coefficients, frequency dependent weighting factors can be computed. These weighting factors are then used to eliminate noise, whilst maintaining structural information in the wavelet representation of the images. Reconstruction of the modified wavelet coefficients yields an image with suppressed noise but including all structures detected as correlations between the input images.

This paper is structured as follows: After summarizing the basic concepts of the wavelet transformation in section 2 the noise reduction algorithm is introduced in detail in section 3. In section 4 the achieved results for the specific applications in computed tomography and fluoroscopy are presented.

## 2 Wavelet Transformation

Wavelets are generated from a single basis function  $\psi(t)$  called mother wavelet by means of scaling and translation:

$$\psi_{s,\tau}(t) = \frac{1}{\sqrt{|s|}} \psi\left(\frac{t-\tau}{s}\right); \quad s, \tau \in R, s \neq 0, \quad (1)$$

where  $s$  is the scaling parameter and  $\tau$  is the translation parameter. Wavelets must have zero mean and have bandpass like spectrum. For the computation of the discrete wavelet transformation only discrete pairs of  $s$  and  $\tau$  are used. Taking the discrete parameters

$$s_j = 2^{-j} \quad \text{and} \quad \tau_k = k \cdot s_j = k \cdot 2^{-j}; \quad j, k \in N^0, \quad (2)$$

where  $k$  is the translation and  $j$  the scale index, results in a dyadic sampling. Using these parameters a family of wavelets, spanning the  $L^2(R)$  can be derived from a mother wavelet  $\psi(t)$  as follows:

$$\psi_{j,k}(t) = \sqrt{2^j} \psi(2^j t - k). \quad (3)$$

The discrete wavelet transform (DWT) of a 1D function  $f(t)$  can then be computed by projecting the function onto the set of wavelets:

$$c_{j,k} = \int_{-\infty}^{\infty} f(t) \psi_{j,k}^*(t) dt, \quad (4)$$

where  $\psi_{j,k}^*(t)$  is the complex conjugate of  $\psi_{j,k}(t)$ .

The algorithm introduced by Mallat [9], allows a fast computation of the discrete dyadic wavelet transformation. The wavelet coefficients are computed by iteratively decomposing the signal into its high-pass filtered details and low-pass filtered approximation, reducing the resolution of the signal in each iteration by a factor of two. It can be shown that the discrete dyadic wavelet decomposition can be computed by an iterated filter bank (see [10] for details).

When dealing with images the two-dimensional wavelet transformation needs to be used. The one-dimensional transformation can be applied to the rows and the columns

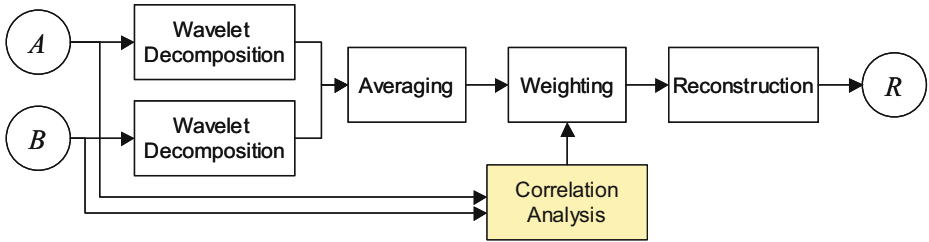


in succession, which is referred to as a separable two-dimensional wavelet transformation. After this decomposition four two-dimensional blocks of coefficients are available, including on the one hand the lowpass filtered approximation image  $C$  and three detail images  $D^x$ ,  $D^y$ ,  $D^{xy}$ . Analogously to the 1D case, the multiresolution wavelet decomposition can be computed iteratively from the approximation coefficients.

At every decomposition level, the detail images include high frequency structure information in horizontal, vertical and diagonal direction together with noise in the respective frequency band. Goal of the noise suppression method is to detect those detail coefficients which represent structure information. These coefficients should be kept unchanged, while coefficients, which are due to noise should be eliminated or at least be suppressed.

### 3 Filtering Algorithm

Fig. 1 shows a brief overview of the different steps used for the noise reduction algorithm. Although the algorithm can also be extended to work with more than two input images, without loss of generality, only the case of two images will be considered in the following.



**Fig. 1.** Overview of the noise reduction method

The two input images  $A$  and  $B$  are both decomposed into multiple frequency bands by a 2D discrete dyadic wavelet transformation. Of course, for the reduction of high frequency noise only those decomposition levels covering the frequency bands of the noise spectrum are of interest. Therefore, it is not necessary to compute the wavelet decomposition up to the coarsest scale. In our experiments, two to four decomposition levels were sufficient. For each decomposition level a similarity matrix is computed based on correlation analysis. The frequency dependent local discrepancy measurement can be based directly on the comparison of the original input images or on the wavelet representation of the input images. By the application of a predefined weighting function to the computed similarity values a level dependent weighting factor is computed. The resulting mask should preferably include ones in regions where structure information has been detected and values smaller than one elsewhere. The averaged wavelet coefficients of the input images, i.e. the detail coefficients, can then be weighted according to this mask. Averaging in the wavelet domain allows the computation of just one inverse wavelet transformation in order to get a noise suppressed result image  $R$ .

### 3.1 Correlation Analysis

Goal of the correlation analysis is to estimate the probability of a coefficient to correspond to structural information. This estimate is based on the measurement of the local homology of the input images. In the following, three different methods of similarity computation will be introduced, measuring the similarity based on the original input images, secondly based on the approximation coefficients and thirdly directly from the detail coefficients. The core idea behind all of these methods is similar: For the three blocks of detail coefficients  $D_l^x, D_l^y, D_l^{xy}$  of the wavelet decomposition, including horizontal, vertical and diagonal details, a corresponding similarity matrix  $S_l$  is computed for every level  $l$  up to the maximum decomposition level. Then, according to the defined weighting function the detail coefficients are weighted with respect to their corresponding values in the similarity matrix.

**Correlation Coefficient Based Methods:** One popular method for measuring the similarity of noisy data is the computation of the empirical correlation coefficient [11]:

$$r_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}}, \quad (5)$$

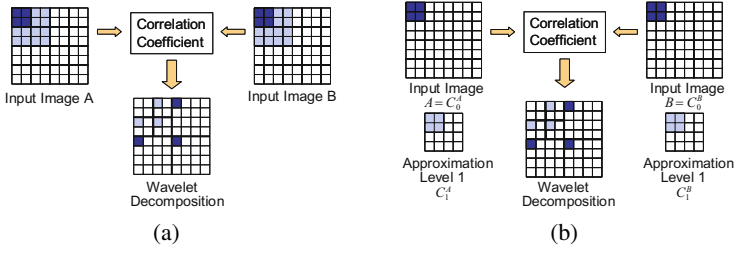
where  $x = x_1, x_2, \dots, x_n$  and  $y = y_1, y_2, \dots, y_n$  are two sequences of data each with  $n$  data points. The mean values of  $x_i$  and  $y_i$  are denoted as  $\bar{x}$  and  $\bar{y}$ . The empirical correlation coefficient also known as *Pearson's correlation* is independent from both origin and scale and takes values out of the interval  $[-1; 1]$ , whereas one means perfect correlation, zero no correlation and minus one perfect anticorrelation.

This correlation coefficient can now be used in order to compute the local homology between the input images, by taking blocks of pixels out of the two images and computing the correlation coefficient (see Fig.2(a)). Of course, the pixels used for similarity measurement at a respective position should be closely associated with the particular detail coefficient. This should later on be weighted according to the computed similarity value. Preferably all pixels from the original input image, which influenced the detail coefficient at the current position and scale, through the computation of the wavelet decomposition, should be incorporated into the similarity computations. It is clear that with increasing decomposition level the size of the pixel regions in the original image must also increase. Additionally, it is necessary to take the length  $m$  of the wavelet filters into consideration. Altogether, the number of pixels  $n_l$  of the original image influencing a coefficient at level  $l$  can be computed iteratively according to:

$$n_l = 2 \cdot n_{l-1} + m - 2; \quad \text{with } n_1 = m. \quad (6)$$

With this size adaptation the application of the algorithm in combination with arbitrary wavelets, which can be represented by FIR lowpass and highpass filters, becomes possible. However, the computational costs are quite high, because of the increasing size of the pixel regions in dependence on the decomposition level.

Improved results can be achieved with respect to performance as well as image quality if the correlation computations are not based on the original input images but on the



**Fig. 2.** Correlation computations based on correlation coefficients - (a) based on the original input images, (b) based on the approximation images of the previous decomposition level

approximation images. The multiresolution wavelet decomposition is computed iteratively. Thus the detail coefficients at level  $l$  are gained from the approximation image of the previous decomposition level. A very close connection between the detail coefficients and the computed similarity values can be obtained if the pixel regions are also taken from the approximation images of the previous decomposition level. The advantage of this approach is that the size of the pixel regions no longer depends on the decomposition level (see Fig.2(b)). Only the length of the wavelet filters needs to be considered. The disadvantage is that the approximations at all scales need to be stored for this method, although only the approximations of the maximum decomposition level are needed for perfect reconstruction.

Both of the methods mentioned so far have to deal with the same problem. The image regions are adjusted to the length of the filter used for analysis, but not to the coefficients of the filter. All intensity values within the considered pixel region are weighted equally. The result is that edges of higher contrast dominate the correlation values, as long as they occur within the region covered by the filter. However, if the filter coefficients should be taken into consideration all three blocks of detail coefficients must be treated separately, because the corresponding 2D filters are different. A third alternative method, where the similarity is directly computed from the detail coefficients circumvents this problem.

**Gradient Approximation:** The core idea behind the similarity measurement based on detail coefficients at level  $l$  is to use the fact that horizontal and vertical detail coefficients  $D_l^x$  and  $D_l^y$  can be regarded as approximations of the partial derivatives of the approximation image  $C_{l-1}$ . Coefficients in  $D_l^x$  show high values at positions where high frequencies in  $x$ -direction are present and  $D_l^y$  where sudden changes in contrast in  $y$ -direction can be found. If these two aspects are considered together, we get an approximation of the gradient field of  $C_{l-1}$ :

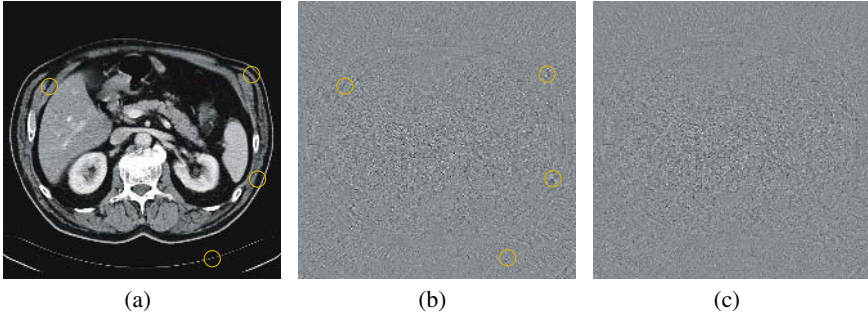
$$\nabla C_{l-1} = \begin{pmatrix} \partial C_{l-1} / \partial x \\ \partial C_{l-1} / \partial y \end{pmatrix} \approx \begin{pmatrix} D_l^x \\ D_l^y \end{pmatrix}. \quad (7)$$

The detail coefficients in  $x$ - and  $y$ -direction of both decompositions approximate the gradient vectors with respect to equation (7). The similarity can then be measured by

computing the angle between the corresponding gradient vectors [8]. The goal is to obtain a similarity value in the range of  $[-1, 1]$ , analogously to the correlation computations above. Therefore, we take the cosine of the angle resulting in:

$$S_l = \frac{D_l^{Ax} D_l^{Bx} + D_l^{Ay} D_l^{By}}{\sqrt{(D_l^{Ax})^2 + (D_l^{Ay})^2} \sqrt{(D_l^{Bx})^2 + (D_l^{By})^2}}, \quad (8)$$

where the superscript  $A$  refers to the first and  $B$  to the second input image. The gradient approximation method and the more time consuming computation of the correlation coefficients explained above are closely related. Nevertheless, the approaches are not identical and do not generally lead to the same results. The application of the algorithm for noise reduction based on the gradient approximation, as introduced so far, sometimes leads to visible artifacts in the resulting images. Fig. 3(b) and the difference image Fig. 3(b) give an example where this problem can be seen in case of using the Haar wavelet. Noticeably, the artifacts predominantly emerge where diagonal structures



**Fig. 3.** Artifacts due to weighting down correlated diagonal coefficients with gradient approximation method - (a) noise suppressed image with gradient approximation without separated treatment of diagonal coefficients, (b) difference image to average of input images, (c) difference image after special treatment of diagonal coefficients

appear in the image, and their shape generally enforces the assumption that diagonal coefficients at different decomposition levels are falsely weighted down. Reason for this is that diagonal patterns exist, which lead to vanishing detail coefficients in  $x$ - and  $y$ -direction. If the norm of one of the approximated gradient vectors is too small or even zero, no reliable information about the existence of correlated diagonal structures can be obtained from equation (8).

The simplest possibility for eliminating the artifacts is to weight only the detail coefficients  $D_l^x$  and  $D_l^y$  based on the similarity measurement  $S_l$  and leave the diagonal coefficients  $D_l^{xy}$  unchanged. Of course this avoids artifacts in the resulting images, but, unfortunately, noise included in the diagonal coefficients remains unchanged, leading to a lower signal-to-noise ratio for the denoised image. From equation (8), we can recognize that the similarity value is computed only with respect to  $D_l^x$  and  $D_l^y$ . The

diagonal coefficients do not influence the computation. However, the idea to extend the approximated gradient vector (see equation (7)) by the diagonal coefficients to a three dimensional vector does not lead to the desired improvements. In case of vanishing detail coefficients in  $x$ - and  $y$ - direction, no quantitative relation between the diagonal coefficients can be gained. Moreover, the extension of the approximated gradient vector by the diagonal coefficient is not a suitable solution. A diagonal coefficient can be interpreted as second order derivative and is therefore very sensitive to noise. Mixing it with the detail coefficients in  $x$ - and  $y$ -direction generally leads to less reliable similarity measurements.

In order to avoid artifacts while still reducing noise in the diagonal coefficients, only the detail coefficients  $D_l^x$  and  $D_l^y$  are weighted, depending on the similarity measurement computed from equation (8). The diagonal detail coefficients are treated separately. The weighting function for the diagonal coefficients is based on the correlation analysis between  $D_l^{Axy}$  and  $D_l^{Bxy}$ :

$$S_l^{xy} = \frac{2D_l^{Axy} D_l^{Bxy}}{\left(D_l^{Axy}\right)^2 + \left(D_l^{Bxy}\right)^2}. \quad (9)$$

Using this extension for separated weighting of the diagonal coefficients, denoising results are free of artifacts (see Fig.3(c)).

### 3.2 Weighting Function

The simplest possible method for weighting the coefficients is to use a thresholding approach. If the similarity value  $S_l$  at position  $(x, y)$  is above a defined value  $\tau_l$ , the detail coefficients are kept unchanged, otherwise they are set to zero [8]. The weighting function can be defined as

$$W_l(S_l(x, y)) = \begin{cases} 1 & \text{if } S_l(x, y) \geq \tau_l \\ 0 & \text{otherwise} \end{cases}. \quad (10)$$

However, the choice of an appropriate threshold very much depends on the noise level of the input images. Therefore, with increasing noise level in the input images the threshold should be set less strictly and with some tolerance. Preferably the threshold should be chosen level dependent, meaning that the threshold should be abated for higher decomposition levels. Generally the use of continuous weighing functions like

$$W_l^{powN}(S_l(x, y)) = \left(\frac{1}{2}(S_l(x, y) + 1)\right)^N \in [0, 1], \quad (11)$$

where no hard decision about the maintenance or the discarding of coefficients is required, leads to better results. The power  $N$  can also be chosen level adaptive.

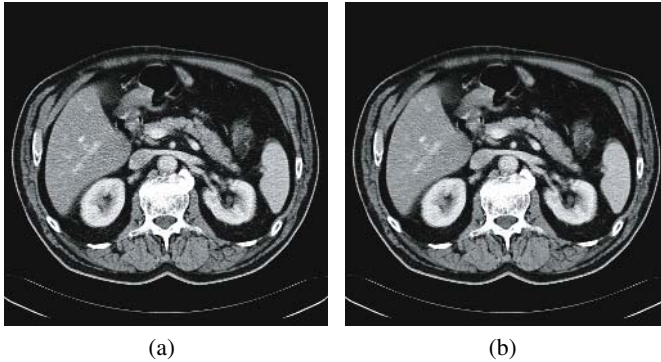
## 4 Experimental Evaluation

### 4.1 Computed Tomography

One important application of the noise reduction algorithm introduced above can be found in X-ray computed tomography (CT). In CT always a tradeoff between pixel

noise, dose of radiation and image resolution must be found. Reducing the dose of radiation for example by a factor of two increases the noise level in the images by a factor of  $\sqrt{2}$ . Goal of the application of the noise reduction algorithm to CT images is to achieve improved image quality without increasing the dose of radiation, or, the other way round, to reduce the dose of radiation without impairing image quality.

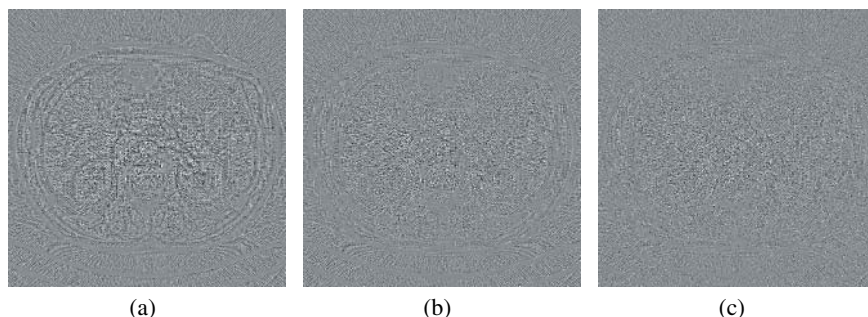
Spatially identical images with uncorrelated noise can be generated through separate reconstruction from disjoint sets of projections. For example two images can be reconstructed, each using only every second projection. Specifically, one image is computed from the even, and the other one from the odd numbered projections. Due to the reconstruction with only half of the projections, the noise level of the two generated images increases by a factor of  $\sqrt{2}$ . By averaging the wavelet coefficients of the input images, the result image corresponds to the image reconstructed with the complete set of projections, where additionally noise is reduced. Usually, a loss of image resolution through splitting the projections into two halves can be obviated because the overall number of projections in CT can be assured to be high enough.



**Fig. 4.** Application of the noise reduction algorithm to CT images - (a) average of input images (standard deviation:  $\sigma \approx 52$  HU), (b) denoised result ( $\sigma \approx 25$  HU)

Fig.4(b) shows the noise suppressed result image in comparison to the average of the input images Fig.4(a). It can be seen clearly that especially in homogeneous image regions, as for example in the region of the liver, noise is reduced, while structures and also small details are preserved. For clinical tests we used two CT slices, one from the abdomen, the other from the thorax. For each slice the average image and nine different configurations of denoised images were computed (see [12], [13] for details). The nine noise suppressed images were compared to the average image by two radiologists independently. All images of the tests were unlabeled. The result was that the average of the input images has never been judged superior to the noise suppressed images. In average the pixel noise  $\sigma$  in the noise suppressed images has been reduced by 50% in comparison to the average of input images.

The different approaches for correlation analysis can be assessed by comparing the difference images, which are presented in Fig.5. It can be seen that the correlation

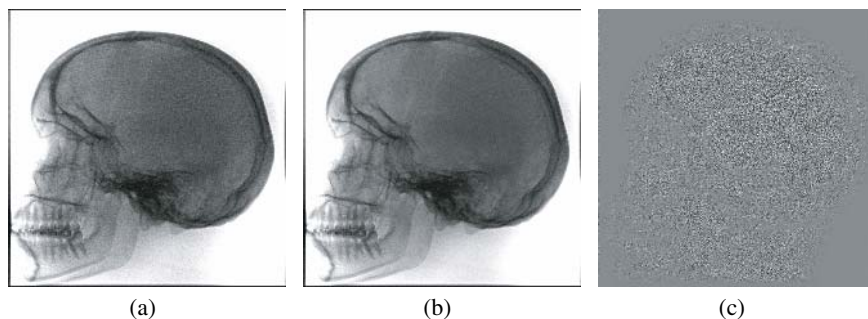


**Fig. 5.** Comparison of correlation analysis methods: difference images between result image and the average of input images - (a) CC based on original images, (b) CC based on approximation images, (c) gradient approximation

coefficient (CC) based method, where the pixel regions are taken from the original input images is less precise than the other two approaches, because structure information is also included in the difference images. The other two approaches lead to nearly the same good results. In regions of edges no noise is reduced, but the quality of the edge is kept unchanged. In the shown examples the Haar wavelet has been used. The experimental results with different wavelets showed, that especially biorthogonal spline wavelets, like the CDF9/7 wavelet [14], are well suited for the noise reduction algorithm.

## 4.2 Fluoroscopy

A second clinical application of the introduced method for noise reduction can be found in fluoroscopy, where sequences of x-ray projections are acquired. Therefore, achieving the maximum image quality with a minimum of radiation dose is required. In Fig.6 the initial experimental results achieved for x-ray images of a human skull are presented.



**Fig. 6.** Application of the noise reduction method to fluoroscopy images of the human skull - (a) average of input images, (b) denoised image, (c) difference image

## 5 Conclusion

We presented a novel edge-preserving wavelet based method for noise reduction. The algorithm works on two input images, which show the same information whereas the noise between the input images is uncorrelated. Using this property, correlation computations can be used in order to differentiate between structures and noise. Three different approaches of correlation analysis have been discussed. Especially the gradient approximation approach with the introduced separated treatment of the diagonal wavelet coefficients allows an artifact free and computationally efficient noise suppression. The application of the algorithm to computed tomography images showed that a noise reduction of approximately 50% is possible without loss of structure information. Even fine edges and small structures are preserved. For the application to fluoroscopy images it must be assured that the patient does not move. Otherwise the method must be used in combination with image registration algorithms.

## References

1. Catte, F., Lions, P.L., Morel, J.M., Coll, T.: Image selective smoothing and edge-detection by nonlinear diffusion. *SIAM Journal on Numerical Analysis* **29** (1992) 182–193
2. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: *IEEE International Conference on Computer Vision, Bombay, India (1998)* 839–846 <http://www.cse.ucsc.edu/~manduchi/Papers/ICCV98.pdf>.
3. Hubbard, B.: *Wavelets: Die Mathematik der kleinen Wellen*. Birkh"auser Verlag, Basel, Schweiz (1997)
4. Donoho, D.L., Johnstone, I.M.: Ideal spatial adaptation by wavelet shrinkage. *Biometrika* **81** (1994) 425–455
5. Xu, Y., Weaver, J., Healy, D., Lu, J.: Wavelet transform domain filters: A spatially selective noise filtration technique. *IEEE Transactions on Image Processing* **3** (1994) 747–758
6. Faghih, F., Smith, M.: Combining spatial and scale-space techniques for edge detection to provide a spatially adaptive wavelet-based noise filtering algorithm. *IEEE Transactions on Image Processing* **11** (2002) 1062–1071
7. Pizurica, A., Philips, W., Lemahieu, I., Acheroy, M.: A versatile wavelet domain noise filtration technique for medical imaging. *IEEE Transactions on Image Processing* **22** (2003) 1062–1071
8. Hoeschen, C., Buhr, E., Tischenko, O.: Verfahren zur Reduktion von Rauschstrukturen in Arrays von Pixelwerten (2004) *Offenlegungsschrift DE10305221A1* 2004.08.26.
9. Mallat, S.: A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Maschine Intelligence* **11** (1989) 674–693
10. Strang, G., Nguyen, T.: *Wavelets and Filter Banks*. Wellesley- Cambridge Press (1996)
11. Weisstein, E.: Correlation coefficient (2005) <http://mathworld.wolfram.com/CorrelationCoefficient.html>.
12. Borsdorf, A.: Noise reduction in CT images by identification of correlations (2005)
13. Borsdorf, A., Raupach, R., Hornegger, J.: Reduction of quantum noise in CT-images by identification of correlations. In: *Proceedings of the 2nd Russian-Bavarian Conference on Bio-Medical Engineering, Moskow (2006)*
14. Getreuer, P.: Filter coefficients to popular wavelets (2005) <http://www.mathworks.com>.



# Template Based Gibbs Probability Distributions for Texture Modeling and Segmentation

Dmitrij Schlesinger

Dresden University of Technology

**Abstract.** We present a new approach for texture modeling, which joins two ideas: well defined patterns used as "elementary texture elements" and statistical modeling based on Gibbs probability distributions. The developed model is useful for a wide range of textures. Within the scope of the method it is possible to pose such tasks as e.g. learning the parameters of the prior model, texture synthesis and texture segmentation in a very natural and good founded way. To solve these tasks we propose approximative schemes based on the Gibbs Sampler combined with the Expectation Maximization algorithm for learning. Preliminary experiments show good performance and accuracy of the method.

## 1 Introduction

The aim of this paper is to develop a texture model, which allows to join the following two ideas. The first one is built on assumptions, that:

- a given texture can be characterized by an "ideal pattern" (we will call it template), which itself can be understood as a kind of image;
- the template (as well as the image to be processed) can be spatially transformed in some well defined way;
- a textured image can be characterized by a distribution of transformed templates over its domain.

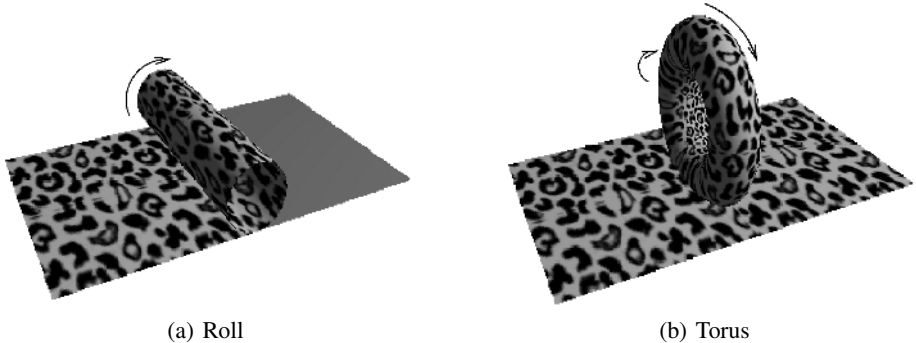
The second idea is to use a statistical framework for modeling. We argue for that, first of all because such modeling allows to pose tasks in a very reasonable and well founded way. Another reason is that statistical approaches are already widely used for texture modeling (see e.g. [5,6]), but very often without taking into account geometrical properties of a template (for instance only taking into account pairwise interactions between pixels or only using templates to compute features). The third reason is that we want to consider spatial transformations of images, that can be described by means of a displacement field. The latter can be very good modeled by Gibbs probability distributions (see e.g. [2]). And last but not least, we would like to apply our approach to texture segmentation. The segmentation task on its part can be posed (and often successfully solved) e.g. as a Bayes decision task for appropriate chosen statistical models based on Markov random fields (see e.g. [2,3,6]).

In this work we present a statistical texture model, which we call "Template based Gibbs probability distribution". In the next section we give a formal definition of the model followed by task formulations, appearing in that context. Based on the developed

model we formulate in Section 3 the task of unsupervised texture segmentation. In Section 4 we give some preliminary results. Finally, we conclude with a discussion about possible applications of the presented model and open questions.

## 2 The Model

We begin to explain the model by introducing an illustrative example. Typical images of a well defined texture are wallpapers. Let us consider the "production process" of wallpapers (see Fig. 1(a)). The image is produced by a roll, which is itself colored by a



**Fig. 1.** Producing of wallpapers

predefined pattern. Let us generalize this a little bit, because in our case the pattern on the roll should be periodic itself – i.e. the image should be periodic in two directions. This can be done by replacing the roll by a torus as shown in Fig. 1(b). Formally the torus can be understood as a "set of positions", equipped with a neighborhood structure. In our case a color value should be assigned in addition to each torus position – thus specifying a pattern on the torus. Loosely speaking, such a torus together with its pattern defines the painted texture. On the other hand, in order to define the produced image uniquely, some "rolling parameters" should be given in addition. For instance, a torus position can be specified for each pixel of the image. Such an assignment can be understood as a "complete interpretation of the image by means of a given pattern". In doing so, we account for deviations from the ideal rolling process. A "non ideal rolling process" can be imagined as if the underlying sheet of paper would be stretched a little bit during the rolling. Or equivalently, as if an ideal image would be spatially transformed in addition by applying an elastic transformation – e.g. a smooth displacement field. Therefore, the specification of the "rolling parameters" is not reducible to some small set of parameters like rolling speed, initial position of the torus on the image or similar. Summarized, a natural recognition task is to find the best interpretation (the corresponding torus position for each image pixel) given an image and a template (torus with a pattern). The learning task would be to estimate for example the pattern on the torus given an image.

Let us formalize all considerations made so far. We denote by  $R$  the set of image pixels, where  $r = (i, j) \in R$  is a particular pixel. Furthermore, we define a neighborhood

structure on the set of pixels by introducing the set of pairs  $E \subset \{\{r, r'\}\}$ . For instance 4-neighborhood can be used for this matter. We will understand the template as a set  $K$  of positions, equipped with its own neighborhood structure, which corresponds to a torus. We refer an element of the template as  $k = (i', j') \in K$  and call it state or label. The set  $\{(i', j') : i' = 1 \dots n, j' = 1 \dots m\}$  of positions can be considered alternatively as an orthogonal lattice, on which e.g. a usual 4-neighborhood structure is defined. In addition the rows with  $i' = 1$  and  $i' = n$  as well as the columns with  $j' = 1$  and  $j' = m$  are defined to be neighbors. A "complete interpretation" of the set  $R$  is a mapping (we will call it position labeling)  $f : R \rightarrow K$ , which assigns one element  $k \in K$  of the template to each pixel  $r \in R$ .

In the next step it is necessary to express in an appropriate manner, which labelings should be considered as "good" ones and which not. A simple way is to follow the principle: "two neighboring image pixels should be labeled by states, which are neighboring (or at least not far from each other) on the torus". We prefer statistical modeling and consider an a-priori probability distribution of labelings as a Gibbs probability distribution of second order as follows:

$$P(f) = \frac{1}{Z} \prod_{\{rr'\} \in E} g_{rr'}(f(r), f(r')) \quad (1)$$

with the normalizing constant  $Z$  and functions  $g_{rr'} : K \times K \rightarrow \mathbb{R}$ , which express our assumptions about the "goodness" of labelings, i.e. follow the principle mentioned above. In practice we use the following functions  $g_{rr'}$ :

$$g_{rr'}(k, k') = \exp \left[ -\frac{\text{dist}((k + \Delta k), k')}{\sigma_g} \right], \quad (2)$$

where  $\text{dist}(k_1, k_2)$  is the distance between two positions  $k_1$  and  $k_2$  (for instance the squared length of the shortest line connecting positions  $k_1$  and  $k_2$  on the torus). The label  $(k + \Delta k)$  represents the "best choice" for the neighboring node  $r'$  given a state  $k$  in the node  $r$ . For horizontal edges (i.e.  $r' = r + (0, 1)$ ) we use  $\Delta k = (0, 1)$ , for vertical ones (i.e.  $r' = r + (1, 0)$ ) holds  $\Delta k = (1, 0)$ .

The conditional probability distribution of observations (images) given a labeling is defined as follows. Let us denote an image as a mapping  $x : R \rightarrow V$ , transforming the set  $R$  of pixels into the set  $V$  of colors (grayvalues),  $x(r) = v \in V$  denotes the color in the pixel  $r$ . We assume conditionally independent probability distribution

$$P(x | f) = \prod_{r \in R} q(x(r), f(r)), \quad (3)$$

where the function  $q : V \times K \rightarrow \mathbb{R}$  is the probability distribution  $P(v | k)$  to observe the grayvalue  $v$  given the state  $k$ . At this point we would like to note, that all previous considerations were made under the assumption, that there exists an ideal pattern (painted on the torus). This can be easily generalized in the following way. Each position on the torus can be characterized by a *color distribution*, rather than by a single "ideal" color. As a *special case* one can consider e.g. a Gaussian probability distribution

$$q(v, k) = P(v | k) = \frac{1}{\sqrt{2\pi}\sigma_k} \exp \left[ -\frac{(v - \mu_k)^2}{\sigma_k^2} \right], \quad (4)$$

where  $\mu_k$  and  $\sigma_k$  are the parameters of the Gaussian, "attached" to the state  $k$ . In this case the mean value  $\mu_k$  can be interpreted as an "ideal" (most probable) color at the position  $k$  of the torus.

Summarized the joint probability distribution is defined by

$$P(x, f) = \frac{1}{Z} \prod_{\{rr'\} \in E} g_{rr'}(f(r), f(r')) \cdot \prod_{r \in R} q(x(r), f(r)). \quad (5)$$

Now we are ready to formulate the tasks we are interested in. We will distinguish two types of "basic" tasks, namely the recognition task and the learning task. Let us consider the first one: find the best position labeling  $f$  for a given image  $x$ . We formulate this problem as a Bayes decision task with respect to an additive cost function of the type  $C(f, f') = \sum_r c(f(r), f'(r))$ . This leads to the following decision strategy [7]. First, it is necessary to compute the marginal a-posteriori probabilities for the states

$$P(f(r) = k | x) = \sum_{f: f(r)=k} P(f | x), \quad \forall r \in R, k \in K. \quad (6)$$

Based on them the decision is obtained by solving for each pixel independently the local optimization problem

$$f^*(r) = \arg \min_k \sum_{k'} P(f(r) = k' | x) \cdot c(k, k'). \quad (7)$$

Let us consider the learning task, i.e. the task of template estimation given a textured image. We follow the Maximum Likelihood principle, that is we maximize the probability of observation (of the image) with respect to the unknown parameters, which are the conditional probability distributions  $q(v, k) = P(v | k)$  of grayvalues for each position  $k$ :

$$P(x; q) = \sum_f P(f) \cdot P(x | f; q) \rightarrow \max_q. \quad (8)$$

Since we can not perform the summation over all labelings we use the Expectation Maximization algorithm [1,9] for approximation. The standard approach gives<sup>1</sup> for general functions  $q$

$$q^{(n+1)}(v, k) \sim \sum_{r: x(r)=v} P(f(r) = k | x; q^{(n)}), \quad (9)$$

where  $P(f(r) = k | x; q^{(n)})$  are again the marginal a-posteriori probabilities of states in the  $n$ -th iteration. Consequently we need these probabilities for both the recognition task (7) and the learning task (8). Unfortunately, we do not know how to compute them in a closed fashion. They can be however estimated approximately using the Gibbs Sampler [4]. For details we refer to [2,3,8,10].

In practice it is not usefull to deal with functions  $q$  in general form, simply because the number of free parameters grows very quickly with the size of the template. One image is practically not enough to learn them all, especially taking into account, that

<sup>1</sup> Since the corresponding derivation is rather standard, we omit details here. Similar applications of the EM-algorithm with more details can be found e.g. in [3,8,10].

the learning can be performed only approximately using the EM-scheme and the Gibbs Sampler. Due to these reasons we use in practice Gaussian probability distributions (4) for  $q$ . In this case the update step of the EM-algorithm is

$$\begin{aligned}\mu_k^{(n+1)} &= \frac{\sum_r x(r) \cdot P(f(r) = k | x; \mu^{(n)}, \sigma^{(n)})}{\sum_r P(f(r) = k | x; \mu^{(n)}, \sigma^{(n)})}, \\ \sigma_k^{(n+1)} &= \sqrt{\frac{\sum_r (x(r) - \mu_k^{(n+1)})^2 \cdot P(f(r) = k | x; \mu^{(n)}, \sigma^{(n)})}{\sum_r P(f(r) = k | x; \mu^{(n)}, \sigma^{(n)})}}.\end{aligned}\quad (10)$$

### 3 Texture Segmentation

In this section we apply our model for texture segmentation. We imagine the generative model as follows. First, the segmentation is generated, i.e. a name (segment number) is assigned to each pixel. Second, the position labelings  $f$  are filled in for each segment separately using the basic model, considered in the previous section. At the last stage the image  $x$  is generated given the segmentation and the position labeling. Summarizing, we have to build a statistical model for triples "segmentation, position labeling, observation".

First of all, let us introduce an additional field  $s : R \rightarrow L$  called segmentation field, which maps the set  $R$  of nodes into the set  $L$  of segments. The a-priori probability distribution for pairs  $(s, f)$  is built by the following principle:

- If the segments  $s(r)$  and  $s(r')$  in two neighboring pixels  $r$  and  $r'$  are the same, the functions  $g_{rr'}$  (see (2)) from the basic model (5) are used.
- If the segments  $s(r)$  and  $s(r')$  in two neighboring pixels  $r$  and  $r'$  are different, a constant function  $g_{rr'} \equiv a$  is used, where the constant  $a$  represents the "penalty" for two neighboring pixels if they are assigned to different segments.

Consequently the a-priori probability distribution is defined as

$$P(s, f) = \frac{1}{Z} \prod_{\{r, r'\} \in E} \hat{g}_{rr'}(s(r), s(r'), f(r), f(r')), \quad (11)$$

where functions  $\hat{g}_{rr'}$  are

$$\hat{g}_{rr'}(l, l', k, k') = \begin{cases} g_{rr'}(k, k') & \text{if } l = l' \\ a & \text{otherwise.} \end{cases} \quad (12)$$

The conditional probability distribution should be changed in a similar way, taking into account, that there are functions  $q$  for each segment (each segment is filled in with its own texture):

$$P(x | f, s) = \prod_{r \in R} \hat{q}(x(r), f(r), s(r)). \quad (13)$$

We would like to note, that for  $\sigma_g \rightarrow \infty$  in (2) (i.e. if the topology of the torus "does not matter") this model becomes the usual Potts model for segmentation (see e.g. [3,6]).

If the Gaussian probability distributions (4) are used, the conditional probability distribution  $P(x | s)$  becomes conditionally independent with Gaussian mixtures for  $P(v | l)$ .

We formulate the task of texture segmentation as Bayes decision task with respect to the additive delta function  $C(s, s') = \sum_r \mathbb{1}(s(r) \neq s'(r))$  for misclassification of segments, ignoring the correctness of recognition of the position labeling  $f$  (since it is not necessary to make a decision about position labeling in context of the segmentation problem). In this case the task is to find the most probable segment for each pixel:

$$s^*(r) = \arg \max_l P(s(r) = l | x) \quad \forall r \in R \quad (14)$$

with the marginal a-posteriori probabilities for segments

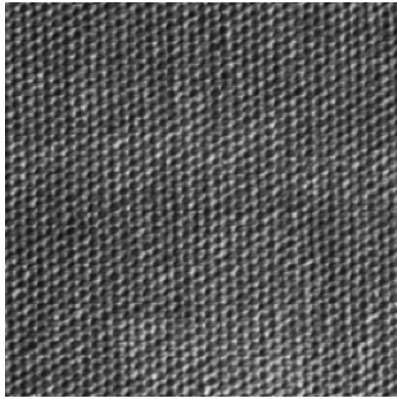
$$P(s(r) = l | x) \sim \sum_{s: s(r)=l} \sum_f P(s, f) \cdot P(x | s, f). \quad (15)$$

We estimate these probabilities in the same manner as for  $P(f(r) = k | x)$  using the Gibbs Sampler.

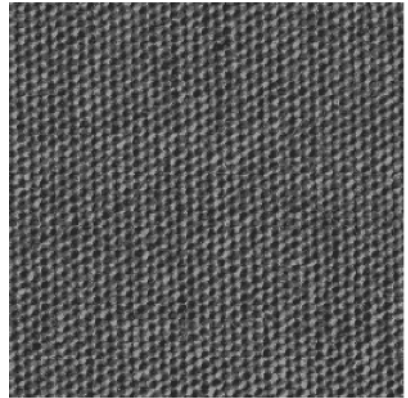
## 4 Results

First of all, we would like to present an example for texture parameters learning. In Fig. 2 an image of a real texture is shown. In Fig. 2(c) one of generated position labelings  $f$  is presented, where the states are encoded as gray values in an illustrative manner, shown in Fig. 2(d). The estimated templates are given in Fig. 2(e) and 2(f) (the dispersion map is scaled to fit into the grayvalue range 0...256). To give more impression, we present also an artificially generated image (see Fig. 2(b)) produced in the following way. For each pixel  $r$  the most probable grayvalue for the corresponding label  $f(r)$  is chosen – i.e. the grayvalue in a pixel  $r$  is set to  $\mu_{f(r)}$ . We will refer images generated in such a manner as "reconstructed images". In our opinion the reconstructed texture in the Fig. 2(b) looks very realistic.

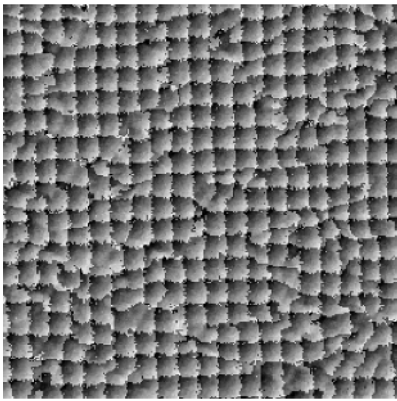
The results of texture segmentation for an artificial image are presented in Fig. 3. The original image Fig. 3(a) was produced as follows. Firstly, a manually painted binary image was chosen as the true segmentation. Each segment was then filled in with different textures. These textures were in addition manually normalized in such a way, that their histograms "look similar". Therefore, it was practically not possible to segment the image correctly without taking into account texture information. The textures were pre-learned using texture examples (i.e. images, consisting of only one texture). Then, the Gibbs Sampler was started with pre-learned templates and randomly chosen segmentation  $s$  and position labeling  $f$ . During the sampling process the templates were further learned by (10) in an unsupervised manner. At the same time the relative frequencies of occurrences of labels (histograms to estimate the marginal probabilities  $P(s(r) = l | x)$  for segments) were observed, based on which the final decision for segmentation was made for each pixel. The resulting segmentation (overlaid with the true segmentation) is shown in Fig. 3(c). The pixels, where the resulting segmentation is not correct, are depicted by gray color. The total percentage of misclassified pixels is 1.2%.



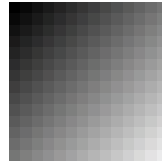
(a) Original texture



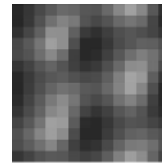
(b) Reconstructed texture



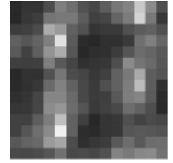
(c) Position labeling



(d) Encoding

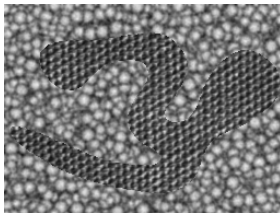


(e) Template:  $\mu$

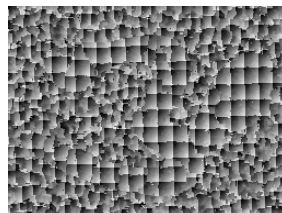


(f) Template:  $\sigma$

**Fig. 2.** Example of a texture



(a) Original image



(b) Position labeling

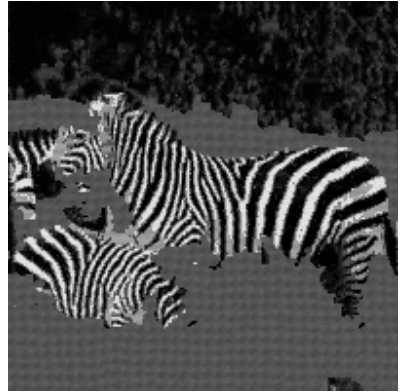


(c) Segmentation

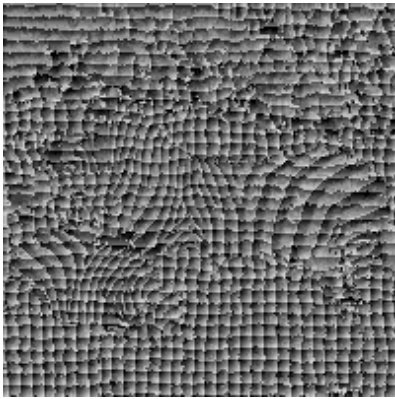
**Fig. 3.** Texture segmentation, an artificial example



(a) Original image



(b) Reconstructed image



(c) Position labeling



(d) Segmentation



(e) 1-th:  $\mu$



(f) 1-th:  $\sigma$



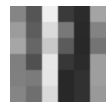
(g) 2-th:  $\mu$



(h) 2-th:  $\sigma$



(i) 3-th:  $\mu$



(j) 3-th:  $\sigma$

**Fig. 4.** Texture segmentation, a real example

Finally we tested our method for a real segmentation task, shown in Fig. 4. We would like to note especially, that the model was able to produce more or less reasonable results even in the situation, where the textures, presented in the image, do not correspond exactly to the model. This is clearly visible for the "grass" texture (shown with gray color in Fig. 4(d)). The parameters of this texture were obtained by the EM-algorithm in a way, which can be characterized as "the best possible way in the scope of the used model" – since this texture can not be represented adequately by the model (see the



reconstructed image in Fig. 4(b)), an almost homogenous coloring combined with high dispersion values was estimated (see Fig. 4(g) and 4(h)).

## 5 Conclusion

In this work we presented a statistical texture model. Typical tasks were considered such as e.g. learning of model parameters, texture segmentation etc.

We would like to point out, that the aim of this paper was to develop just a model, but not algorithms or methods for some particular tasks. It is easy to see, that many other tasks can be formulated in terms of our model in a very natural way. For example those tasks can be considered, where not the processed image itself (or its interpretation) is of interest, but just the texture parameters, such as e.g. the size of the template, color distribution on the template etc. Another interesting task is to estimate parameters of template distribution over the image, such as density, characteristic lengths, regularity etc. Related practical tasks appear for instance in medical imaging, microscopy, inspection and other applications, where the processed image consists of a "large number of very similar objects".

The main question we would like to discuss in that context is the following. In this work we did not consider in detail certain model parameters, like e.g. the size of the etalon, types of the used functions  $g_{rr'}$  and  $q_r$  in (5), possible cost functions for the recognition task (7) etc. On the other hand, for a particular application it is almost always necessary to "specialize" the general model in some reasonable way to reach good performance, because general models in their pure form are often not able to produce good practical results. In that situation the explanation of the model would be not complete without further recommendations about possibilities to use our method in particular applications, i.e. possibilities to incorporate additional a-priori knowledge in the model.

The main direction to do that is based on the fact, that the template itself can be considered as a kind of image (with its own neighborhood structure, coloring properties etc.). This means, that many general image processing methods can be applied directly to the used template (rather than to the processed image). For example, some additional coloring properties of the template can be required – e.g. that the neighboring color values (or color distributions) should be "similar", an implicit color reduction can be performed by fixing the number of color values in the etalon etc. Another possibility is to require, that the shape of the template should have some predefined form, modeling this form by splines, active contours, levelsets etc. The coloring of the template can be for instance defined as a color function, given up to a relatively small number of parameters. Furthermore the coloring of the template can be modeled e.g. using Gibbs probability distributions for images in a classical manner. In this case the template can be considered as a random variable rather than a parameter of the whole probability distribution. Consequently it becomes possible to incorporate additional a-priori knowledge about the template by introducing an a-priori probability distribution over the set of templates, instead of simply fixing it or learning it according to the Maximum Likelihood principle (which corresponds in fact to the maximum a-posteriori decision with respect to parameters).

The above considerations relate to the conditional probability distribution, i.e. the part of the model, which is determined by the template. Another way to specialize the model is to change the a-priori part in a way, that it would fit better a particular application. In this paper we considered e.g. only very simple functions  $g_{rr'}$ . For a particular application it would be obviously profitable to choose the type of these functions, taking into account special properties of the problem.

As we can see, there are many ways to use the developed model for different problems. Further developments of the model, precise examinations of the above mentioned possibilities as well as generalizations and specializations will be subject of future work.

## References

1. A. P. Dempster, N. M. Laird, and D. B. Durbin, *Maximum likelihood from incomplete data via the EM algorithm*, Journal of the Royal Statistical Society **39** (1977), 185–197.
2. B. Flach and R. Sara, *Joint non-rigid motion estimation and segmentation*, IWCMIA04, 2004, pp. 631–638.
3. B. Flach, D. Schlesinger, E. Kask, and A. Skulisch, *Unifying registration and segmentation for multi-sensor images*, DAGM 2002 (Luc Van Gool, ed.), LNCS 2449, Springer, 2002, pp. 190–197.
4. Stuart Geman and Donald Geman, *Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images*, IEEE Transactions on Pattern Analysis and Machine Intelligence **6** (1984), no. 6, 721–741.
5. G. L. Gimel'farb, *Image textures and gibbs random fields*, Dordrecht : Kluwer Academic Press, 1999.
6. I. Kovtun, *Texture segmentation of images on the basis of markov random fields*, Tech. report, TUD-FI03, May 2003.
7. Schlesinger M.I. and Hlavác V., *Ten lectures on statistical and structural pattern recognition*, Kluwer Academic Publishers, Dordrecht, May 2002.
8. D. Schlesinger, *Gibbs probability distributions for stereo reconstruction*, DAGM 2003 (B. Michaelis and G. Krell, eds.), LNCS, vol. 2781, 2003, pp. 394–401.
9. Michail I. Schlesinger, *Connection between unsupervised and supervised learning in pattern recognition*, Kibernetika **2** (1968), 81–88, in Russian.
10. D. Shlezinger, *Strukturelle Ansätze für die Stereorekonstruktion*, Ph.D. thesis, Technische Universität Dresden, 2005, in German, <http://nbn-resolving.de/urn:nbn:de:swb:14-1126171326473-57594>.

# Efficient Combination of Probabilistic Sampling Approximations for Robust Image Segmentation

Jens Keuchel and Daniel Küttel

Institute of Computational Science, ETH Zurich, 8092 Zurich, Switzerland  
Jens.Keuchel@inf.ethz.ch, dkuettel@student.ethz.ch

**Abstract.** Methods based on pairwise similarity relations have been successfully applied to unsupervised image segmentation problems. One major drawback of such approaches is their computational demand which scales quadratically with the number of pixels. Adaptations to increase the efficiency have been presented, but the quality of the results obtained with those techniques tends to decrease. The contribution of this work is to address this tradeoff for a recent convex relaxation approach for image partitioning. We propose a combination of two techniques that results in a method which is both efficient and yields robust segmentations. The main idea is to use a probabilistic sampling method in a first step to obtain a fast segmentation of the image by approximating the solution of the convex relaxation. Repeating this process several times for different samplings, we obtain multiple different partitionings of the same image. In the second step we combine these segmentations by using a meta-clustering algorithm, which gives a robust final result that does not critically depend on the selected sample points.

## 1 Introduction

Unsupervised image segmentation is of essential relevance for many computer vision applications. The goal is to find groups of similar image elements (usually the pixels) according to some locally extracted low-level cues like brightness, color, texture, etc., without having any prototypical information available. In this context, research has recently focused on techniques that are based on pairwise similarities [1,2,3,4]. As the input data is encoded directly in the similarity values, these approaches offer more flexibility than methods that operate on feature vectors, since no assumptions on the distribution of the data points are necessary [5]. Interpreting the similarity values as weighted edges connecting the pixels, segmentation can be formulated as a graph partitioning problem (Section 2). Depending on how the corresponding extremal cuts are defined and computed, this results in spectral relaxation [1,6,7], convex relaxation [4], deterministic annealing [2], or stochastic clustering approaches [3].

However, pairwise segmentation techniques have one major handicap: with increasing image size, they soon become computationally infeasible in terms of memory and solution time as the required number of similarity values grows quadratically with the number of pixels. Different suggestions have been made

to overcome this problem. One idea is to replace pixels as the basic image features by larger coherent patches [8,4,7]. However, this introduces the question of how to aggregate pixels appropriately without losing valuable information and destroying the optimal cuts. Another common approach is to revert to sparse graph representations where only pixels in a certain neighborhood are connected to each other [2,1]. While especially spectral methods benefit from this idea by using special eigenvector calculations, the convex relaxation technique we will consider here does not since it still needs to compute dense matrices. Besides, sparse representations may result in oversegmentation of large homogeneous regions, as long-range connections are not considered appropriately [5].

In this work, we investigate the Nyström method as an alternative technique that is based on probabilistic sampling of the input data: picking only a small random subset of the image pixels, a small scale partitioning problem is defined that can be solved efficiently. Since the number of coherent parts in an image is typically much lower than the number of pixels, this solution usually generalizes well to a solution of the full original problem. While the Nyström method has recently been applied successfully for normalized cuts [5] and kernel-based learning tasks [9,10], our contribution consists in adopting it in the context of convex relaxation methods for image segmentation (Section 3). Related approaches use probabilistic SVD techniques [11] or multiplicative weights updating [12] for computing fast approximations of semidefinite programming relaxations.

While probabilistic sampling is appealing concerning computational efficiency, it also results in a decrease of the segmentation quality. Our second contribution addresses this tradeoff by suggesting to compute multiple partitionings of the same image based on different samplings, and combining them afterwards to obtain a final, more robust segmentation that depends less critically on the selected sample points. This idea of building cluster ensembles has become a very active field of research recently (see [13,14,15] and references therein). In this work, we will employ the meta-clustering algorithm (MCLA) proposed in [13], which is based on directly “clustering clusters” (Section 4), and which has proven to be a strong competitor in comparison to other aggregation techniques [15]. Moreover, this method has the advantage that it does not rely on representing the individual segmentations by co-association matrices containing non-zero entries for every pair of pixels that belong to the same group. Since these matrices are prohibitively large for image segmentation problems, corresponding approaches [13,16,15] cannot be applied in our context.

Experimental results given in Section 5 demonstrate that by combining the probabilistic Nyström approach for pairwise grouping with the MCLA ensemble technique, we obtain fast and stable image segmentations.

## 2 Graph-Based Image Partitioning

Unsupervised image segmentation problems based on pairwise similarities can be interpreted as graph partitioning problems. Representing the image by a graph  $G(V, E)$  with  $|V| = n$  vertices corresponding to the image pixels, the entries

of the symmetric similarity matrix  $W \in \mathbb{R}^{n \times n}$  define pairwise edge weights  $w_{ij} \in \mathbb{R}_0^+$  that are usually obtained from a specific combination of feature values like position, brightness, color, texture, etc. Finding a reasonable segmentation then corresponds to seeking ‘good’ cuts through this graph. More specifically, in the binary case (which is also considered in this work) the graph is split into two coherent parts  $S$  and  $\bar{S} = V \setminus S$  by minimizing a suitable cost function  $f(S)$  which depends on the weight of the corresponding cut:  $\text{cut}(S) = \sum_{i \in S, j \in \bar{S}} w_{ij}$ . If a partition is represented by an indicator vector  $x \in \{-1, +1\}^n$  with  $x_i = 1$  for  $i \in S$ , and  $L = D - W$  denotes the Laplacian matrix of the graph ( $D$  is the diagonal matrix with the vertex-degrees  $d_i = \sum_{j \in V} w_{ij}$  on its diagonal), we can write the weight of a binary cut as

$$\text{cut}(x) = \frac{1}{4} x^\top L x. \quad (1)$$

A segmentation into multiple parts can be obtained by applying the binary method hierarchically to the obtained segments.

Since directly minimizing the cut-weight (1) favors separating small sets of isolated vertices from the rest of the graph [7], different measures  $f(S)$  have been proposed in the literature to prevent such unbalanced partitionings. A very popular approach is to scale the cut value (1) appropriately, which leads to optimization criteria like normalized cuts [1,5], average cuts [6], or ratio cuts [8], that often can be solved approximately by efficient eigenvector calculations.

In this work, we revert to an alternative technique that is based on a classical idea from spectral graph theory: instead of scaling the cut value, an additional balancing constraint is introduced. This leads to the following equivalent problem formulations [4]:

$$\begin{aligned} \min_{x \in \{-1, +1\}^n} x^\top L x & \quad \iff \quad \max_{x \in \{-1, +1\}^n} x^\top W x \\ \text{s.t. } c^\top x = 0 & \quad \quad \quad \text{s.t. } c^\top x = 0, \end{aligned} \quad (2)$$

where  $c \in \mathbb{R}^n$  denotes a fixed weight vector. The equivalence of the two problems is easily seen since  $x^\top L x = x^\top D x - x^\top W x = \sum_i d_i - x^\top W x$  for binary vectors  $x \in \{-1, +1\}^n$ . It may be criticized that this approach is too restrictive since it only admits cuts of a specific vertex balance; however, during the following solution process the balancing constraint is relaxed intrinsically and merely serves as a bias to guide the search to meaningful segmentations.

In order to solve the combinatorial problem (2), we use the convex relaxation method proposed in [4]. First, the problem variables are lifted into a higher dimensional matrix space by rewriting the objective function as  $x^\top W x = \text{tr}(W x x^\top)$  and replacing the positive semidefinite rank one matrix  $x x^\top$  by an arbitrary positive semidefinite matrix  $X \in \mathcal{S}_+^n$ . Lifting the constraints in a similar way, we obtain the following relaxation of (2):

$$\begin{aligned} \max_{X \in \mathcal{S}_+^n} \text{tr}(W X) \\ \text{s.t. } \text{tr}(c c^\top X) = 0 \\ X_{ii} = 1 \quad i = 1, \dots, n. \end{aligned} \quad (3)$$

This problem is a *semidefinite program (SDP)*, which due to its convexity can be solved to arbitrary precision in polynomial time using standard interior point methods (cf. [17]). In a last step we recover a binary solution  $x$  from the solution matrix  $X$  based on the randomized-hyperplane technique introduced in [18]. Since we are going to partition images on the basis of pixels, we set  $c = e = (1, \dots, 1)^\top$  in (2), which results in graph vertices of equal importance.

### 3 Probabilistic Approximation

As already stated in the introduction, graph-based partitioning methods soon become intractable with increasing image size. In order to reduce the computational effort we employ an idea that is derived from the *Nyström method*, a technique for fast computation of low-rank matrix approximations which is based on probabilistically sampling the matrix entries [5,10]. In our context, this translates into picking a small random subset of the image pixels to implicitly approximate the huge similarity matrix  $W \in \mathbb{R}^{n \times n}$  with a matrix  $\hat{W}$  of considerably lower rank  $s \ll n$ , for which an approximate solution of the original partitioning problem can be deduced efficiently.

To motivate this idea, we first show that the original partitioning problem (2) can also be interpreted as a special matrix approximation problem. To this end, observe that the Frobenius norm  $\|xx^T\|_{\mathbb{F}}^2 = n^2$  is constant for each partitioning vector  $x \in \{-1, +1\}^n$ . Comparing the rank one matrix  $xx^T$  with the similarity matrix  $W$  then results in

$$\|W - xx^T\|_{\mathbb{F}}^2 = \|W\|_{\mathbb{F}}^2 + \|xx^T\|_{\mathbb{F}}^2 - 2 \operatorname{tr}(Wxx^T) = \|W\|_{\mathbb{F}}^2 + n^2 - 2x^T W x.$$

Hence, as  $\|W\|_{\mathbb{F}}^2$  is also constant, problem (2) is equivalent to seeking the best rank one approximation to  $W$  in Frobenius norm (subject to additional constraints). This suggests to replace the probably full-rank matrix  $W$  by an appropriate low-rank approximation before applying the relaxation (3): without changing the problem setting too drastically, the approximation process is simplified significantly in this way.

As a first step, the Nyström method requires to pick a fixed number  $s$  of suitable sample points. Since a uniform distribution of the input data is an adequate assumption for dense similarity matrices [19], we simply select  $s$  pixels independently at random from the image. Assuming without loss of generality that the samples precede the remaining points, we can subdivide the symmetric similarity matrix  $W$  into smaller submatrices:

$$W = \begin{pmatrix} A & B \\ B^\top & C \end{pmatrix},$$

with  $A \in \mathbb{R}^{s \times s}$ ,  $B \in \mathbb{R}^{s \times n-s}$  and  $C \in \mathbb{R}^{n-s \times n-s}$ . Hence,  $A$  represents the similarities among the random samples, whereas  $B$  contains the weights between the samples and the remaining points.

The Nyström method now uses the sampled  $n \times s$  submatrix  $S := (A \ B)^\top$  to directly approximate the complete similarity matrix  $W$  with a rank- $s$  matrix  $\hat{W}$

by implicitly approximating the large submatrix  $C$  of unknown similarity values with the matrix  $B^\top A^{-1}B$ :

$$W \approx \hat{W} = \begin{pmatrix} A & B \\ B^\top & B^\top A^{-1}B \end{pmatrix} = \begin{pmatrix} A \\ B^\top \end{pmatrix} A^{-1} (A B) = SA^{-1}S^\top. \quad (4)$$

The big advantage of this idea is that the eigenvalue decomposition of the approximating matrix  $\hat{W} = \hat{Q}_s \hat{\Sigma}_s \hat{Q}_s^\top$  (where  $\hat{\Sigma}_s \in \mathbb{R}^{s \times s}$  and  $\hat{Q}_s \in \mathbb{R}^{n \times s}$  only contain the nonzero eigenvalues and the corresponding eigenvectors) can be calculated very efficiently: Considering the eigenvalue decomposition  $P\Sigma P^\top$  of the  $s \times s$  matrix

$$\widetilde{W} := A + A^{-\frac{1}{2}}BB^\top A^{-\frac{1}{2}} = A^{-\frac{1}{2}}S^\top SA^{-\frac{1}{2}}, \quad (5)$$

it is easy to verify that the eigenvalues of  $\widetilde{W}$  and  $\hat{W}$  coincide,  $\hat{\Sigma}_s = \Sigma$ , and that

$$\hat{Q}_s = SA^{-\frac{1}{2}}P\Sigma^{-\frac{1}{2}}. \quad (6)$$

Hence, this technique yields a fast approximation of the original similarity matrix  $W$  by approximating its eigenvectors.

A crucial aspect of the Nyström method is the calculation of the inverse  $A^{-1}$  and the square root  $A^{\frac{1}{2}}$  of the submatrix  $A$ : if  $A$  is singular or has negative eigenvalues, then these matrices are not defined. As a remedy for these cases, Fowlkes et al. [5] propose to use the pseudoinverse instead of the inverse (if any of the eigenvalues of  $A$  are zero) or to apply a modified technique that does not need to calculate the square root  $A^{\frac{1}{2}}$  (if  $A$  is indefinite). However, besides increasing the computational effort, these modifications may lead to a significant loss in numerical precision, and thus should only be applied when necessary [5]. We will therefore assume that the similarity matrix  $W$  and with it the submatrix  $A$  are *positive definite*, which guarantees that both the inverse  $A^{-1}$  and the square root  $A^{\frac{1}{2}}$  exist. In general, this is not a rigorous restriction since most common similarity measures are derived as kernel functions. Even if this is not the case, we can simply modify  $W$  by adding a multiple of the identity matrix,  $W + \gamma I$  with  $\gamma \in \mathbb{R}^+$  large enough to ensure positive definiteness without changing the eigen-structure of  $W$ .

In order to apply the Nyström method to the SDP relaxation approach (3), we first replace the similarity matrix  $W$  by the low-rank approximation  $\hat{W}$  from (4), and then compute a corresponding low-rank approximation  $\hat{X}$  to the solution  $X^*$  of (3) by means of the small matrix  $\widetilde{W}$  given in (5). To this end, we use the fact that during the final step of the SDP method, the randomized hyperplane technique [18] reduces the rank of the optimal solution  $X^*$  based on its Cholesky decomposition  $X^* = GG^\top$ . We therefore suggest to compute the low-rank approximation  $\hat{X}$  by finding an appropriate approximative Cholesky factor  $\hat{G} \in \mathbb{R}^{n \times s}$ , so that  $\hat{X} = \hat{G}\hat{G}^\top$ . If we define  $\hat{G} = SA^{-\frac{1}{2}}\tilde{G}$  in analogy to computing the eigenvectors  $\hat{Q}_s$  of  $\hat{W}$  in (6), the objective function of the SDP relaxation (3) can be approximated as:

$$\begin{aligned} \text{tr}(WX) &\approx \text{tr}(\hat{W}\hat{X}) = \text{tr}(SA^{-1}S^\top \hat{G}\hat{G}^\top) = \text{tr}(SA^{-1}S^\top SA^{-\frac{1}{2}}\tilde{G}\tilde{G}^\top A^{-\frac{1}{2}}S^\top) \\ &= \text{tr}((A^{-\frac{1}{2}}S^\top SA^{-\frac{1}{2}})(A^{-\frac{1}{2}}S^\top SA^{-\frac{1}{2}})\tilde{G}\tilde{G}^\top) = \text{tr}(\widetilde{W}^\top \tilde{X}). \end{aligned}$$

Hence, we propose to compute the approximative Cholesky factor  $\hat{G}$  from the Cholesky factor  $\tilde{G}$  of the small matrix  $\tilde{X} := \tilde{G}\tilde{G}^\top$  that is obtained as the optimal solution of the following small-size version of the SDP problem (3):

$$\begin{aligned} \max_{\tilde{X} \in \mathcal{S}_+^s} \quad & \text{tr}(\tilde{W}^2 \tilde{X}) \\ \text{s.t.} \quad & \text{tr}(ee^\top \tilde{X}) = 0 \\ & \tilde{X}_{ii} = 1 \quad i = 1, \dots, s. \end{aligned} \tag{7}$$

Since the unit norm constraint on the rows  $G_i$  of the original Cholesky factor  $G$  required by  $1 = X_{ii} = G_i G_i^\top = \|G_i\|^2 = 1$  is not necessarily satisfied by the approximation  $\hat{G}$ , we additionally normalize its rows. In the last step, a corresponding binary solution is calculated by adapting the randomized hyperplane technique: using random vectors  $r$  from the unit sphere in  $\mathbb{R}^s$ , we obtain binary vectors  $x = \text{sgn}(\hat{G}r) \in \{-1, +1\}^n$ , from which we pick the best one according to an adjusted version  $x^\top Sx_s$  of the objective function. Here,  $x_s \in \mathbb{R}^s$  denotes the vector that only contains the first  $s$  entries of  $x$ .

## 4 Aggregation to Ensemble Solutions

Although the probabilistic approximation technique presented in the last section yields satisfactory segmentations, these are usually noisy and depend strongly on the choice of the samples. However, the gain in efficiency allows us to compute several solutions based on different samplings, and to merge the resulting segmentations to a final, more robust and less noisy ensemble solution.

To this end, we employ the meta-clustering algorithm (MCLA) presented by Strehl and Ghosh [13]. The basic idea of this method is to consider each cluster of the set of sampling-based segmentations as an object, find groups of similar objects (“clustering clusters”) and label the original pixels by assigning them to the group in which they participate most strongly. On order to solve this partitioning problem, Strehl and Ghosh define a meta-graph  $G'(V', E')$  with the extracted clusters of the segmentations as meta-vertices  $V'$ , i.e. for  $m$  different segmentations each containing  $k$  clusters we obtain  $mk$  meta-vertices. The pairwise meta-edges  $(p, q) \in E'$  are weighted proportionally to the pixel overlap between the corresponding clusters  $V'_p$  and  $V'_q$  by computing the ratio between their intersection and their union:  $w'_{pq} = |V'_p \cap V'_q| / |V'_p \cup V'_q|$ . Since the individual clusters of one segmentation do not overlap, this results in an  $m$ -partite meta-graph.

This meta-graph  $G'$  is then partitioned into  $k'$  balanced meta-clusters, where we set  $k' = k$  to obtain the same number of clusters as in the individual sampling-based segmentations. While for this purpose, we could use generalizations of the methods presented in Section 2, we follow the suggestion in [13] and use the partitioning package METIS [20]. For every pixel  $i$ , its level of association  $a_i(C_l) \leq 1$  with each meta-cluster  $C_l$  is computed by summing the occurrences of  $i$  in the clusters  $V'_p \in C_l$  and dividing by  $|C_l|$ . In the last step, each pixel is

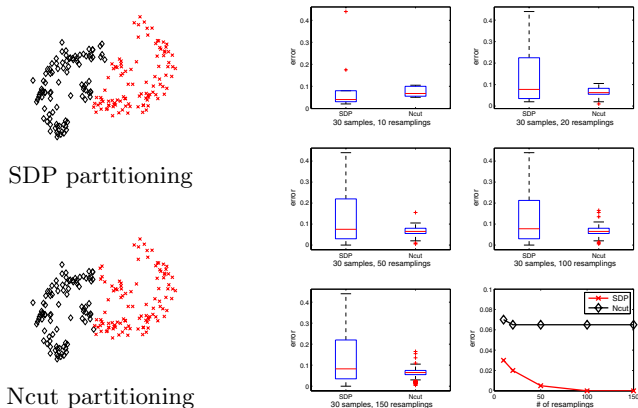


assigned to the meta-cluster  $C_l$  for which  $a_i(C_l)$  is maximal. Although this may result in fewer than  $k$  parts, we note that MCLA yields a stable aggregation of the different segmentations.

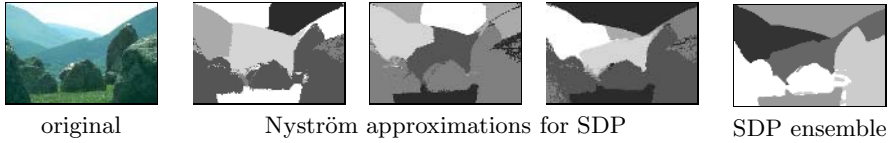
## 5 Experimental Results

In order to obtain suitable similarity values for the following experiments, we consider feature vectors  $y_i$  for the image pixels  $i$  that contain the color in the perceptually uniform  $L^*u^*v^*$  space along with the position of  $i$  within the image. The position is included here to encode some spatial information, since the need for dense similarity matrices prohibits using small neighborhood structures for this purpose. The corresponding similarity values  $w_{ij}$  are then calculated from the Mahalanobis distances between these (appropriately scaled) feature vectors as  $w_{ij} = \exp(-\frac{1}{2}(y_i - y_j)^\top \Sigma^{-1}(y_i - y_j))$ , which results in a positive definite similarity matrix  $W$ . While more intricate similarity measures could be used within our framework, such an investigation is beyond the scope of this paper. Besides computing MCLA ensembles for the sampling-based convex relaxation (SDP) method, we also apply it to the Nyström approximation to normalized cuts as described in [5] for comparison.

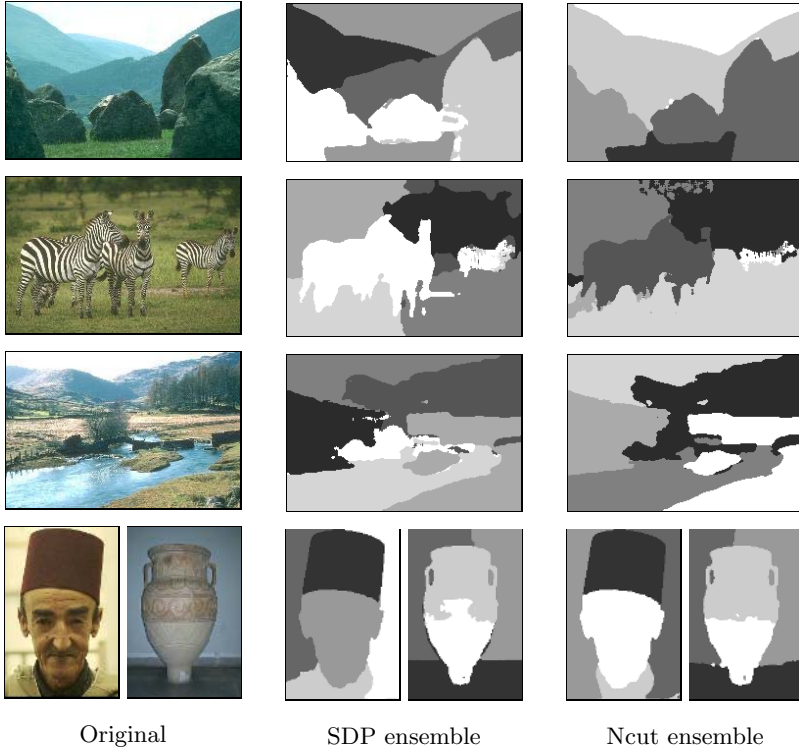
In a first experiment, we test the quality of the ensemble approach on the small artificial point set depicted in Figure 1, left, which contains  $n = 200$  points distributed equally over two spiral-shaped clusters. The complexity of this problem setting (based on point positions only) is indicated by the solutions obtained with both convex relaxation and normalized cuts, which do not reproduce the



**Fig. 1. Performance of the sampling-ensemble combination** for an artificial point set, containing two spiral-shaped clusters. The box-plots indicate the distribution of the differences between the solution on the complete point set (shown on the left) and the approximative solutions obtained for different numbers of resamplings (for fixed sample number  $s = 30$ ). The last figure indicates the high quality of the final aggregated results.



**Fig. 2. Segmentation based on different samplings.** The smoothed ensemble solution is much more appealing than the individual segmentations.



**Fig. 3. Ensemble solutions** obtained from 11 different hierarchical segmentations, with Nyström approximations being based on less than 1% of the pixels

spirals exactly. Figure 1, right, gives the performance results based on changing the number of resamplings while keeping the number of samples  $s$  fixed. The box-plots show that the individual sampling-based SDP solutions have a high variability, with a median error value slightly below 0.1. The important point, however, is that the aggregation results in a solution that is better than the average sampled solution, and for which the error converges to zero with increasing number of resamplings (last plot on the right). In comparison to that, Nyström-based normalized cut solutions are less variable, which results in ensemble solutions of the same quality as the average sampling-based clusterings. This is due to the very accurate approximation of the leading eigenvectors in

this case, which reproduces the spiral structure and hence prevents appropriate clustering of the eigenvector entries. However, the robustness of the ensemble results is also approved by normalized cuts.

To evaluate the performance for large real world scenes, we use images from the Berkeley segmentation dataset [21]. In order to produce partitionings into more than two segments, we employ hierarchical segmentation for a fixed number of segments ( $k = 6$ ). In each step, we randomly pick  $s = 100$  sample pixels, which corresponds to less than 1% of the entire image. The final ensemble result obtained from 11 different sampling-based solutions is additionally smoothed with a small linear filter to remove remaining noise. Figure 2 gives an example that demonstrates the power of the aggregation process: in comparison to the individual segmentations, the combined result is much more appealing and robust. The results in Figure 3 reveal that the Nyström approximations for both the convex relaxation approach as well as for normalized cuts give satisfactory aggregated segmentations. The main objects were found, with only a few inaccurate edges that are mainly due to the fact that we only use basic color information without any particular effort to tune the scaling parameters or the hierarchical process. Concerning the computational effort, it took just about 90 seconds for the SDP relaxation and 20 seconds for normalized cuts to find the individual sampling-based segmentations. In comparison to that, the final aggregation process with an average duration of 15 seconds is negligible.

## 6 Conclusion

In this paper, we have demonstrated the power of combining a sampling-based approximation method with a cluster aggregation procedure for unsupervised image segmentation. While the computational effort is greatly reduced by reverting to the Nyström method for computing individual partitionings, the arising problems concerning sample selection and inaccuracy are effectively dealt with during the following merging process. Experimental results reveal that robust image segmentations of appealing quality are obtained with this approach.

Our technique offers several directions for future research: For example, it would be useful to employ other cues like texture or edge separation to compute additional approximate segmentations as a basis of the aggregation process. In contrast to approaches that combine different cues *before* computing the segmentation, our ensemble technique does not require weighting the features against each other, but simply aggregates the results *after* segmenting the image. In this context, it might also be beneficial to merge solutions that besides different samplings are based on different parameter settings or similarity calculations. Moreover, the association values produced by MCLA provide a measure of confidence for each pixel to belong to a segment. These confidence values yield important information on the stability of the result, and can be used to modify the sample distributions for subsequent steps by moving more samples to currently unstable segment boundaries.

## References

1. Shi, J., Malik, J.: Normalized cuts and image segmentation. *IEEE Trans. PAMI* **22**(8) (2000) 888–905
2. Hofmann, T., Puzicha, J., Buhmann, J.M.: Unsupervised texture segmentation in a deterministic annealing framework. *IEEE Trans. PAMI* **20**(8) (1998) 803–818
3. Shental, N., Zomet, A., Hertz, T., Weiss, Y.: Learning and inferring image segmentations using the GBP typical cut algorithm. In: *Proc. ICCV, IEEE Comp. Soc.* (2003) 1243–1250
4. Keuchel, J., Schnörr, C., Schellewald, C., Cremers, D.: Binary partitioning, perceptual grouping, and restoration with semidefinite programming. *IEEE Trans. PAMI* **25**(11) (2003) 1364–1379
5. Fowlkes, C., Belongie, S., Chung, F., Malik, J.: Spectral grouping using the Nyström method. *IEEE Trans. PAMI* **26**(2) (2004) 214–225
6. Sarkar, S., Soundararajan, P.: Supervised learning of large perceptual organization: Graph spectral partitioning and learning automata. *IEEE Trans. PAMI* **22**(5) (2000) 504–525
7. Wu, Z., Leahy, R.: An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *IEEE T-PAMI* **15**(11) (1993) 1101–1113
8. Wang, S., Siskind, J.M.: Image segmentation with ratio cut. *IEEE Trans. PAMI* **25**(6) (2003) 675–690
9. Williams, C.K.I., Seeger, M.: Using the Nyström method to speed up kernel machines. In: *NIPS 13*. (2001) 682–688
10. Drineas, P., Mahoney, M.: On the Nyström method for approximating a Gram matrix for improved kernel-based learning. *J. Mach. Learning Res.* **6** (2005) 2153–2175
11. Keuchel, J., Schnörr, C.: Efficient graph cuts for unsupervised image segmentation using probabilistic sampling and SVD-based approximation. In: *3rd International Workshop on Statistical and Computational Theories of Vision, Nice, France* (2003)
12. Arora, S., Hazan, E., Kale, S.: Fast algorithms for approximate semidefinite programming using the multiplicative weights update method. In: *Proc. FOCS*. (2005) 339–348
13. Strehl, A., Ghosh, J.: Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *J. Mach. Learning Res.* **3** (2003) 583–617
14. Fischer, B., Buhmann, J.M.: Bagging for path-based clustering. *IEEE Trans. PAMI* **25**(11) (2003) 1411–1415
15. Lange, T., Buhmann, J.M.: Combining partitions by probabilistic label aggregation. In: *Proc. KDD*. (2005) 147–156
16. Fred, A.L., Jain, A.K.: Data clustering using evidence accumulation. In: *ICPR* (4). (2002) 276–280
17. de Klerk, E.: *Aspects of Semidefinite Programming*. Kluwer Academic Pub. (2002)
18. Goemans, M.X., Williamson, D.P.: Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM* **42**(6) (1995) 1115–1145
19. Frieze, A.M., Kannan, R., Vempala, S.: Fast Monte-Carlo algorithms for finding low-rank approximations. In: *Proc. on FOCS*. (1998) 370–378
20. Karypis, G., Kumar, V.: A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM J. Sci. Comput.* **20**(1) (1998) 359–392
21. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: *Proc. ICCV, IEEE Comp. Soc.* (2001) 416–423

# Diffusion-Like Reconstruction Schemes from Linear Data Models

Hanno Scharr

ICG III, Research Center Jülich, 52425 Jülich, Germany

**Abstract.** In this paper we extend anisotropic diffusion with a diffusion tensor to be applicable to data that is well modeled by linear models. We focus on its variational theory, and investigate simple discretizations and their performance on synthetic data fulfilling the underlying linear models. To this end, we first show that standard anisotropic diffusion with a diffusion tensor is directly linked to a data model describing single orientations. In the case of spatio-temporal data this model is the well known brightness constancy constraint equation often used to estimate optical flow. Using this observation, we construct extended anisotropic diffusion schemes that are based on more general linear models. These schemes can be thought of as higher order anisotropic diffusion. As an example we construct schemes for noise reduction in the case of two orientations in 2d images. By comparison to the denoising result via standard single orientation anisotropic diffusion, we demonstrate the better suited behavior of the novel schemes for double orientation data.

## 1 Introduction

Anisotropic diffusion has been widely used in computer vision and image processing (see [13,26] for recent overviews). It is a scale-space and image reconstruction technique and has been used for e.g. image inpainting, super-resolution, noise removal, or reduction of JPEG artifacts (e.g. [24] and many others). In this paper we introduce a diffusion-like process that can be thought of as higher order anisotropic diffusion. As example application we select image reconstruction, especially noise removal, because its performance is simple to evaluate and visualize (even though the currently best performing image denoising algorithm [15] is not of diffusion type). Other denoising methods are closely related to diffusion such as bilateral filtering [23], or channel smoothing [5].

The basic idea of diffusion-based denoising is to smooth a degraded original image by applying a nonlinear diffusion process whose diffusion tensor allows oriented, anisotropic smoothing. Depending on the choice of the so-called edge-stopping function  $\rho$  [4] not only smoothing, but also sharpening of structures can be achieved [25]. One of the most common ways to construct a diffusion tensor is to calculate the structure tensor [3,9] and exchange its eigenvalues  $\mu_i$  by means of an edge stopping function  $\rho(\mu_i)$  (cmp. [25]). We will show that this way to construct a diffusion tensor is an approximation of the diffusion tensor we derive via a variational approach using a single orientation model (cmp. Sec. 2).

Several publications show how to construct diffusion schemes in a principled way. They introduce knowledge about the data to be restored via training data [17,18,27]. While [18,27] propose learning schemes to select an edge-stopping function, [17] also constructs the filters used in a diffusion-like scheme. In standard diffusion schemes the applied filters are first order derivatives. However the scheme put forward in [17] only extends nonlinear diffusion with a scalar diffusivity as introduced by [14]. This is in contrast to our scheme which does allow directed filtering like tensor driven anisotropic diffusion does.

All the above mentioned denoising schemes are based on the general idea to average over data that belongs to the same population (except maybe [17]). In many cases additional knowledge about the data is available in form of linear models. Such linear models have been proposed for e.g. optical flow [6] with physical brightness changes [8] and multiple orientations or motions [21,10]. Many PDEs known from physics are also of this linear type. To the best of the authors knowledge there is no reconstruction method currently available that allows to use such model knowledge. In this paper we introduce diffusion-like schemes respecting given linear models (cmp. Sec. 3).

There are many different implementation approaches to numerically solve (or simulate) anisotropic diffusion. We use simple two-level explicit finite-difference schemes. But besides this, there are e.g. three-level [7], multigrid [1] or spectral methods [7], finite elements with grid adaptation [16] and many more. We have not implemented all these schemes, giving a 'best performing' implementation is beyond the scope of this paper. But as numerical implementation is critical for performance, we implemented for each model – single and double orientation – two similar schemes each and selected the best scheme for further tests.

The contributions of this paper are: **a**) a novel derivation of standard anisotropic diffusion with a diffusion tensor from a cost function, **b**) a theory for extension of this tensorbased diffusion to general linear models, **c**) an example using a double orientation model (X-junctions), **d**) a simple way to discretize such an extended diffusion, **e**) a rudimental performance evaluation on simple synthetic X-junction data as an experimental proof of concept.

## 2 Variational Derivation of Tensorbased Diffusion

The aim of the scheme to derive is to reconstruct the 'underlying', 'real' data  $g$  from measured data  $f$ . Let  $f$  and  $g$  be 3-dimensional. A usual reconstruction approach is to assume that  $g$  is a smooth function. A membrane-like behavior of  $g$  can be achieved by minimizing the cost function

$$E(g) = \int_{\Omega} (g - f)^2 + \alpha |\nabla g|^2 dx \quad (1)$$

where  $\nabla = (\partial_{x_1}, \dots, \partial_{x_3})^T$  is the gradient operator in 3-dimensions. The positive weight  $\alpha$  forces the solution to be smoother when  $\alpha$  becomes larger. The first term in Eq. 1 is usually called *data term*, the second one *smoothness term*. An extension of this constraint has been proposed by Mumford and Shah [11]. Its connection to (not tensor driven) anisotropic diffusion can be found in [20].

## 2.1 Smoothness Via a Linear Model

In this section we show that the well known brightness constancy constraint equation (BCCE, defined below) leads to tensorbased anisotropic diffusion, when used as a smoothness constraint in a cost function. Although our experiments in Sec. 4 are all done using 2d images, we use a 3d formulation here, because in this form the BCCE is most well known.

Let us assume that  $g$  is a densely sampled image sequence, thus  $x_1 = x$ ,  $x_2 = y$ , and  $x_3 = t$ . In contrast to the above membrane assumption let us further assume that  $g$  should fulfill the BCCE

$$\partial_x g dx + \partial_y g dy + \partial_t g dt = 0 \quad \Leftrightarrow \quad \nabla^T \mathbf{g} \mathbf{u} = 0 \quad (2)$$

where  $\mathbf{u} = (dx, dy, dt)^T$  is a parameter vector. This is a typical linear model, where linear means linear in  $\mathbf{u}$ . In order to avoid the trivial solution  $\mathbf{u} = 0$  one usually either defines  $|\mathbf{u}| = 1$  or  $dt = 1$ . Here we prefer the first assumption, because it is more clearly related to orientation estimation. From this vector optical flow, i.e.  $x$ - and  $y$ -displacements  $u_x$  and  $u_y$ , respectively, can be calculated via  $u_x = dx/dt$  and  $u_y = dy/dt$ . There are many ways to estimate such a parameter vector (see e.g. [2]). For now let us assume we know the probability distribution  $p(\mathbf{u})$  at each pixel and thus could give its expectation value  $\langle \mathbf{u} \rangle$ . We can then formulate the cost function

$$E(g) = \int_{\Omega} (g - f)^2 dx + \alpha \int_{\Omega} \langle (\nabla^T \mathbf{g} \mathbf{u})^2 \rangle dx \quad (3)$$

To the best of the authors knowledge using expectation values in a cost function in this way is new. The solution  $\tilde{g}$  minimizing  $E$  has to fulfill the variational condition  $\delta E(\tilde{g}) = 0$ , where  $\delta$  is a variation. The variation of  $E$  gives us

$$\delta E(g) = \int_{\Omega} 2(g - f) \delta g dx + \alpha \int_{\Omega} \langle 2(\nabla^T \mathbf{g} \mathbf{u}) \mathbf{u}^T \rangle \nabla \delta g dx \quad (4)$$

In the last term on the right hand side  $\nabla \delta g$  can not be determined, thus we partially integrate this term. Please note that each partial integration step flips the sign of this term. For odd order derivative filters like the ones in  $\nabla$  this is equivalent to mirroring the filter. This will be important later. We get

$$\begin{aligned} \delta E(g) &= \int_{\Omega} 2(g - f) \delta g dx - \alpha \int_{\Omega} \nabla^T \langle 2(\nabla^T \mathbf{g} \mathbf{u}) \mathbf{u} \rangle \delta g dx \\ &= \int_{\Omega} 2(g - f) \delta g dx - 2\alpha \int_{\Omega} \nabla^T \langle \mathbf{u} \mathbf{u}^T \rangle \nabla g \delta g dx \end{aligned} \quad (5)$$

Thus the Euler-Lagrange equation  $\tilde{g}$  has to fulfill is

$$0 = (\tilde{g} - f) - \alpha \nabla^T \langle \mathbf{u} \mathbf{u}^T \rangle \nabla \tilde{g} \quad (6)$$

$\langle \mathbf{u} \mathbf{u}^T \rangle$  is a square, symmetric, positive semidefinite matrix. From Eq. 6 we derive the update equation

$$\partial_t g = (f - g) - \alpha \bar{\nabla}^T \langle \mathbf{u} \mathbf{u}^T \rangle \nabla g \quad (7)$$

where  $g$  is treated as a volume and  $\partial_t$  is the derivative with respect to the 'iteration time' of the scheme (not to be confused with the time in an image sequence), and  $\bar{\nabla} = -\nabla$  is a mirrored version of  $\nabla$ . This equation describes anisotropic diffusion of  $g$  with *diffusion tensor*  $\langle \mathbf{u} \mathbf{u}^T \rangle$ . The data term acts as a (grey value) source.

## 2.2 Approximation of the Diffusion Tensor

A standard procedure to get a diffusion tensor is to first calculate the structure tensor  $J$  [3,9] (also a square, symmetric, positive semidefinite matrix)

$$J(\mathbf{x}) = \int w(\mathbf{x} - \mathbf{x}')(\nabla g(\mathbf{x}'))(\nabla^T g(\mathbf{x}'))d\mathbf{x}' \quad (8)$$

where  $w$  is a smoothing kernel, in our case a 5-tab Gaussian (variance  $\sigma = 1$ ) applied in all dimensions. Then one exchanges its eigenvalues  $\mu_i$  by means of an edge stopping function  $\rho(\mu_i)$  (cmp. [25]). We will now show that such a diffusion tensor approximates  $\langle \mathbf{u}\mathbf{u}^T \rangle$ . Being a square, symmetric matrix  $\langle \mathbf{u}\mathbf{u}^T \rangle$  is diagonalizable with eigenvalues  $\lambda_i$ . The eigenvector corresponding to its largest eigenvalue will be close to  $\langle \mathbf{u} \rangle$ . In fact if  $p(\mathbf{u})$  is symmetric around  $\langle \mathbf{u} \rangle$ ,  $\langle \mathbf{u} \rangle$  is exactly an eigenvector of  $\langle \mathbf{u}\mathbf{u}^T \rangle$ .

For local orientation or optical flow estimation using Eq. 2, we may calculate grey value derivatives  $g_x, g_y, g_t$  for all pixels via suitable convolution kernels. For each pixel we get a constraint equation with several unknowns, i.e. the parameter vector  $\mathbf{u}$ . In order to solve for these unknowns, we assume that all equations in a local neighborhood are solved by the same  $\mathbf{u}$ , giving us an over determined system of equations. It is well known that the eigenvector with respect to the smallest eigenvalue of the structure tensor  $J$  is a weighted total least squares solution  $\mathbf{u}_{TLS}$  of this system (see e.g. [9]). If more than one eigenvalue is close to zero,  $\mathbf{u}_{TLS}$  can be any linear combination of the corresponding eigenvectors. A detailed error analysis for  $\mathbf{u}_{TLS}$  can be found in [12].

We see that if eigenvector  $\mathbf{u}_{TLS}$  is a good approximation of  $\langle \mathbf{u} \rangle$  the structure tensor  $J$  has approximately the same eigensystem as  $\langle \mathbf{u}\mathbf{u}^T \rangle$ . But when the eigenvalues  $\mu_i$  of  $J$  are small, the eigenvalues  $\lambda_i$  of  $\langle \mathbf{u}\mathbf{u}^T \rangle$  are large and vice versa. Thus we may approximate  $\langle \mathbf{u}\mathbf{u}^T \rangle$  by taking the structure tensor and exchanging its eigenvalues by

$$\lambda_i = \rho_i(\mu_i) \quad (9)$$

where the functions  $\rho_i$  decay monotonically and  $\rho_i(0) = 1$ . These  $\rho$ -functions are usually selected heuristically. In [18] they are linked to image statistics. In this paper we stick with the structure tensor and heuristics for  $\rho_i$ , because it is not the aim to derive the best possible  $\langle \mathbf{u}\mathbf{u}^T \rangle$ . We rather want to demonstrate how to incorporate more complex linear models into a diffusion-like reconstruction scheme. This is done in Sec. 3.

## 2.3 Discretization of Anisotropic Diffusion

We want to compare to anisotropic diffusion in the experiments on 2d images. Thus we give two discretization schemes for the smoothness term in Eq. 7 (cmp. [19]). The model equation in 2d is

$$\nabla^T g\mathbf{p} = 0 \quad (10)$$

with  $\nabla = (\partial_x, \partial_y)$ . We discretize  $\nabla$  using separable convolution filters

$$\partial_x = \mathcal{D}_x * \mathcal{S}_y \quad \partial_y = \mathcal{S}_x * \mathcal{D}_y \quad (11)$$



where lower indices give the application direction,  $\mathcal{D}$  is a 1d first order derivative filter and  $\mathcal{S}$  is a 1d smoother. For the calculation of  $\nabla$  and  $\bar{\nabla}$  in Eq. 7 we investigate 2 filter sets, i.e. 2 schemes. The first one we call  $2 \times 2$

$$\mathcal{D} = [1, -1, 0] \quad \mathcal{S} = [0.5, 0.5, 0] \quad (12)$$

We write the 2-tab filters as 3-tab filters in order to have the storage point (middle of the filter) well defined, also for the mirrored version  $\bar{\nabla}$ . The second scheme we call  $3 \times 3$

$$\mathcal{D} = [0.5, 0, -0.5] \quad \mathcal{S} = [0.1839, 0.6322, 0.1839] \quad (13)$$

The resulting explicit update scheme using  $\bar{D} = \bar{\nabla}$  is defined below in Eq. 26.

The diffusion tensor  $\langle \mathbf{u}\mathbf{u}^T \rangle$  in Eq. 7 we approximate in both schemes via the structure tensor  $J$  (Eq. 8) as proposed in Sec. 2.2. There, gradient  $\nabla g$  is calculated using separable  $5 \times 5$  filters (cmp. Eq. 11) optimized for accurate single orientation estimation

$$\begin{aligned} \mathcal{D} &= [0.08382, 0.3323, 0.0, -0.3323, -0.08382] \\ \mathcal{S} &= [0.02342, 0.2416, 0.4700, 0.2416, 0.02342] \end{aligned} \quad (14)$$

$J$  is then converted into a diffusion tensor via Eq. 9 and

$$\rho(\mu) = \begin{cases} 1 & \text{for } \mu \leq \sigma \\ 1 - \exp\left(\frac{-\mu}{\mu - \sigma}\right) & \text{else} \end{cases} \quad (15)$$

as proposed in [19].

### 3 Diffusion Extended

#### 3.1 General Linear Models

The single orientation model  $\nabla^T g \mathbf{u} = 0$  (cmp. Eq. 2) above can be rewritten as

$$(D^T g) \mathbf{p} = 0 \quad (16)$$

where  $\mathbf{p}$  is the parameter vector and  $D$  is an operator vector applied to the data  $g$ . In Eq. 2 we have  $\mathbf{p} = \mathbf{u}$  and  $D = \nabla$ . For any linear model that can be written in the form Eq. 16, we can construct a diffusion-like reconstruction scheme as done for the single orientation model above via the cost function (cmp. Eq. 3)

$$E(g) = \int_{\Omega} (g - f)^2 d\mathbf{x} + \alpha \int_{\Omega} \langle ((D^T g) \mathbf{p})^2 \rangle d\mathbf{x} \quad (17)$$

As above the solution  $\tilde{g}$  minimizing  $E$  has to fulfill the variational condition  $\delta E(\tilde{g}) = 0$  (cmp. Eq. 4)

$$\delta E(g) = \int_{\Omega} 2(g - f) \delta g d\mathbf{x} + \alpha \int_{\Omega} \langle 2(D^T g \mathbf{p}) \mathbf{p}^T \rangle D \delta g d\mathbf{x} \quad (18)$$

The expression  $D \delta g$  can not be evaluated. But we can express  $D$  via a series of partial derivative operators (or equivalently express  $Dg$  as a Taylor series,  $D$  and  $\delta$  commute). For each term in the series we get a term in the integral which we partially integrate. For each partial integration step the sign of the respective

term flips. Thus terms with odd derivatives have negative sign, terms with even derivatives have positive sign. This behavior can be reproduced by mirroring of each derivative operator, i.e. by mirroring  $D$  around its center (cmp. [27] eq. 14). Thus the corresponding Euler-Lagrange equation  $\tilde{g}$  has to fulfill is

$$0 = (\tilde{g} - f) - \alpha \bar{D}^T \langle \mathbf{p}\mathbf{p}^T \rangle D \tilde{g} \quad (19)$$

where  $\bar{D}$  is a mirrored version of  $D$ . The expression  $\langle \mathbf{p}\mathbf{p}^T \rangle$  again is a square, symmetric, positive semidefinite matrix. Analog to the approach in Sec. 2.2 we can approximate it via the extended structure tensor  $J_D$

$$J_D(\mathbf{x}) = \int w(\mathbf{x} - \mathbf{x}') (Dg(\mathbf{x}')) (D^T g(\mathbf{x}')) d\mathbf{x}' \quad (20)$$

belonging to the model  $(D^T g)\mathbf{p} = 0$  and exchange the eigenvalues  $\mu_i$  of  $J_D$  via Eq. 9.

### 3.2 Diffusion with Two Orientations

In this example, we construct a diffusion-like scheme for enhancement of transparently overlaid structures resulting in two local orientations in 2d data. The linear model describing this is (cmp. [22], eq. 11)

$$\partial_{xx} g p_1 + \partial_{xy} g p_2 + \partial_{yy} g p_3 = 0 \quad \text{or} \quad D^T g \mathbf{p} = 0 \quad (21)$$

where  $D = (\partial_{xx}, \partial_{xy}, \partial_{yy})^T$  is an operator vector containing second order partial derivative operators,  $g$  is the image data and  $\mathbf{p}$  is a parameter vector (containing mixed orientation parameters, that we do *not* need to disentangle, cmp. [22]). This operator vector  $D$  is now plugged into Eq. 19. Further  $\langle \mathbf{p}\mathbf{p}^T \rangle$  in that equation is replaced by the extended structure tensor from Eq. 20 with eigenvalues exchanged via Eq. 9 and edge stopping function Eq. 15.

As above the discretization of  $D$  is done using separable convolution filters

$$\partial_{xx} = \mathcal{L}_x * \mathcal{S}_y \quad \partial_{xy} = \mathcal{D}_x * \mathcal{D}_y \quad \partial_{yy} = \mathcal{S}_x * \mathcal{L}_y \quad (22)$$

where lower indices give the application direction,  $\mathcal{L}$  is a discrete 1d second order derivative,  $\mathcal{S}$  is a 1d smoother and  $\mathcal{D}$  is a 1d first order derivative filter.

For the calculation of  $D$  and  $\bar{D}$  in Eq. 19 we investigate 2 schemes. The first one we call  $2 \times 2$ , because the first order derivative and smoother are 2-tab filters

$$\mathcal{L} = [1, -2, 1] \quad \mathcal{D} = [1, -1, 0] \quad \mathcal{S} = [0.5, 0.5, 0] \quad (23)$$

We write the 2-tab filters as 3-tab filters in order to have the storage point (middle of the filter) well defined, also for the mirrored version  $\bar{D}$ . The second scheme we call  $3 \times 3$

$$\mathcal{L} = [1, -2, 1] \quad \mathcal{D} = [0.5, 0, -0.5] \quad \mathcal{S} = [0.21478, 0.57044, 0.21478] \quad (24)$$

The extended structure tensor  $J_D$  (Eq. 20) needed to approximate  $\langle \mathbf{p}\mathbf{p}^T \rangle$  in Eq. 19 is calculated using  $5 \times 5$  filters optimized for accurate double orientation estimation

$$\begin{aligned} \mathcal{L} &= [0.2068, 0.1729, -0.7593, 0.1729, 0.2068] \\ \mathcal{D} &= [0.06295, 0.3741, 0, -0.3741, -0.06295] \\ \mathcal{S} &= [0.01531, 0.2316, 0.5062, 0.2316, 0.01531] \end{aligned} \quad (25)$$

The resulting update scheme is defined below in Eq. 26.

## 4 Experiments

In this section we compare denoising results using double orientation schemes (Sec. 3.2) and standard anisotropic diffusion schemes (Sec. 2.3).

### 4.1 Approach

In the variational derivations above we showed how to construct a data term and a smoothness term. Using the data term together with a quadratic smoothness term yields a convex regularizer with a single global minimum. In contrast to this, it is also common in the diffusion community to omit the data term and use an Euler-forward update scheme

$$g^{t+\tau} = g^t - \tau \bar{D}^T \langle \mathbf{p} \mathbf{p}^T \rangle D \tilde{g}^t \quad (26)$$

where the upper index  $t$  indicates the diffusion time and  $\tau$  is a (small) time step (cmp. Eq. 7). Then the initial data  $g^0$  has to be the noisy (or otherwise corrupted) data  $f$  and the smoothing process has to be stopped manually or due to some criterion. In the experiments shown here we do not use the data term and run the smoothing process as long as the peak signal to noise ratio (PSNR, cmp. [15])

$$PSNR = 20 \log \frac{255}{\|g - g_0\|_2} \quad (27)$$

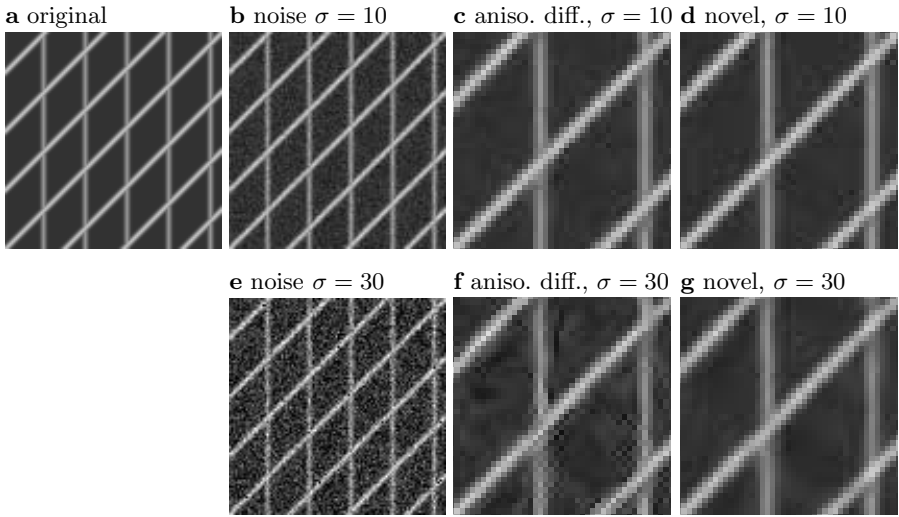
raises. This is practicable because in all experiments we have ground truth data  $g_0$  (not to be confused with initial noisy image  $g^0$  from Eq. 26) available. The parameters in the  $\rho$ -function (Eq. 15) were  $\sigma = 1e-8$  (fixed) and  $c$  was optimized using gradient ascent on the PSNR, starting with  $c = 15$ .

### 4.2 Results

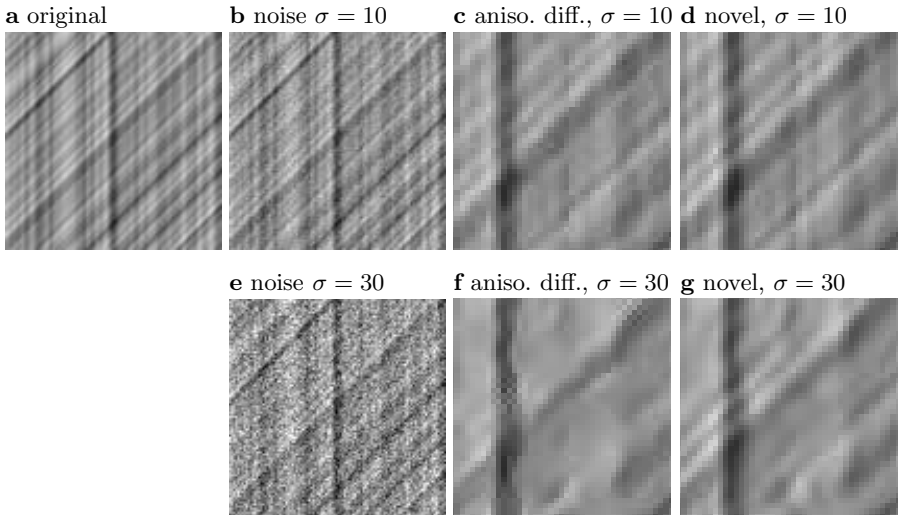
We focus on denoising of data where a linear model describing the signal is available, namely 2d double orientation signals. All images are 8bit scalar data in the range interval  $[0, 255]$ .

The first image we focus on is a synthetic image, where vertical and diagonal lines are non-transparently superimposed (see Fig. 1a). We added Gaussian noise with standard deviation  $\sigma \in \{10, 20, 30, 40, 50, 60\}$  (see e.g. Fig. 1b,e). We then applied standard anisotropic diffusion ( $2 \times 2$  and  $3 \times 3$  schemes) and our novel diffusion using the double orientation model in 2d (also  $2 \times 2$  and  $3 \times 3$  schemes). The resulting PSNR values are shown in Tab. 1. We observe, that standard anisotropic diffusion gives best results when using the  $2 \times 2$ -scheme. The novel scheme gives best results when using the  $3 \times 3$ -scheme. In addition the novel schemes outperform standard anisotropic diffusion, the more, the more structural information is available, i.e. for low noise levels. Thus for smoothing of X-junctions the novel scheme outperforms diffusion.

This is even more prominent, when the transparent overlay model is completely fulfilled (see Fig. 2a). Exemplarily we show the reconstructions and optimum PSNR values (Fig. 2c-f) for noise levels  $\sigma = 10$  (PSNR=28.1, cmp. Fig. 2b) and  $\sigma = 30$  (PSNR=18.7, cmp. Fig. 2e).



**Fig. 1.** Nontransparently superimposed lines. **a** Original image, **b,e** noise added, **c,f** reconstruction using standard anisotropic diffusion ( $2 \times 2$  scheme, zoomed in for better display), **d,g** reconstruction using the novel scheme ( $3 \times 3$  scheme, zoomed in for better display).



**Fig. 2.** Transparently superimposed 1d structures. **a** Original image, **b,e** noise added ( $\sigma = 10$ , PSNR=28.1 and  $\sigma = 30$ , PSNR=18.7), **c,f** reconstruction using standard anisotropic diffusion ( $2 \times 2$  scheme, zoomed in for better display, **c** PSNR=31.3, **f** PSNR=25.7), **d,g** reconstruction using the novel scheme ( $3 \times 3$  scheme, zoomed in for better display, **d** PSNR=35.1, **g** PSNR=27.3).

**Table 1.** PSNR values for all tested schemes and all noise levels added to the image in Fig. 1a

noise (PSNR)	aniso. diff.		new scheme	
	$2 \times 2$	$3 \times 3$	$2 \times 2$	$3 \times 3$
$\sigma = 10$ (28.1)	33.1	31.2	34.1	36.0
$\sigma = 20$ (22.1)	29.1	26.9	29.4	31.8
$\sigma = 30$ (18.7)	26.7	24.4	26.6	28.9
$\sigma = 40$ (16.2)	24.1	22.4	23.9	25.7
$\sigma = 50$ (14.6)	22.5	20.9	22.2	23.6
$\sigma = 60$ (13.2)	20.8	19.5	20.5	21.5

## 5 Summary, Conclusions and Future Work

In this paper we introduced diffusion-like schemes respecting general linear models. These schemes are derived via a variational approach using expectation values of parameter vectors in cost functions in a novel way. The resulting diffusion tensor or extended diffusion tensor  $\langle \mathbf{pp}^T \rangle$  has been shown to be well approximated by a diffusion tensor derived via a structure tensor (as common in literature). Using this framework, we constructed a reconstruction scheme respecting two orientations simultaneously. When data belongs to this model, e.g. X-junctions or edges in transparent overlays, the proposed scheme outperforms anisotropic diffusion. Y-junctions, more general texture or 'natural images' have to be modeled by richer models. Our two orientations approach therefore does not outperform specialized state of the art denoising schemes like [15]. But we want to emphasize that diffusion-like processes not only can be used for denoising, but also for other applications as stated in the introduction (cmp. e.g. [24]). Results shown in Sec. 4 demonstrate the good performance of the novel algorithms for the special case of double orientations. What is more, they are a proof of concept backing the new theory.

In usual data most areas show no or single orientations, two or more orientations are less prominent. Thus, in future work, we will look into model selection and switching, mainly for speed up. In addition, the proposed model is a 2d two orientations model which we will extend to 2d three orientations (for texture) and 3d two orientations (for transparent motion). The usage of other models, like motion and brightness changes will also be of interest.

## References

1. S.T. Acton. Multigrid anisotropic diffusion. *Trans. Im. Proc.*, 7:280–291, 1998.
2. J.L. Barron, D.J. Fleet, and S.S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77, 1994.
3. J. Bigün and G. H. Granlund. Optimal orientation detection of linear symmetry. In *ICCV*, pages 433–438, London, UK, 1987.

4. M. J. Black, G. Sapiro, D. H. Marimont, and D. Heeger. Robust anisotropic diffusion. *IEEE Transactions on Image Processing*, 7(3):412–432, March 1998.
5. M. Felsberg, P.-E. Forssén, and H. Scharr. Channel smoothing: Efficient robust smoothing of low-level signal features. *PAMI*, 28(2), 2006.
6. D.J. Fleet, M.J. Black, Y. Yacoob, and A.D. Jepson. Design and use of linear models for image motion analysis. *Int. J. Computer Vision*, 36(3):171–193, 2000.
7. J. Fröhlich and J. Weickert. Image processing using a wavelet algorithm for nonlinear diffusion. Technical Report 104, Laboratory of Technomathematics, University of Kaiserslautern, P.O. Box 3049, 67653 Kaiserslautern, Germany, 1994.
8. H. Haußecker and D. J. Fleet. Computing optical flow with physical models of brightness variation. *PAMI*, 23(6):661–673, June 2001.
9. B. Jähne. *Spatio-temporal image processing*, volume 751. Springer, Berlin, 1993.
10. C. Mota, I. Stuke, and E. Barth. Analytic solutions for multiple motions. In *Proc. ICIP*, pages 917–920, 2001.
11. D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Comm. Pure Appl. Math. XLII*, pages 577–685, 1989.
12. O. Nestares, D. J. Fleet, and D. Heeger. Likelihood functions and confidence bounds for total-least-squares problems. In *CVPR'00*, volume 1, 2000.
13. M. Nielsen, P. Johansen, O.F. Olsen, and J. Weickert, editors. *Scale-space theories in computer vision, LNCS*, volume 1682. Springer, Berlin, 1999.
14. P. Perona and J. Malik. Scale space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12:629–639, 1990.
15. J. Portilla, V. Strela, M.J. Wainwright, and E.P. Simoncelli. Image denoising using scale mixtures of gaussians in the wavelet domain. *IEEE TIP*, 12(11), 2003.
16. T. Preußner and M. Rumpf. An adaptive finite element method for large scale image processing. In *Scale-space theories in computer vision, Lecture Notes in Computer Science*, volume 1682, pages 223–234, Berlin, 1999. Springer.
17. S. Roth and M.J. Black. Fields of experts: A framework for learning image priors. In *CVPR*, volume 2, pages 860–867, 2005.
18. H. Scharr, M.J. Black, and H.W. Haussecker. Image statistics and anisotropic diffusion. In *ICCV 2003*, Nice, France, 2003.
19. H. Scharr and H. Spies. Accurate optical flow in noisy image sequences using flow adapted anisotropic diffusion. *Signal Processing: Image Communication*, 2005.
20. C. Schnörr. A study of a convex variational diffusion approach for image segmentation and feature extraction. *J. Math. Im. and Vis.*, 8(3):271–292, 1998.
21. M. Shizawa and K. Mase. Simultaneous multiple optical flow estimation. In *ICPR'90*, pages 274–278, 1990.
22. I. Stuke, T. Aach, E. Barth, and C. Mota. Analysing superimposed oriented patterns. In *6th IEEE SSIAI*, pages 133–137, 2004.
23. C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *ICCV*, Bombay, India, 1998.
24. D. Tschumperle and R. Deriche. Vector-valued image regularization with pde's: A common framework for different applications. In *CVPR*, 2003.
25. J. Weickert. *Anisotropic diffusion in image processing*. Teubner, Stuttgart, 1998.
26. J. Weickert and C. Schnörr. A theoretical framework for convex regularizers in pde-based computation of image motion. *IJCV*, pages 245–264, Dec. 2001.
27. S.C. Zhu and D. Mumford. Prior learning and gibbs reaction-diffusion. *PAMI*, 19(11):1236–1250, 1997.

# Reduction of Ring Artifacts in High Resolution X-Ray Microtomography Images

Maria Axelsson, Stina Svensson, and Gunilla Borgefors

Centre for Image Analysis, Swedish University of Agricultural Sciences,  
Lägerhyddsvägen 3, SE-75237 Uppsala, Sweden  
{maria,stina,gunilla}@cb.uu.se

**Abstract.** Ring artifacts can occur in reconstructed images from X-ray microtomography as full or partial circles centred on the rotation axis. In this paper, a 2D method is proposed that reduces these ring artifacts in the reconstructed images. The method consists of two main parts. First, the artifacts are localised in the image using local orientation estimation of the image structures and filtering to find ring patterns in the orientation information. Second, the map of the located artifacts is used to calculate a correction image using normalised convolution. The method is evaluated on 2D images from volume data of paper fibre imaged at the European Synchrotron Radiation Facility (ESRF) with high resolution X-ray microtomography. The results show that the proposed method reduces the artifacts and restores the pixel values for all types of partial and complete ring artifacts where the signal is not completely saturated.

## 1 Introduction

In X-ray microtomography images from third generation scanning systems, ring artifacts can sometimes occur as partial or full rings centred on the rotation axis in the reconstructed volumes. The ring artifacts are superimposed on the original pixel values and both positive and negative erroneous values influence further processing and measurements in the volumes.

Ring artifacts can originate from a number of sources. The rings can be due to defective detector elements or shifts in the output from individual detector elements or sets of detectors [1]. The third generation scanning systems are highly sensitive to detector flaws because the same detector element measures the signal at a given radius. Ring artifacts can also occur due to variations or imperfections in the incoming beam [2], or be due to variations in the beam together with effects of the point spread function of the detector elements [3].

Calibration of the tomography system is necessary for the ring artifacts to be reduced in the reconstructed images. A common method to reduce the ring artifacts is flat-field correction [1,3], where the radiographs are corrected using one image of the beam without the sample and one dark image without the beam. This method is often not enough to reduce the rings fully. Some tomography systems use translations of the detector to reduce the part of the ring artifacts that

is due to the defective detector elements [4]. A method using time-delay integration to remove the problems with the miscalibrations is described in [5]. The ring artifacts can also be reduced by filtering methods before the reconstruction of the images [6,7].

Methods for reduction of the ring artifacts in the reconstructed images are rarely described in the literature, but one such method is presented in [8], where the reconstructed image is transformed to polar coordinates and the ring artifacts occur as line artifacts. Correction is done in polar coordinates with no distinction between pixels with and without artifacts. The rings are also assumed to be full circles.

In this paper, a method that reduces both full and partial ring artifacts in X-ray microtomography images is presented. The method is applied on the reconstructed images. The pixels that are affected by ring artifacts are first detected and the erroneous pixel values are then adjusted to reduce the ring artifacts. Calculation of a correction image is done in polar coordinates for fast calculation, but the original image resolution is preserved, as the correction part of the method is done in the original pixel coordinates. The important features of this method are that only pixels with estimated artifacts are altered in the correction step, that it uses local correction to remove both partial and full ring artifacts and that it does not alter the original resolution of the image through interpolation. As the artifacts occur on planes perpendicular to the rotation axis of the scanning system, the proposed method is in 2D. The method assumes a known centre of rotation from the reconstruction of the image data and that the image structures has a faster variation than the ring artifacts when following a circle with a given radius.

The correction part of the proposed method can be used as a stand alone method to remove ring artifacts. It will give a reasonable ring artifact reduction if it can be assumed that all pixels on any circle contain ring artifacts and that the contrast of the depicted object is not sensitive to alterations of the original pixel values. This reduced method is not applicable if a restoration to original pixel values is the aim of the ring artifact reduction.

## 2 Method

The method consists of two main parts, finding the ring artifacts and correcting them. It can be further decomposed into the following steps:

- Estimate the local orientation in the image, using a representation of orientation with a structure tensor in each pixel. (Section 2.1)
- Find orientations in the tensor field that correspond to circular patterns around the known rotation axis and create a certainty map with the probability of each pixel to contain a ring artifact. (Section 2.2)
- Calculate a correction image from the certainty map and the original image using normalised convolution. (Section 2.3)
- Correct the original image by removing the estimated correction image. (Section 2.4)



For some illustrative experiments and how to choose the parameters for specific applications, see Section 3.

## 2.1 Orientation Estimation

The first step in the method is local orientation estimation in all pixels in the image. Local orientation derived from a pixel neighbourhood can be represented with an orientation tensor. The tensor representation that is used here is thoroughly described in [9]. The orientation is defined as the main direction in which the signal varies in a neighbourhood around a pixel.

Each neighbourhood is assumed to contain signal energy in only one direction, i.e., the signal is assumed to be locally *simple*. The image is convolved with a set of phase invariant quadrature filters. The filter outputs are combined into a tensor, which is a  $2 \times 2$  symmetrical matrix, and the eigensystem of this tensor describe the local orientation in the neighbourhood. The eigenvector with the largest corresponding eigenvalue represents the estimated local orientation. If the signal in the neighbourhood is not *simple*, the tensor will not have one single dominant orientation, but be more isotropic.

A certainty value for the local orientation in each neighbourhood can be estimated using the isotropy of the tensor, i.e., the strength of the local orientation. The certainty value  $c_t$ ,

$$0 \leq c_t = \frac{l_1 - l_2}{l_1} \leq 1 \quad (1)$$

is calculated for each tensor, where  $l_1$  is the largest eigenvalue and  $l_2$  is the second largest eigenvalue of the tensor eigensystem. A large value indicates strong directional information and a small value indicates weak directional information.

## 2.2 Artifact Identification

Ring artifacts can be identified in the tensor field using the first eigenvector, that represents the local orientation in the pixel, and the corresponding certainty value,  $c_t$ . Since the centre of rotation is known and the ring artifacts are centred around this point, this can be used to find patterns corresponding to ring artifacts. A vector field with normalised vectors from the centre of rotation is created. The rings can be found where the absolute value of the scalar product between each of these vectors and the first eigenvector in the corresponding pixel gives a high response. Since the scalar product is a cosine function, it is sharpened to reduce high responses from structures that are not oriented along part of a circle as,

$$c_r = |\hat{\mathbf{r}} \cdot \hat{\mathbf{e}}_1|^n, \quad (2)$$

where  $\hat{\mathbf{r}}$  is the normalised vector from the centre of rotation,  $\hat{\mathbf{e}}_1$  is the normalised first eigenvector of the tensor in the pixel and  $n$  is a positive scalar. To further reduce high responses from areas where the estimated orientation is not strong,  $c_r$  is multiplied with the corresponding certainty value of the orientation in the

pixel,  $c_t$ , to generate a certainty map. This certainty map contains a certainty value for each pixel that corresponds to the probability of the pixel to be influenced by a ring artifact.

The certainty map is transformed to polar coordinates using the same image resolution for the radial direction as for the Cartesian coordinates and dense interpolation in the angular direction to preserve the resolution of the pixels that are furthest from the centre of rotation. The certainty map is smoothed using a one pixel wide kernel in the radial direction and a fixed length in Cartesian coordinates for the kernel length in the angular direction. The smoothing is performed to ensure that pixels with both ring artifacts and actual image structures is included in the calculation of the correction image, even though these local neighbourhoods may contain weak directional information due to contradicting information about the local orientation.

### 2.3 Correction Image Generation

A correction image for the ring artifacts is created in polar coordinates using normalised convolution [9,10]. The normalised convolution uses the smoothed certainty map in polar coordinates,  $\tilde{\mathbf{C}}$ , to create the correction image for the ring artifacts as,

$$\tilde{\mathbf{I}}_{err} = \frac{\mathbf{a} * \tilde{\mathbf{C}}\tilde{\mathbf{I}}_{in}}{\mathbf{a} * \tilde{\mathbf{C}}}, \quad (3)$$

where  $\tilde{\mathbf{I}}_{err}$  is the correction image,  $\tilde{\mathbf{I}}_{in}$  is the input image and  $\mathbf{a}$  is an applicability function, all in polar coordinates. The original image is interpolated in the same manner as the certainty map.

The applicability function used in this step is a kernel that is one pixel wide in the radial direction and has fixed length in Cartesian coordinates for the length in the angular direction. The same filter as in Section 2.2 can be used. The kernel must be longer than the image structures it is used to enhance, to ensure that the mean value is that of the artifact and not the local mean of any real object. Partial artifacts are not reduced well if the filter is too long. The pixels close to the centre of rotation must be treated separately, using shorter filters, if partial artifacts are to be reduced for these pixels. For full ring artifacts this is not a problem, since the kernel will cover the full circle.

The normalised convolution can be used as a stand alone method to remove the ring artifacts if an actual restoration of the image intensities is not necessary. If the certainty map is set to one for all pixels, the normalised convolution is reduced to regular convolution along the radii. In that case, the correction image will not only be calculated using pixels that contain artifacts but of all pixels covered by the applicability function. Hence the correction step will also correct all pixels regardless of their certainty to contain an artifact.

### 2.4 Artifact Correction

The estimated correction image and the certainty map are transformed to Cartesian coordinates before the correction step, here denoted  $\mathbf{I}_{err}$  and  $\mathbf{C}$ . As the ring

artifacts are superimposed on the original image,  $\mathbf{I}_{in}$ , each pixel value can be corrected by subtracting the estimated error as,

$$i_{corr} = i_{in} - c(i_{err} - m) \quad (4)$$

where  $c$  is the pixel value in the certainty map  $\mathbf{C}$  and  $i_{corr}$ ,  $i_{in}$  and  $i_{err}$  denotes the pixel values in the respective images. The mean value of the ring artifacts,  $m$ , is normally set to half the dynamic range of the image. In this step, the multiplication by  $c$  is done to ensure areas without artifacts are not changed. The certainty values can be mapped towards higher values before the correction, for a full recovery of the ring artifacts. If the mapping is not used, weak rings are often not well reduced.

### 3 Experiments and Results

The proposed method is evaluated on a set of microtomographic images. How to choose parameters when using the proposed method is presented in Section 3.2 and examples of results are found in Section 3.3.

#### 3.1 The Images

The test images used for evaluation of the method are 2D slices from a set of volume images of paper scanned at the European Synchrotron Radiation Facility (ESRF) in Grenoble, France. The paper samples were imaged in absorption mode at beamline ID19 using synchrotron X-ray microtomography. The size of a volume element (voxel) in the 3D images is  $0.7\mu\text{m} \times 0.7\mu\text{m} \times 0.7\mu\text{m}$ . A cylindrical region of interest, see Figure 1, with a diameter of 1.43mm and a height dependent on the sample thickness was imaged for each sample. The resolution of the images is  $2048 \times 2048 \times$  the sample thickness. The dynamic range of the images is 8 bits. All volume images contain ring artifacts to different degrees, from partial to full ring artifacts seen as wave-like patterns in slices perpendicular to the rotation axis, see Figure 1 and 2 (left column). In the set of 2D images used for the evaluation of the method both images with and without artifacts are present.

#### 3.2 Parameters for Images of Paper Fibre

The parameters in the proposed method are adapted to the image of interest using some prior information about the image. The knowledge that is needed for adapting the parameters are the centre of rotation from the reconstruction of the image and the approximate width of the ring artifacts in the radial direction, measured in pixels. The model of the rings also assumes a slower variation of the rings in the angular direction than of the actual image data.

Four quadrature filters were used in the experiments. The filters were optimised [9] to generate small errors in both the spatial and the Fourier domain.

The approximate size of the ring artifacts is used to choose a proper bandwidth and centre frequency for the quadrature filters in the orientation estimation. A relative bandwidth of 3 octaves and a centre frequency of  $\pi/6$  radians were used for the radial function of the quadrature filters. The scalar  $n$  was set to 10. This parameter is also chosen dependent on the size of the ring artifacts. If the ring artifacts are strong and the orientation estimates give high answers for all the artifacts,  $n$  should be large to reduce high answers from other structures which are partly aligned as the ring artifacts. If  $n$  is smaller the certainty map is more blurred but the method can correct weak artifacts with low response from the quadrature filters better.

The length of the applicability function,  $\mathbf{a}$ , in the experiments was 300 pixels in Cartesian coordinates. The length is a trade off between a good approximation of the artifact and an effective reduction of partial rings. The same applicability function was used to smooth the certainty map before the normalised convolution. For the radii close to the origin all kernel lengths in the angular direction longer than half the size of the image in the angular direction was truncated to the same length.

In the correction step, the certainty map was mapped towards higher certainties to allow the original pixel values to change more and make sure that all of the detected ring artifacts was removed. This also allows for some of the artifact free pixels to be slightly altered. Here multiplication by a factor 3 and truncation to 1 for values above 1 was used as the mapping.

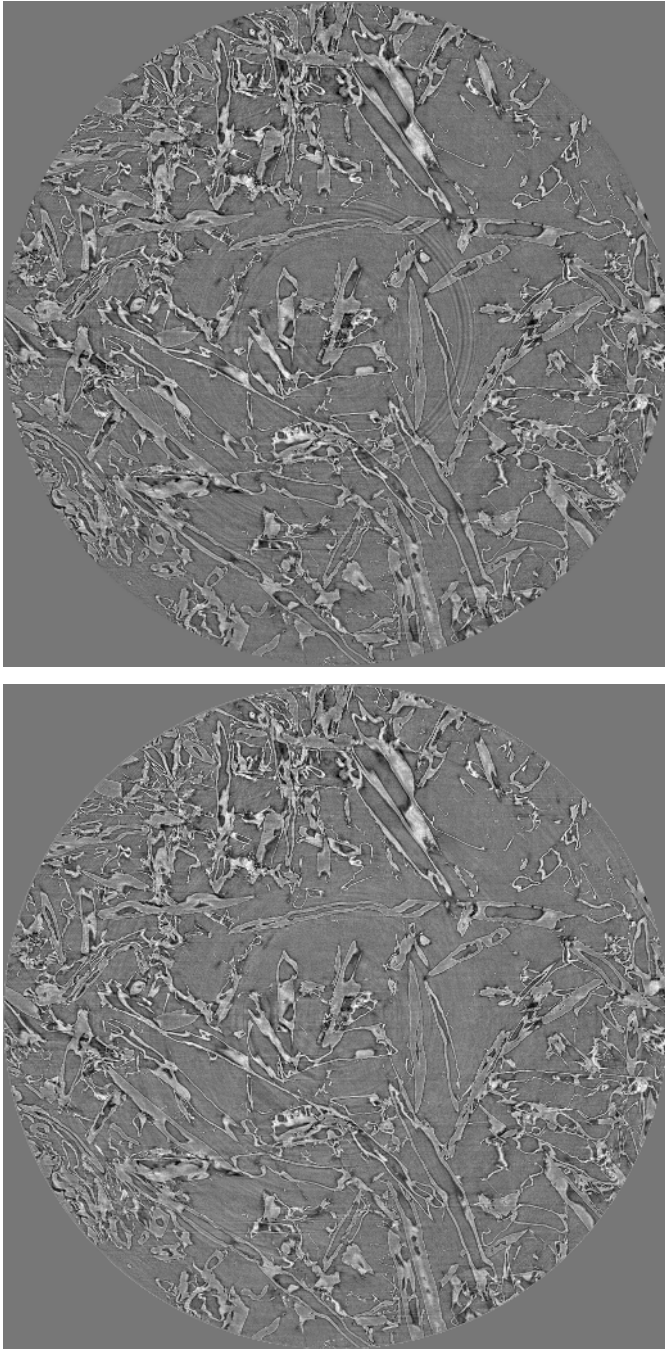
The parameter  $m$  was set to half the dynamic range of the image, in this case 128. It can also be set to the mean value of the image intensities in the volume if half the dynamic range is far from the actual mean of the artifacts.

When adjusting the parameters for new data sets using images with the same resolution of the image structures, the length of the applicability function and the mapping of the certainty values are the two parameters that are used to tune the method for better performance. None of the parameters presented here is sensitive to changes and experiments with similar parameter setups also yield a good reduction of the ring artifacts.

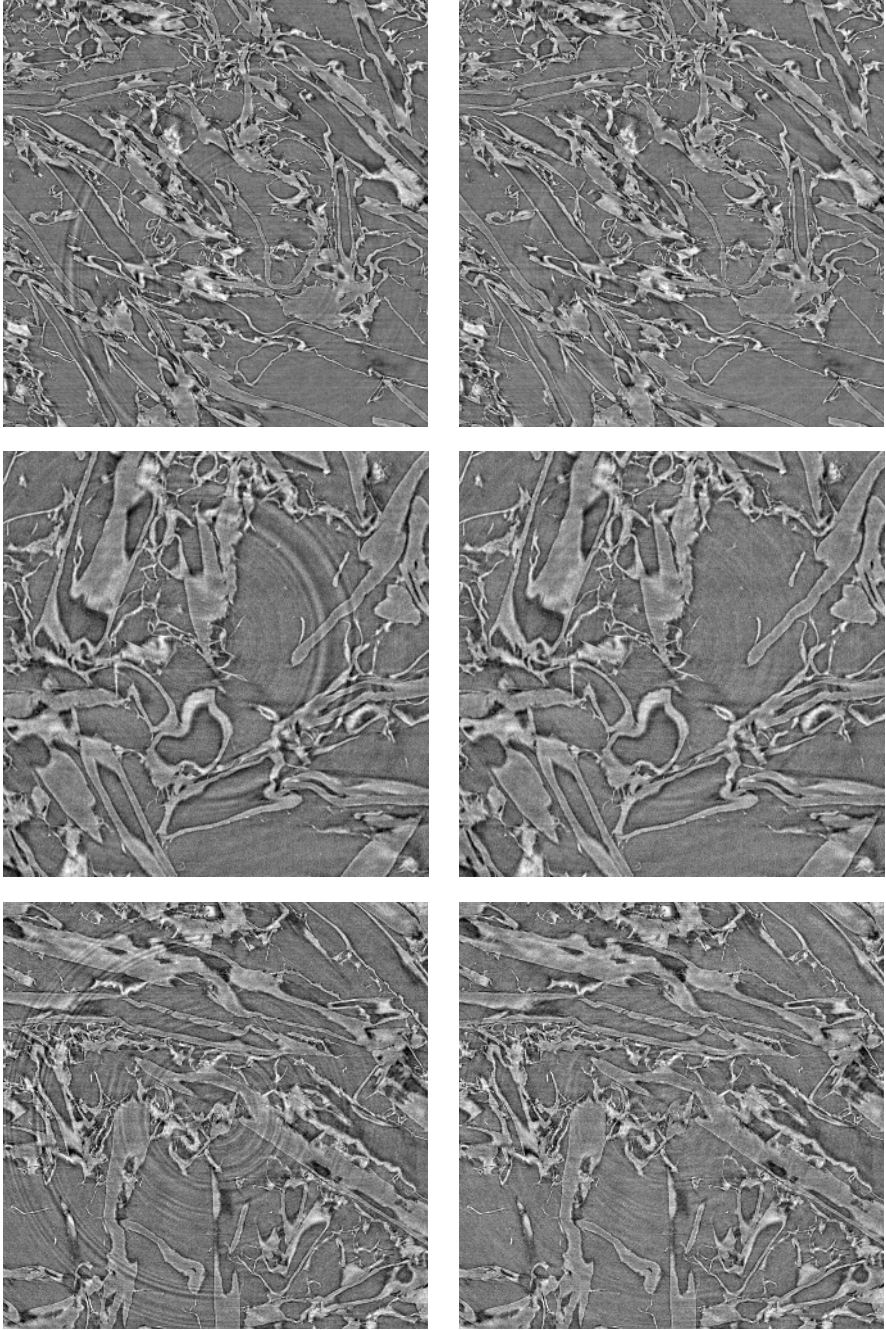
### 3.3 Results

Results from the experiments using the proposed method for ring artifact reduction with the parameters from Section 3.2 is seen in Figure 1 and 2. The original images are to the left and the filtered images with reduced ring artifacts to the right. The proposed method reduces the rings and preserves the image contrast. Structures that have the same local orientation as the ring artifacts can be affected by the method, but as can be seen here it is not a problem for this type of images.

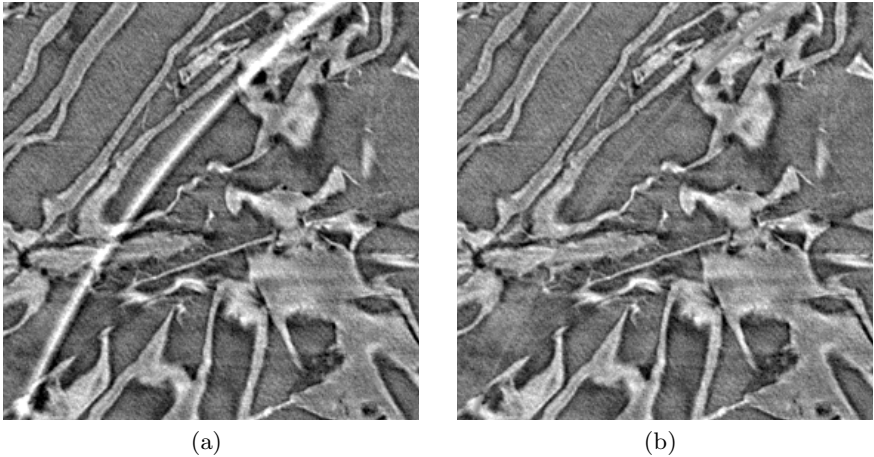
Figure 3 shows an example of a partly saturated ring artifact. This type of artifact is also reduced by the proposed method, but restoration to original pixel values is not possible for all parts of the artifact, since the saturation of the image has caused a loss of information in some pixels.



**Fig. 1.** (Top) A slice perpendicular to the rotation axis with ring artifacts. (Bottom) Corrected ring artifacts using the proposed method.



**Fig. 2.** (Left column) Ring artifacts occurring in slices perpendicular to the rotation axis in reconstructed volumes of microtomography images of paper. Note that these images are cropped. (Right column) Corrected ring artifacts using the proposed method.



**Fig. 3.** (a) A partly saturated ring artifact occurring in a microtomography image of paper. (b) Corrected image. As can be seen, the loss of information due to the saturation limits the restoration to original pixel values.

## 4 Discussion

An automatic 2D image analysis method for reduction of full and partial ring artifacts in reconstructed high resolution X-ray microtomography images is proposed in this paper. The method can be used not only for X-ray microtomography images but for any projection reconstruction data with ring artifacts. The centre of rotation must be known and the ring artifacts are assumed to have a slower variation than the image structures when following a circle with a given radius.

The method is evaluated on a set of 2D slices from volume images of paper fibre imaged with X-ray microtomography. As can be seen in Section 3.3, the method performs well on different types of ring artifacts that occur in these volumes. One possible drawback of the method is that image structures having a local orientation equal to the ring artifacts can be affected. The smoothing of the certainty map before the normalised convolution reduces the probability of these false corrections. As can be seen from the results, the original image structures are well preserved for further image processing and measurements.

The proposed method is fast since it uses linear filtering in polar coordinates to build the correction image. The same procedure in Cartesian coordinates is more time consuming. The main features of the proposed method are local correction to remove both partial and full rings, no interpolation of the original image data, and correction only of pixels that contain ring artifacts.

As some pixels in the partly saturated ring artifacts have lost all the information about the original depicted object, these pixels can not be recovered by the proposed method. Pixels that are close to saturation are also difficult to recover

correctly. This problem needs to be addressed in future work. Pixel values from neighbouring pixels could be used for restoration of these values using normalised convolution in 3D. This is feasible if the image is assumed to be varying slowly compared to the saturated artifact width in the radial direction.

**Acknowledgements.** Financial support and volume image data from the ESRF Long Term project ME-704 is gratefully acknowledged. Financial support from the EU project AEP QLK5-CT-2002-00772 and the Paper and Fibre Research Institute (PFI) in Trondheim is also gratefully acknowledged.

## References

1. Vidal, F.P., Letang, J.M., Peix, G., Cloetens, P.: Investigation of artefact sources in synchrotron microtomography via virtual X-ray imaging. *Nuclear Instruments and Methods in Physics Research B* **234** (2005) 333–348
2. Antoine, C., Nygård, P., Gregersen, Ø.W., Holmstad, R., Weitkamp, T., Rau, C.: 3D images of paper obtained by phase-contrast X-ray microtomography: image quality and binarisation. *Nuclear Instruments and Methods in Physics Research A* **490** (2002) 392–402
3. Cloetens, P.: Contribution to Phase Contrast Imaging, Reconstruction and Tomography with Hard Synchrotron Radiation – Principles, Implementation and Applications. PhD thesis, Vrije Universiteit Brussel (1999)
4. Jenneson, P., Gilboy, W., E.J, M., P.J., G.: An X-ray micro-tomography system optimised for the low-dose study of living organisms. *Applied Radiation and Isotopes* **58** (2003) 177–181
5. Davis, G.R., Elliott, J.C.: X-ray microtomography scanner using time-delay integration for elimination of ring artefacts in the reconstructed image. *Nuclear Instruments and Methods in Physics Research A* **394** (1997) 157–162
6. Raven, C.: Numerical removal of ring artifacts in microtomography. *Review of Scientific instruments* **69**(8) (1998) 2978–2980
7. Tang, X., Ning, R., Yu, R., Conover, D.: Cone beam volume CT image artifacts caused by defective cells in x-ray flat panel imagers and the artifact removal using a wavelet-analysis-based algorithm. *Medical Physics* **28**(5) (2001) 812–825
8. Sijbers, J., Postnov, A.: Reduction of ring artefacts in high resolution micro-CT reconstructions. *Physics in Medicine and Biology* **49**(14) (2004) N247–N253
9. Granlund, G.H., Knutsson, H.: *Signal Processing for Computer Vision*. Kluwer Academic Publishers (1995) ISBN 0-7923-9530-1.
10. Knutsson, H., Westin, C.F.: Normalized and differential convolution: Methods for interpolation and filtering of incomplete and uncertain data. In: *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. (1993) 515–523



# A Probabilistic Multi-phase Model for Variational Image Segmentation<sup>\*</sup>

Thomas Pock and Horst Bischof

Institute for Computer Graphics and Vision, Graz University of Technology  
Infeldgasse 16/2, A-8010 Graz, Austria  
{pock, bischof}@icg.tugraz.at  
<http://www.icg.tugraz.at>

**Abstract.** Recently, the *Phase Field Method* has shown to be a powerful tool for variational image segmentation. In this paper, we present a novel multi-phase model for probability based image segmentation. By interpreting the phase fields as probabilities of pixels belonging to a certain phase, we obtain the model formulation by maximizing the mutual information between image features and the phase fields. For optimizing the model, we derive the Euler Lagrange equations and present their efficient implementation by using a narrow band scheme. We present experimental results on segmenting synthetic, medical and natural images.

## 1 Introduction

Image segmentation is one of the best studied but still unsolved low level vision problems. Its primal goal is to partition a given image domain  $\Omega$  into  $K$  non-overlapping regions  $\Omega_i$ ,  $i = 1 \dots K$ , so that the objects covered by each region share some specific properties.

The active contour or snake model, introduced by Kass Witkin and Terzopoulos [1], has shown to be a powerful tool for image segmentation. The major (technical) difficulty of active contours is to properly represent the interface of the contour. Basically, there are two concepts for interface representation. In an explicit representation, the points belonging to the interface are spatially sampled. Although this representation is very efficient in both memory and computational terms, it has drawbacks with respect to topological changes and strong curvatures. In an implicit representation, the interface is defined as the isocontour of a higher-dimensional function. Hence, topological changes can be handled in a very natural way [2].

The *Level Set Method*, introduced by Osher and Sethian [3] is based on such an implicit representation. The moving front (interface) is defined by the roots of a continuous function  $\phi$ . For computational simplicity, the level set function is chosen as a signed distance function. For the task of image segmentation, numerous variants, involving image gradients (e.g. [4]) and region based forces

---

<sup>\*</sup> This work was supported by the Austrian Science Fund (FWF) under the grant P17066-N04.

(e.g. [5]) have been proposed to drive the front to the desired image boundaries. Although the level set method has been proven to be a robust numerical tool for a great number of applications, it also has some significant disadvantages [2]:

- The numerical solution of the level set equation is obtained by a forward Euler method with a global time step restriction and thus converges slowly.
- The interface of the level set function has to be tracked explicitly.
- For numerical stability, the level set function should periodically be reinitialized to a signed distance function.
- Multiple regions and open curves (curves having ends) cannot be handled in a straight-forward manner.

The *Variational* approach, which is closely related to the level set method, is based on defining an appropriate energy functional whose minimizer is an optimal solution to a given image. The celebrated Mumford-Shah segmentation model [6] was one of the first models for variational image segmentation. In its original setting, the Mumford-Shah functional is hard to minimize, due to the lacking convexity and regularity of the edge-length term. A solution was presented by Ambrosio and Tortorelli [7] via  $\Gamma$ -convergence, where the edge set is represented by means of a phase field  $z \in [0, 1]$ . Suppose,  $z = 1$  almost everywhere and sharply drops down to 0 in a  $\epsilon$ -neighborhood around the edges. Furthermore, consider the following energy:

$$L_{z,\epsilon} = \int_{\Omega} \epsilon |\nabla z|^2 \mathbf{d}\mathbf{x} + \int_{\Omega} \frac{(1-z)^2}{4\epsilon} \mathbf{d}\mathbf{x}, \quad (1)$$

where  $\epsilon$  is the phase field parameter which controls the size of the transition band. The remarkable result, associated with this energy is that  $L_{z,\epsilon}$  converges to the edge length, as  $\epsilon \rightarrow 0^+$ . It should also be noted, that this type of energy is not limited to closed curves, as it is the case for level set functions. Another possibility is to represent the edges as the transition of a phase field  $w$ , having two distinct phases [8], [9]. The Energy associated with this two-phase model is

$$L_{w,\epsilon} = \int_{\Omega} \epsilon |\nabla w|^2 \mathbf{d}\mathbf{x} + \int_{\Omega} \frac{(w(1-w))^2}{\epsilon} \mathbf{d}\mathbf{x} \quad (2)$$

The difference between (1) and (2) mainly consists of the type of function used in the second term. The former one uses a single well potential with a well at 1, the latter one uses a double well potential with equidepth wells at 0 and 1. Deriving the Euler Lagrange equation of (2), one obtains the so-called Allen-Cahn equation [10]

$$\frac{\partial L_{w,\epsilon}}{\partial w} \equiv \frac{\partial w}{\partial t} = -2\epsilon \Delta w + \frac{2}{\epsilon} w(1-w)(1-2w). \quad (3)$$

It has been shown that this equation approximates the motion of an interface by its mean curvature as  $\epsilon \rightarrow 0^+$ , which is important to get smooth boundaries [11]. The approximative formulation (3), offers some potential advantages:

- The explicit tracking of the moving interface (as it is the case for the level set method) is completely avoided.
- Since the phase fields energies (e.g. (2)) are quadratic, they can be solved by fast state-of-the-art solvers.
- Curves having edges or ends can easily be represented.
- The phase field parameter  $\epsilon$  can also be used for multiscale analysis (e.g. [12]).

In this paper we utilize the phase field approach to derive a very general segmentation model which can easily handle multiple regions. In contrast to the level set method, where the level set function defines clear cut boundaries between the regions, we interpret the phase fields as probabilities of pixels belonging to regions (inspired by [13]). This model is very general, since it allows pixel to belong to several regions. In Section 2 we derive the model formulation based on maximizing the mutual information between the phase fields and a probability function based on image features. Although we aim for being very general, we demonstrate that even using simple Gaussian functions as a prior for the image features is sufficient for a large number of segmentation problems while having the advantage of being very cheap in terms of computational costs. For optimization, we derive the Euler Lagrange equations of the model and show how they can be solved via alternating minimization. In Section 3 we present a fast and robust narrow band scheme to compute the model. In Section 4, the model is validated by different numerical experiments including natural and medical images. As a byproduct, we show that the pixel probabilities can also be used for border matting of objects having fuzzy boundaries. In the last Section we give some concluding remarks and present some ideas for future directions.

## 2 Segmentation Model

Consider a random variable  $\mathcal{P} \in \{1, \dots, K\}$  of region labels and a vector  $\mathbf{P}(\mathbf{x}) = (p_1(\mathbf{x}), \dots, p_K(\mathbf{x}))^T \in \mathbb{R}^K$ , describing the probability that a pixel  $\mathbf{x} \in \Omega \subseteq \mathbb{R}^n$  belongs to region  $\Omega_\ell$ , where  $\Omega = \cup_\ell \Omega_\ell$  and  $\cap_\ell \Omega_\ell = \emptyset$  for  $\ell \in [1, \dots, K]$ . This leads to the following partition of the image domain  $\Omega$ :

$$\Omega(\mathbf{x}) = \sum_{\ell=1}^K p_\ell(\mathbf{x}) \equiv 1, \quad p_\ell \geq 0 \quad (4)$$

where  $K$  is the number of image regions. Furthermore, let  $\mathcal{F} \subseteq \mathbb{R}^m$  be a random vector describing image features e.g. RGB channels of a color image. A segmentation is said to be optimal with respect to  $\mathcal{F}$ , if the information of  $\mathcal{F}$  which is covered by  $\mathcal{P}$  is maximal. This can be achieved by maximizing the mutual information between  $\mathcal{F}$  and  $\mathcal{P}$ .

$$I(\mathcal{F}; \mathcal{P}) = H(\mathcal{F}) - H(\mathcal{F} | \mathcal{P}), \quad (5)$$

where  $H(\cdot)$  and  $H(\cdot | \cdot)$  denote the entropy and the conditional entropy. This criterion was also used by the authors of [14] and [15] but it is important to

note, that our model is more general since it allows each pixel to belong to several regions with a certain probability.

Since  $H(\mathcal{F})$  is independent of the region labels, maximizing (5) can be turned into minimizing  $H(\mathcal{F} | \mathcal{P})$ . In order to obtain smooth boundaries, we additionally penalize the objective function by the phase field energy (2) which has, in a slightly different form, also been used in [13] for ‘‘Soft-Mumford-Shah’’ segmentation. Hence, the objective function we wish to minimize is

$$E = H(\mathcal{F} | \mathcal{P}) + \alpha L_{\mathbf{P}, \epsilon}, \quad (6)$$

where

$$L_{\mathbf{P}, \epsilon} = \sum_{\ell=1}^K \int_{\Omega} \epsilon |\nabla p_{\ell}|^2 + \frac{(p_{\ell}(1-p_{\ell}))^2}{\epsilon} \mathbf{d}\mathbf{x}, \quad (7)$$

with the constraints that

$$\sum_{\ell=1}^K p_{\ell} = 1 \quad \text{and} \quad p_{\ell} \geq 0 \quad (8)$$

The parameter  $\alpha$  is used to control the influence of the phase field energy term. By the weak law of large numbers, the entropy term  $H(\mathcal{F} | \mathcal{P})$  can be approximated by (up to negligible constants)

$$\tilde{H}(\mathcal{F} | \mathcal{P}) = - \sum_{\ell=1}^K \int_{\Omega} p_{\ell} \log(\text{Prob}(\mathcal{F} | \mathcal{P} = \ell)) \mathbf{d}\mathbf{x}. \quad (9)$$

## 2.1 Parametric Versus Non-parametric Density Estimation

Evaluating (9) requires the estimation of  $\text{Prob}(\mathcal{F} | \mathcal{P} = \ell)$ . This can either be done by assuming a certain probability density function or by a non-parametric density estimator. Although non-parametric density estimation would be preferable, we decided to use parametric models. The main reason is the reduced computational complexity which becomes significant for 3D segmentation problems. Moreover, we found that even using single Gaussians for each region is sufficient for a great number of segmentation tasks. However, we emphasize that our image model easily generalizes to more complex models such as *Gaussian Mixture Models* [16] or non-parametric density estimators [17].

For this paper, as mentioned above, we proceed by assuming that each component of  $\mathcal{F}$  follows a Gaussian probability density functions (PDFs), each associated with region  $\Omega_{\ell}$ .

$$G_{\ell}(\mathbf{s}, \boldsymbol{\mu}_{\ell}, \boldsymbol{\Sigma}_{\ell}) = \frac{1}{(2\pi)^{m/2} |\boldsymbol{\Sigma}_{\ell}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{s} - \boldsymbol{\mu}_{\ell})^T \boldsymbol{\Sigma}_{\ell}^{-1}(\mathbf{s} - \boldsymbol{\mu}_{\ell})\right), \quad (10)$$

where  $\boldsymbol{\mu}_{\ell}$ ,  $\boldsymbol{\Sigma}_{\ell}$  are the  $m \times 1$  mean vector and the  $m \times m$  covariance matrix. With this (9) becomes (up to negligible constants)

$$\tilde{H}(\mathcal{F} | \mathcal{P}) = \sum_{\ell=1}^K \int_{\Omega} p_{\ell} ((\mathbf{s}(\mathbf{x}) - \boldsymbol{\mu}_{\ell})^T \boldsymbol{\Sigma}_{\ell}^{-1}(\mathbf{s}(\mathbf{x}) - \boldsymbol{\mu}_{\ell}) + \log(|\boldsymbol{\Sigma}_{\ell}|)) \mathbf{d}\mathbf{x}, \quad (11)$$

where  $\mathbf{s}(\mathbf{x})$  is the feature vector at pixel position  $\mathbf{x}$ . By substituting (11) into (6) we have the complete energy functional of our segmentation model:

$$E = \tilde{H}(\mathcal{F} | \mathcal{P}) + \alpha L_{\mathbf{P}, \epsilon}, \quad (12)$$

## 2.2 Optimization of the Constraint Energy Functional

The minimization of (12) requires the estimation of the parameters  $\Theta = \{\boldsymbol{\mu}_\ell, \boldsymbol{\Sigma}_\ell\}$ ,  $\ell \in [1, \dots, K]$ , describing the Gaussian PDFs and the estimation of the optimal pixel ownership  $\mathbf{P}(\mathbf{x})$ . This multivariate optimization problem is solved via the alternating minimization algorithm (AM), which is a well established technique. Since the AM algorithm works iteratively, the estimation of the parameters and the pixel ownerships at the  $n$ -th iteration are denoted as  $\Theta^{(n)}$  and  $\mathbf{P}^{(n)}$ .

Given a pixel ownership  $\mathbf{P}^{(n)}(\mathbf{x})$ , the currently optimal parameters  $\Theta^{(n)}$  of the Gaussian PDFs are estimated by solving

$$\Theta^{(n)} = \operatorname{argmin}_{\Theta} \{E(\Theta | \mathbf{P}^{(n)}, \mathcal{F})\}. \quad (13)$$

Deriving the Euler Lagrange equations of (12) with respect to  $\Theta = \{\boldsymbol{\mu}_\ell, \boldsymbol{\Sigma}_\ell\}$ , one arrives at the following equations:

$$\boldsymbol{\mu}_\ell^{(n)} = \frac{1}{\int_{\Omega} p_\ell^{(n)}(\mathbf{x}) \, \mathbf{d}\mathbf{x}} \int_{\Omega} p_\ell^{(n)}(\mathbf{x}) \, \mathbf{s}(\mathbf{x}) \, \mathbf{d}\mathbf{x} \quad (14)$$

and

$$\boldsymbol{\Sigma}_\ell^{(n)} = \frac{1}{\int_{\Omega} p_\ell^{(n)}(\mathbf{x}) \, \mathbf{d}\mathbf{x}} \int_{\Omega} p_\ell^{(n)}(\mathbf{x}) (\mathbf{s}(\mathbf{x}) - \boldsymbol{\mu}_\ell^{(n)}) (\mathbf{s}(\mathbf{x}) - \boldsymbol{\mu}_\ell^{(n)})^T \, \mathbf{d}\mathbf{x} \quad (15)$$

Subsequently, based on the optimal estimation of  $\Theta$ , the optimal pixel ownership for the next iteration is obtained by

$$\mathbf{P}^{(n+1)} = \operatorname{argmin}_{\mathbf{P}} \{E(\mathbf{P} | \Theta^{(n)}, \mathcal{F})\}. \quad (16)$$

Since the different pixel ownerships  $p_\ell$  are related to each other by the constraint (8) they have to cooperate with each other. To satisfy the constraint  $\sum_{\ell=1}^K p_\ell = 1$ , we introduce a Lagrange multiplier  $\lambda$  and proceed by solving the following equations

$$\frac{\partial}{\partial p_\ell} \left\{ E + \lambda \left( \sum_{\ell=1}^K p_\ell - 1 \right) \right\} = 0. \quad (17)$$

As a result, we get  $\lambda = -\frac{1}{K} \sum_{\ell=1}^K \frac{\partial E}{\partial p_\ell}$ . By substituting  $\lambda$  into (17) we obtain the gradient of the constrained energy functional

$$\frac{\partial \bar{E}}{\partial p_\ell} = \frac{\partial E}{\partial p_\ell} - \frac{1}{K} \sum_{\ell=1}^K \frac{\partial E}{\partial p_\ell} = 0. \quad (18)$$

The expressions  $\frac{\partial E}{\partial p_\ell}$  are given by the gradients of the unconstrained energy functional which yield

$$\frac{\partial E}{\partial p_\ell} = h_\ell + \alpha \left( -2\epsilon \Delta p_\ell + \frac{2}{\epsilon} p_\ell (1 - p_\ell) (1 - 2p_\ell) \right), \quad (19)$$

where  $h_\ell = ((\mathbf{s}(\mathbf{x}) - \boldsymbol{\mu}_\ell)^T \boldsymbol{\Sigma}_\ell^{-1} (\mathbf{s}(\mathbf{x}) - \boldsymbol{\mu}_\ell) + \log(|\boldsymbol{\Sigma}_\ell|))$ . So far, we have not introduced the second constraint, namely  $p_\ell \geq 0$ . We found that it is easier to explicitly deal with this during the optimization procedure rather than introducing an additional Lagrange multiplier.

### 3 Fast and Robust Implementation

Before we start to implement our model, we briefly recall the notation used for image discretization, which will be used throughout the rest of the paper. We are using a 2D Cartesian grid which is defined as  $\{(x_i, y_j) \mid 1 \leq i \leq M, 1 \leq j \leq N\}$ . For convenience, we are using a uniform grid, for which all the subintervals  $\Delta x = [x_{i+1} - x_i]$  and  $\Delta y = [y_{j+1} - y_j]$  are equal in size. By definition, the use of a Cartesian grid implies a rectangular domain  $\Omega = [x_1, x_M] \times [y_1, y_N]$ .

#### 3.1 Discretization of the Euler Lagrange Equations

The implementation of (14) and (15) is straight-forward and does not require further explanations. For a solution of (18), we are using standard Gauss-Seidel iterations of the following approximated Newton-type descent scheme:

$$(p_\ell)_{i,j}^{(k+1)} = (p_\ell)_{i,j}^{(k)} - \frac{1}{(\tau_\ell)_{i,j}} \left( \frac{\partial \bar{E}}{\partial p_\ell} \right)_{i,j}^{(k)}. \quad (20)$$

Due to (18), we only have to discretize (19) which is given by

$$\left( \frac{\partial E}{\partial p_\ell} \right)_{i,j} = (h_\ell)_{i,j} + \alpha \left( -2\epsilon (\Delta p_\ell)_{i,j} + \frac{2}{\epsilon} ((p_\ell)_{i,j} - 3(p_\ell)_{i,j}^2 + 2(p_\ell)_{i,j}^3) \right), \quad (21)$$

where  $(\Delta p_\ell)_{i,j} = ((p_\ell)_{i+1,j} + (p_\ell)_{i-1,j} + (p_\ell)_{i,j+1} + (p_\ell)_{i,j-1} - 4(p_\ell)_{i,j})$  is a finite difference approximation of the Laplace operator. The parameters  $(\tau_\ell)_{i,j}$  are approximations of the second order derivatives of the energy functional and are given by

$$(\tau_\ell)_{i,j} = \frac{\partial}{\partial (p_\ell)_{i,j}} \left( \frac{\partial E}{\partial p_\ell} \right)_{i,j} = \alpha \left( 8\epsilon + \frac{2}{\epsilon} (1 - 6(p_\ell)_{i,j} + 6(p_\ell)_{i,j}^2) \right) \quad (22)$$

As mentioned in the previous section, we need to ensure  $p_\ell \geq 0$  during the Gauss-Seidel iterations. For this purpose, we use the following rule: First, we compute the values  $(p_\ell)_{i,j}^{(k+1)}$ , according to (20). Second, if the new values do not satisfy the constraint  $(p_\ell)_{i,j}^{(k+1)} \geq 0$ , we set  $(\tau_\ell)_{i,j} = (p_\ell)_{i,j} / \left( \frac{\partial \bar{E}}{\partial p_\ell} \right)_{i,j}^{(k)}$ .

Finally, we choose  $\tau_{i,j} = \min\{(\tau_\ell)_{i,j}\}$  and set  $(\tau_\ell)_{i,j} = \tau_{i,j}$  for all  $\ell \in [i, \dots, K]$ . Concerning the stability of the scheme, we can easily derive a lower bound of the phase field parameter  $\epsilon$ . Since  $p_\ell \in [0, 1]$ , we have  $\inf_{p_\ell}\{\tau_\ell\} = 0.5$ . If we require  $\tau_\ell > 0$ , we get  $\epsilon^2 > 1/8$ . We also found that just a few iterations (5-10) are sufficient in order to approximately solve (18) for one step of the AM algorithm.

### 3.2 Fast Narrow Band Implementation

The AM algorithm described above has a computational complexity of  $O(N^2)$ , where  $N$  is the size along one dimension of the image. If we restrict the computations to a narrow band, located around the transition of the phase fields, the computational complexity could be reduced to  $O(kN)$ , where  $k$  is the width of the narrow band. This concept has also been used in the context of level sets [18] and has shown to substantially reduce the computational costs, especially in the context of 3D applications. We propose the following algorithm to generate the narrow band of the phase field.

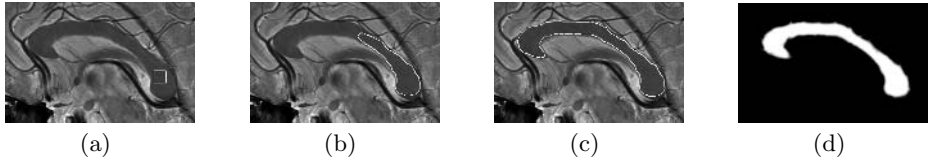
1. Choose all potential transition points:  $\mathcal{N}_0 = \{(x_i, y_j) \mid (\nabla p_\ell)_{i,j}^2 < \gamma\}$ , where  $\gamma$  is a small fixed threshold (e.g.  $\gamma = 0.05$ ).
2. The narrow band  $\mathcal{N}$  is generated by extending  $\mathcal{N}_0$  to a predefined width by solving the Eikonal equation [18] up to a fixed stopping value ( e.g.  $t_{max} = 5$ ).

The narrow band variants of (14) and (15) are straight-forward and thus omitted. To efficiently update the covariance we use the well known property of the covariance matrix,  $\Sigma = E[\mathbf{ss}^T] - \boldsymbol{\mu}\boldsymbol{\mu}^T$ . The narrow band variant of (18) is obtained by simply restricting (20) to the points of the narrow band. Hence, we have reduced the computational complexity to  $O(kN)$  for each iteration of the AM algorithm. After a few iterations of the AM algorithm, the narrow band has to be updated, which is done by the same procedure.

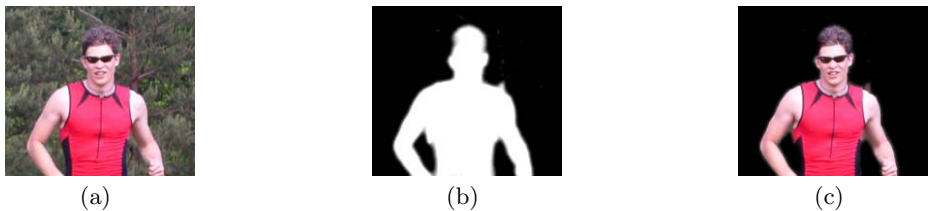
## 4 Experimental Results

In order to show the wide applicability of the proposed segmentation model, we have tested our algorithm on several images. For all experiments we used  $\epsilon = 2.0$ . Fig. 1 shows the two-phase segmentation of the corpus callosum from a brain MRI. As a single feature we used just the image intensity. The parameter  $\alpha$  was set to 15. This example shows, that using simple Gaussian PDFs is sufficient to distinguish the object from a complex background. Moreover, due to our narrow band formulation, only pixels contained in the transition between the phases are taken into account, which essentially increases the robustness of our algorithm. Fig. 2 shows the segmentation of a person from a color image. We used a three-dimensional feature vector composed of the RGB channels. The parameter  $\alpha$  was set to 20. This shows, that our model easily generalizes to higher dimensional feature vectors. Fig. 3 shows the segmentation of two textured image with 5 phases. The parameter  $\alpha$  was set to 20. For the first image, the feature vector was based only on pixel intensities. For the second, more challenging image, we

extended the feature vector by using the coefficients of the first level of a Haar-wavelet transform. Fig. 4 shows a three-phase segmentation of a natural image. The parameter  $\alpha$  was set to 20. The objects *koala*, *trunk* and *leaves* are fairly good extracted.



**Fig. 1.** Segmentation of the corpus callosum from a brain MRI. (a) Initial, (b) intermediate (30 iterations) and (c) final segmentation (100 iterations). (d) Phase field of the final segmentation.



**Fig. 2.** Segmentation of a person from a color image. (a) Original image, (b) final phase field (60 iterations) and (c) extracted person using pixel probabilities of the phase field.

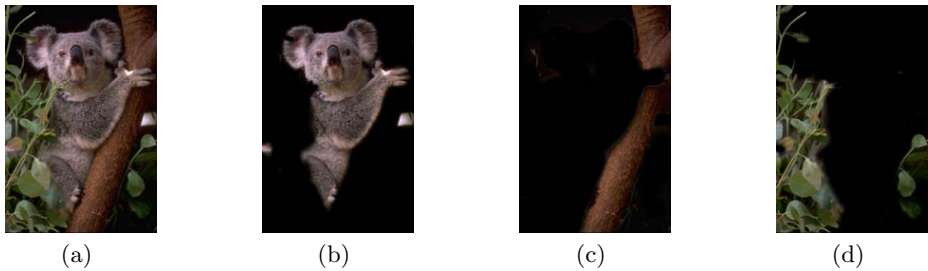


**Fig. 3.** Segmentation of textured images using 5 phases. The original images are shown with superimposed boundaries of the segmentation result.

#### 4.1 Automatic Border Matting

Fig. 5 shows the segmentation of a fox sitting in the grass. The parameter  $\alpha$  was set to 20. Since the fox apparently does not have clear-cut boundaries, one can not define a hard transition between fox and grass. As a byproduct of our method, we directly obtain matted boundaries of the object by using the pixel probabilities of the phase field. Moreover, the border matting of our





**Fig. 4.** Three-phase segmentation of natural image. (a) Original image. (b)-(d) Extracted objects: koala, trunk and leaves.



**Fig. 5.** Segmentation of an object having fuzzy boundaries. (a) Original image, (b) final phase field, (c) extracted object by using phase field probabilities and (d) detail view of (c).

model works adaptively, which means that the transition width of the phase field is automatically adapted to the appearance of the object boundary. This is in contrast to other methods such as [19], where border matting has to be performed in an extra step.

## 5 Conclusion

A probability based multi-phase model for variational image segmentation was presented in this paper. The model deals with phase fields, which are interpreted as the probabilities of pixels belonging to regions. The model formulation is derived by maximizing the mutual information between the phase fields and a PDF based on image features. We demonstrated, that this model is very general and can easily deal with multiple regions. For optimization, we derived its Euler-Lagrange equations and presented a fast and robust narrow band implementation of them. In order to show the wide applicability of the model, the algorithm was tested on several images with the same parameter settings. Additionally, we showed that the phase fields can easily be utilized to automatically obtain matted boundaries of fuzzy objects. As discussed in Section 2.1 we use simple multivariate Gaussians for density estimation, but it is clear that in order to handle more complex cases (illumination changes, texture) more sophisticated models are required. Additionally, when dealing with PDFs, the spatial

dependence of the features inside the regions is completely ignored. So far, the number of regions are user-specified, but it can be automated using multiscale patch statistics [13]. Future work will mainly concentrate on this issues.

## References

1. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. *Int'l J. of Comp. Vision* 1 (1988) 321–331
2. Osher, S., Fedkiw, R.: *Level Set Methods and Dynamic Implicit Surfaces*. Springer (2003)
3. Osher, S., Sethian, J.: Fronts propagating with curvature dependent speed: Algorithms based on hamilton-jacobi formulations. *J. Comput. Phys.* 79 (1988) 12–49
4. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. In: *Int'l. Conf. Computer Vision '95, Boston (1995)* 694–699
5. Chan, T., Vese, L.: Active contours without edges. *IEEE Trans. Image Processing* 10(2) (2001) 266–277
6. Mumford, D., Shah, J.: Optimal approximation by piecewise smooth functions and associated variational problems. *Comm. Pure Appl. Math.* 42 (1989) 577–685
7. Ambrosio, L., Tortorelli, V.: Approximation of functionals depending on jumps by elliptic functionals via  $\Gamma$ -convergence. *Comm. Pure. Appl. Math.* 43 (1990) 999–1036
8. Esedoglu, S., Tsai, Y.H.R.: Threshold dynamics for the piecewise constant Mumford-Shah functional. Technical report, CAM report (04-63) (2004)
9. Shen, J.:  $\Gamma$ -convergence approximation to piecewise constant Mumford-Shah segmentation. In: *Int'l Conf. Advanced Concepts Intell. Vision Systems.* (2005) 499–506
10. Allen, S., Cahn, J.: A microscopic theory for antiphase boundary motion and its application to antiphase domain coarsening. *Acta Metall. Mater.* (1979) 1085–1095
11. Rubinstein, J., Sternberg, P., Keller, J.B.: Fast reaction, slow diffusion, and curve shortening. *SIAM J. Appl. Math.* 49(1) (1989) 116–133
12. Droske, M., Rumpf, M.: Multi scale joint segmentation and registration of image morphology. *IEEE Transaction on Pattern Recognition and Machine Intelligence* (2005) submitted.
13. Shen, J.: A stochastic-variational model for soft Mumford-Shah segmentation. *Int'l J. Biomedical Imaging* (2006)
14. Awate, S.P., Tasdizen, T., Whitaker, R.T.: Unsupervised texture segmentation with nonparametric neighborhood statistics. In: *ECCV 2006, Graz (2006)* to appear.
15. Kim, J., Fisher, J.W., Yezzi, A., Cetin, M., Willsky, A.S.: Nonparametric methods for image segmentation using information theory and curve evolution. In: *IEEE Int'l Conf. on Image Processing, Rochester (2002)*
16. Redner, R.A., Walker, H.F.: Mixture densities, maximum likelihood and the EM algorithm. *SIAM Review* (1984)
17. Parzen, E.: On estimation of a probability density function and mode. *Annals of Mathematical Statistics* 33(3) (1962) 1065–1076
18. Sethian, J.: *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry*. 2<sup>nd</sup> edn. Cambridge University Press (1999)
19. Rother, C., Kolmogorov, V., Blake, A.: Grabcut - interactive foreground extraction using iterated graph cuts. In: *ACM Siggraph.* (2004)

# Provably Correct Edgel Linking and Subpixel Boundary Reconstruction

Ullrich Köthe, Peer Stelldinger, and Hans Meine

University of Hamburg, 22527 Hamburg, Germany

**Abstract.** Existing methods for segmentation by edgel linking are based on heuristics and give no guarantee for a topologically correct result. In this paper, we propose an edgel linking algorithm based on a new sampling theorem for shape digitization, which guarantees a topologically correct reconstruction of regions and boundaries if the edgels approximate true object edges with a known maximal error. Experiments on real and generated images demonstrate the good performance of the new method and confirm the predictions of our theory.

## 1 Introduction

The question, whether or when a computed image segmentation corresponds closely to the underlying real-world partitioning, is fundamental to image understanding. A number of partial results have been obtained in the past, but they are not sufficiently realistic to model many actual imaging situations, since they do not allow measurement errors.

The analysis we are going to present is based on a clear distinction between the ideal geometric image, which cannot be observed in practice, and the actually available digital image. The geometric image has infinite resolution (i.e. is an analog function) and can be thought of as the projection of a 3-dimensional scene, although we do not consider the details of the projection in this work. Instead, we think of the analog image as a given geometric partitioning of the plane into distinct regions. The interior of each region is described by some simple function (e.g. a constant), but the transitions between regions are discontinuous. This ideal analog image is then transformed into a digital image by a real camera. Beyond geometric projection, a real camera is characterized by its point spread function, the sampling grid and its quantization and noise models. The partition of the geometric image must be inferred from the limited information in the digital image. We ask how accurate this reconstruction can be.

Recently we developed a geometric sampling theorem which assumes that sampling points (edgels) are placed roughly along the contour of the regions to be segmented. The edgels can be obtained by an arbitrary edge detector, as long as the accuracy of the detected edges is known. In this paper, we compare common edge detectors in the context of our theory and show how to use them for generating a topologically correct image segmentation.

## 2 The Boundary Reconstruction Algorithm

We consider the task of reconstructing the boundary of a partition of the Euclidean plane from a sampled representation. The plane partition  $\mathcal{P}$  to be recovered is defined by a finite set of *points*  $P = \{p_i \in \mathbb{R}^2\}$  and a set of pairwise disjoint *arcs* connecting these points. The union of the points and arcs is the *boundary* of the partition  $B = P \cup A$ , and the *regions*  $R = \{r_i\}$  are the connected components (maximal connected sets) of the complement of  $B$ .

Previous proofs about topologically correct reconstruction were restricted to *binary* partitions. That is, one can assign two labels (foreground and background) to the regions such that every arc is in the closure of exactly one foreground and one background region. Examples are *r-regular partitions* in [8, 6, 5, 9] and *r-halfregular partitions* in [10]. Both are too restrictive for practical use (see [11] for details). In this paper we use a more general class of feasible plane partitions:

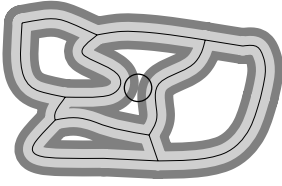
**Definition 1.** *A plane partition  $\mathcal{P}$  is called  $r$ -stable when its boundary  $B$  can be dilated with a closed disc of radius  $s$  without changing its homotopy type for any  $s \leq r$ . We say two points  $x_1, x_2 \in B$  delimit a  $(\theta, d)$ -spike, if the distance from  $x_1$  to  $x_2$  is at most  $d$  and if every path on  $B$  from  $x_1$  to  $x_2$  contains at least one point with  $\angle x_1 x_2 < \theta$ . We say that  $\mathcal{P}$  has no  $(\theta, d)$ -spikes if no pair of boundary points  $x_1, x_2 \in B$  delimits a  $(\theta, d)$ -spike.*

Thus a plane partition is  $r$ -stable if we can replace an infinitely thin boundary with a strip of width  $2r$  such that the number and enclosure hierarchy of the resulting regions is preserved. In particular, “waists” are forbidden, whereas junctions are allowed, see Fig. 1. Obviously, an  $r$ -stable plane partition has no  $(\pi, 2r)$ -spikes. Intuitively, two points delimit a  $(\theta, d)$ -spike, if the shortest boundary path between them does not differ too much from a straight line – it lies inside the shaded region in Fig. 2. In order to digitize such plane partitions, we approximate the *boundary* of the partition with a finite set of *adaptively placed* sampling points. The sampling points should be “near” the boundary:

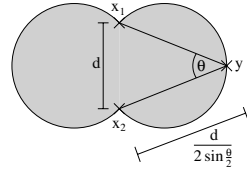
**Definition 2.** *A finite set of sampling points  $S = \{s_i \in \mathbb{R}^2\}$  is called a  $(p, q)$ -sampling of the boundary  $B$  when the distance of every boundary point  $b \in B$  to the nearest point in  $S$  is at most  $p$ , and the distance of every sampling point  $s \in S$  to the nearest point in  $B$  is at most  $q$ . The points in  $S$  are called edgels.*

The Hausdorff distance  $d_H(B, S)$  between the boundary and the sampling points is  $\max(p, q)$ . The exact values of  $p$  and  $q$  depend on where the edgels come from. This is discussed in detail in section 4. Our new edgel linking algorithm is essentially a hysteresis thresholding on the sizes of Delaunay triangles:

1. Compute the Delaunay triangulation  $D$  of the edgels  $S$ .
2. Mark all triangles in  $D$  (including their edges) with a circumradius  $< \alpha$ .
3. Additionally mark Delaunay edges whose circumcircle contains no edgel and has a radius smaller than  $\alpha$ .
4. Find connected components of unmarked triangles and edges.



**Fig. 1.** An  $r$ -stable plane partition does not change the homotopy type when dilated with a disc of radius of at most  $r$  (light gray), while dilations with bigger radius (dark gray) may connect different arcs as marked by the circle (see Def. 1)



**Fig. 2.** Any point which encloses an angle of at least  $\theta$  with  $x_1$  and  $x_2$  must lie inside the shaded region. The shown  $y$  is the one with the maximal distance to the nearer one of  $x_1$  and  $x_2$ . Thus there is a path from  $x_1$  to  $x_2$  inside the shaded region and each of its points has a distance of at most  $\frac{d}{2 \sin \frac{\theta}{2}}$ .

5. For each component from step 4 which does *not* contain any triangle with a circumradius of at least  $\beta$ , mark all its triangles and edges.

The union of marked triangles and edges is a simplicial complex which we denote  $(\alpha, \beta)$ -boundary reconstruction from the edgels. The components of its complement are called  $(\alpha, \beta)$ -holes. Under certain conditions, these holes exactly correspond to the regions of the original  $r$ -stable plane partition, as proven in [11]:

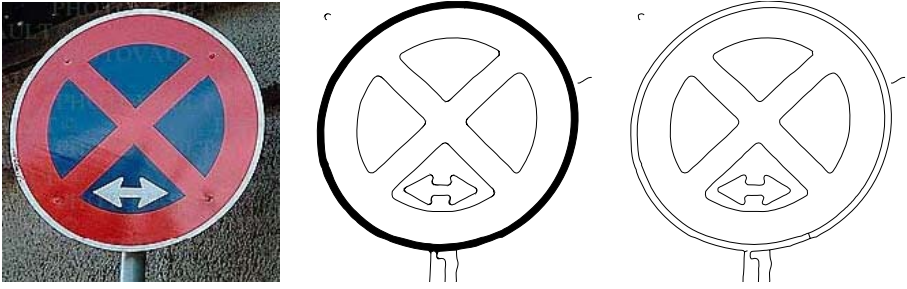
**Theorem 1 (boundary sampling theorem).** *Let  $\mathcal{P}$  be an  $r$ -stable plane partition, and  $S$  a  $(p, q)$ -sampling of  $\mathcal{P}$ 's boundary  $B$ . Then the  $(\alpha, \beta)$ -boundary reconstruction  $\mathcal{R}$  defined by  $S$  is homotopy equivalent to  $B$ , and the  $(\alpha, \beta)$ -holes of  $\mathcal{R}$  are topologically equivalent to the regions  $r_i$  of  $\mathcal{P}$ , provided the following conditions are met:*

1.  $p < \alpha \leq r - q$
2.  $\beta = \alpha + p + q$
3. every region  $r_i$  contains an open  $\gamma$ -disc with  $\gamma \geq \beta + q > 2(p + q)$ .

### 3 Boundary Thinning and Neighborhood Relations

Since the  $(\alpha, \beta)$ -boundary reconstruction may contain triangles, it is not in general thin (i.e. locally 1-dimensional). However, many algorithms that build upon segmentation results cannot handle partially thick boundary representations. Therefore we propose a topology preserving boundary thinning. We call an edge in the  $(\alpha, \beta)$ -boundary reconstruction *simple* if its removal does not change the topology of the reconstructed regions. Simple edges can be easily recognized: they bound an  $(\alpha, \beta)$ -hole on one side and a triangle in the boundary reconstruction on the other. Thinning removes all simple edges iteratively:

1. Find all simple edges of the given  $(\alpha, \beta)$ -boundary reconstruction and put them in a priority queue (the sorting is discussed below).



**Fig. 3.** *left:* original; *center:*  $(\alpha, \beta)$ -boundary reconstruction; *right:* minimal reconstruction after thinning. (Edgels from Canny's algorithm on a color gradient)

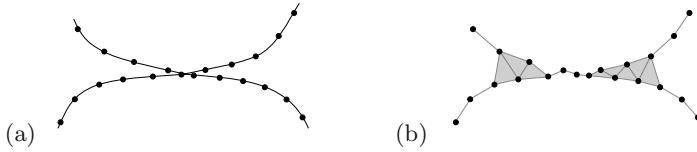
2. As long as the queue is not empty, fetch an edge from the queue and remove it from the boundary reconstruction if it is still simple (it may have lost this property after removal of other edges). Put the edges in the triangle of the removed edge in the queue if they have now become simple.

As far as region topology is concerned, the ordering of the edgels in the priority queue is arbitrary. For example, we can measure the contrast (image gradient) along each edge and remove weak edges first. A particularly interesting ordering is defined by the length of the edges:

**Definition 3.** *A (not necessarily unique) minimal boundary reconstruction is obtained from an  $(\alpha, \beta)$ -boundary reconstruction by means of topology-preserving thinning where the longest edges are removed first.*

The resulting boundaries are illustrated in Fig. 3. Since region topology is preserved, the minimal boundary reconstruction is homotopy equivalent to the boundary  $B$  of the original plane partition  $\partial P$ . The two boundaries are not in general topologically equivalent, because the adjacency relations between regions may differ (see below for details), and the reconstruction may contain short edges, which end in the interior of a region (they can also be removed iteratively).

Since the minimal boundary reconstruction is the shortest possible one with correct topology, the surviving edges connect edgels closest to each other. Neighboring edgels therefore align in an optimal way on the thinned boundary. The length  $d_{\max}$  of the longest surviving edge is a measure of the density of the boundary sampling. The maximum distance  $p$  between a true boundary point and the nearest edgel may be much larger than  $d_{\max}/2$  if the displacement of neighboring edgels is highly correlated as is usually the case in practice. For example, edgels along a circular arc are consistently biased toward the concave side of the curve. When we set  $\alpha' = d_{\max}/2 + \epsilon < p$  (with arbitrarily small  $\epsilon$ ), an  $(\alpha', \beta)$  reconstruction of the edgel set is still correct in the sense of theorem 1: since the minimal reconstruction is a subset of the  $(\alpha', \beta)$  reconstruction, no true regions can get merged. Since  $\alpha' < \alpha$ , no region can get lost, and since  $\beta$  remained unchanged, no additional holes can be created. In fact,  $\beta' = \alpha' + p + q < 2p + q$  would have been sufficient.



**Fig. 4.** Narrow spikes can lead to a boundary reconstruction where originally unconnected regions (a) look like they had a common boundary edge (b)

Theorem 1 does not guarantee that the neighborhood relations between reconstructed regions are the same as of the original regions, as can be seen in Fig. 4. The following theorem shows that neighborhood relations are preserved when the boundary arcs are long enough and free of  $(\theta, d)$ -spikes:

**Theorem 2.** *Let  $\mathcal{P}$  be an  $r$ -stable plane partition with regions  $r_i$  and boundary  $B$  having no  $(\theta, d)$ -spikes. Further, let  $S$  be a  $(p, q)$ -sampling of  $B$  and  $\mathcal{R}$  the  $(\alpha, \beta)$ -boundary reconstruction of  $S$  with regions  $h_i$ , such that all requirements of theorem 1 are fulfilled.  $S_i = \partial h_i \cap S$  denotes the set of edgels on the boundary of  $h_i$ . When  $d \geq 2(\alpha + q)$  and  $p' := d / (2 \sin \frac{\theta}{2}) + q$  the following holds:*

1. *If the distance between the two nearest edgels of  $S_i$  and  $S_j$  exceeds  $2p'$ , the corresponding original regions  $r_i, r_j$  are not adjacent, i.e.  $\partial r_i \cap \partial r_j = \emptyset$ .*
2. *When there exists a point  $x$  with  $d_H(x, S_i) \leq p'$ ,  $d_H(x, S_j) \leq p'$  and  $d_H(x, S_k) > 2p'$  for all  $k \neq i, j$ , the original regions  $r_i, r_j$  are arc-adjacent.*
3. *If two regions  $r_i, r_j$  have a distance greater than  $2(p' + q)$ , the conditions of item 1 are always fulfilled.*
4. *If two regions  $r_i, r_j$  have a common boundary point  $x$  such that  $d_H(x, S_k) > 3p'$  for all  $k \neq i, j$ , the conditions of item 2 are always fulfilled, i.e. adjacency of  $r_i$  and  $r_j$  can be detected in the boundary reconstruction.*

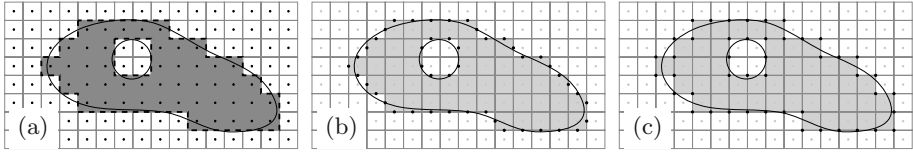
*Proof.* (1) For any  $s_t \in S_i$  let  $x_t \in \partial r_i$  be the nearest boundary point. Then for any two  $s_{t_1}, s_{t_2}$  being connected by a line segment of  $\partial h_i$ , the distance between  $x_{t_1}$  and  $x_{t_2}$  is smaller than  $2(\alpha + q)$ . Since  $(\theta, d)$ -spikes do not exist, the distance of each point of  $\partial r_i$  to the nearest  $x_t$  cannot exceed  $d / (2 \sin \frac{\theta}{2})$  and thus the distance of  $\partial r_i$  to  $\partial h_i$  is bounded by  $p'$ . The same holds for  $h_j$ . When the shortest distance between  $S_i$  and  $S_j$  is larger than  $2p'$ ,  $\partial r_i$  and  $\partial r_j$  cannot intersect.

(2) Both  $S_i$  and  $S_j$  intersect the disc  $\mathcal{B}_{p'}^0(x)$ . Since  $d_H(x, S_k) > 2p'$  for every  $k \neq i, j$ , no part of  $\partial r_k$  can intersect  $\mathcal{B}_{p'}^0(x)$ . Thus  $r_i$  and  $r_j$  are the only regions which intersect  $\mathcal{B}_{p'}^0(x)$ , which is only possible when they have a common edge.

(3) Since the distance between  $r_i$  and  $r_j$  exceeds  $2(p' + q)$ ,  $S_i, S_j$  have to be more than  $2p'$  away from each other.

(4) Due to the absence of  $(\theta, d)$ -spikes, the distance  $d_H(x, S_k), k \neq i, j$  must be greater than  $2p'$ . For the same reasons,  $d_H(x, S_i) \leq p'$  and  $d_H(x, S_j) \leq p'$ .  $\square$

It follows that if every junction of  $\mathcal{P}$  has degree 3, the boundary sampling only needs to be sufficiently accurate (i.e.  $p, q$ , and  $\alpha$  are sufficiently small) in order to reconstruct not only the topology of every region of a plane partition, but also the complete neighborhood relations, i.e. a complete combinatorial map [2] encoding  $\mathcal{P}$ 's abstract topology, without any error.



**Fig. 5.** The *interpixel boundary* (dashed) can be extracted from the subset digitization (a). It includes both the *midcrack digitization* (b) and the *endcrack digitization* (c).

## 4 Application to Popular Segmentation Schemes

In order to apply our boundary reconstruction algorithm, we can derive correct choices for  $\alpha$  and  $\beta$  from the error bounds  $p$  and  $q$  of the edgel detector. First, let us pretend that we have access to the exact projected image, i.e. to the plane partition  $\mathcal{P}$ . One possibility to digitize this partition is the so-called *subset digitization*: We assign the same label to two pixels iff their centers are in the same region. Then, interpixel edges (crack edges) can be defined between pixel facets with different labels, see Fig. 5a. Crack edges give rise to two natural kinds of edgels: *endcrack* and *midcrack* edgels (located on the end or center points of the cracks respectively, Fig. 5b and c). When the boundaries of the plane partition are free of  $(\theta, d)$ -spikes, the following bounds can be derived [11]:  $q = \frac{h}{\sqrt{2}}$  (endcrack) and  $q = \frac{h}{2}$  (midcrack) and  $p = q + (\frac{h}{2} + q) / \sin \frac{\theta}{2}$  (both cases), where  $h \leq \frac{d}{1+\sqrt{2}}$  is the required pixel distance. For example, when  $h = 1$  and the plane partition has no  $(60^\circ, d)$ -spikes with  $d > 2.4$ , we get  $p \approx 1.31$ ,  $q \approx 0.7$  for endcrack and  $p = 1$ ,  $q = 0.5$  for midcrack digitization, i.e. the latter is more accurate.

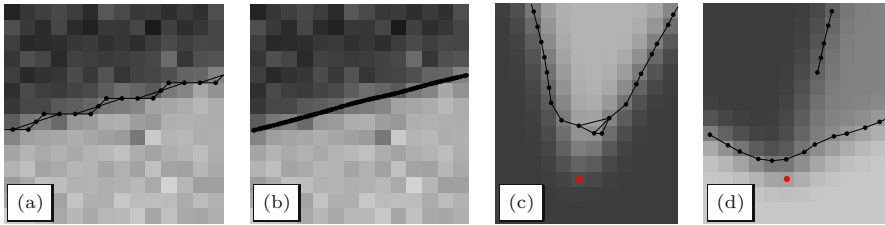
Many segmentation algorithms (e.g. zero-crossing-based edge detectors and the watershed algorithm) compute image labelings similar to subset digitization, which can be used to define endcrack and midcrack edgels. However, their error bounds differ from the ideal ones obtained above. To quantify these differences, we model the transformation from analog to digital images in real cameras:

$$f_{ij} = (\text{PSF} \star f(x, y))_{ij} + n_{ij} \quad (1)$$

where  $f(x, y)$  is the ideal geometric image, PSF is the point spread function, subscripts denote sampling, and  $n_{ij}$  is additive Gaussian noise (quantization is neglected). The PSF (which shall be band-limited) suppresses high spatial frequencies and the resulting smooth transitions between regions allow for sub-pixel accurate edge localization. On the other hand, systematic localization errors are introduced because blurring distorts edges. Noise causes additional statistical errors in  $p$  and  $q$ . We estimate these errors for a number of exemplary edge detectors: we consider two variants of the Haralick detector as representatives of zero-crossing-based algorithms, and three variants of Canny's algorithm to exemplify ridge-based edge detection. Haralick [4] defines edgels at the zero-crossing of the second derivative along the gradient direction:

$$b = f_x^2 f_{xx} + 2f_x f_y f_{xy} + f_y^2 f_{yy} \stackrel{!}{=} 0 \quad (2)$$





**Fig. 6.** Edgels and boundary reconstruction using  $\alpha = 1.55, \beta = 2$ : (a) midcrack variant and (b) subpixel variant of Haralick's algorithm. Note the lower density and higher displacement of the former. (c) Parabola and (d) spline variant of Canny's algorithm. Red dots indicate the ground-truth corner locations.

provided that the third derivative along the same direction is negative (indicating a local gradient maximum), and the gradient magnitude is above a threshold. Crack edges between positive and negative pixels of  $b$  where the constraints are fulfilled define a set of midcrack edgels. Their fixed accuracy can be improved when a continuous function  $\tilde{b}$  is computed by spline interpolation of  $b$ , and edgels are located in  $\tilde{b}$  by means of Newton iteration along the gradient direction. In our implementation of this variant, edgels are placed roughly at a distance of 0.1 pixels along the edge, Fig. 6a, b.

In contrast, Canny's algorithm [3] uses the gradient magnitude  $\sqrt{f_x^2 + f_y^2}$  and looks for relative maxima along the gradient direction. Better localization (significantly smaller  $q$ ) is achieved by either computing the maximum of an approximating parabola across the edge, or by Newton iterations on a continuously interpolated version of the gradient image, Fig. 6c and d. We estimate  $p$  and  $q$  on a large number of images created by numerical solution of the convolution integral (1) at various angles and grid positions, Fig. 6. Derivatives are computed by Gaussian filters at scale  $\sigma_E$ , and the PSF is also Gaussian with scale  $\sigma_{PSF}$ . To avoid aliasing we use  $\sigma_E \geq 1$  and  $\sigma_{PSF} = 1$  (cf. [12]).

First, consider straight edges. A radial symmetric PSF does not distort straight edges and  $q$  should be close to zero (non-zero values reflect discrepancies between the computational theory and its actual realization). Subpixel methods achieve  $q \lesssim 0.05$  pixels. With the exception of the subpixel Haralick operator (which places edgels very densely),  $p$  roughly equals the pixel radius. Row 1 in Table 1 lists the maximum errors we found.

The effect of image noise on straight edge localization was analysed by Canny [3]. When the noise is Gaussian distributed with zero mean and standard deviation  $s_N$ , the expected error (in pixels) is

$$E[\xi] = \frac{s_N \sqrt{6}}{a} \frac{1}{4} \left( 1 + \frac{\sigma_{PSF}^2}{\sigma_E^2} \right)^{3/2} \quad (3)$$

where  $a$  is the height of the step, and  $a/s_N$  is the signal-to-noise ratio (SNR). When  $\sigma_{PSF} \approx \sigma_E$ , we get  $E[\xi] \approx 1.7 \frac{s_N}{a}$ . For  $\sigma_E \rightarrow \infty$ , the error approaches  $0.6 \frac{s_N}{a}$  (the common belief that the error increases with  $\sigma_E$  is only justified in

**Table 1.** Experimental estimates of the maximum errors  $p$  and  $q$  (pixels). Theoretical predictions are given in brackets. Unless noted, there was no noise and  $\sigma_{\text{PSF}} = \sigma_E = 1$ .

	Canny (pixel coordinates)		Canny (parabola)		Canny (spline)		Haralick (midcrack)		Haralick (spline)	
	$p$	$q$	$p$	$q$	$p$	$q$	$p$	$q$	$p$	$q$
straight line	0.79	0.70	0.71	0.05	0.75	0.02	0.70	0.47	0.19	0.46
		[0.7]		[0.0]		[0.0]		[0.5]		[0.0]
straight line SNR = 10	1.0	0.82	0.81	0.47	0.92	0.57	0.90	0.93	0.63	0.85
				[0.52]		[0.52]				[0.52]
straight line $\sigma_E = 2$ , SNR = 10	1.0	0.81	1.0	0.28	1.0	0.28	0.79	0.73	0.57	0.81
				[0.26]		[0.26]				[0.26]
disc, radius = 4		0.73		0.73		0.25		0.74		0.29
						[0.2]				[0.2]
corner, 90°	1.58	0.84	1.38	0.76	1.34	0.69	1.52	0.93	1.15	0.71
	[0.71]		[0.71]		[0.71]		[0.71]		[0.71]	
corner, 15°	4.03	1.3	3.99	0.92	3.96	0.94	3.39	1.33	3.96	1.3
	[3.1]		[3.1]		[3.1]		[3.1]		[3.1]	
junction, degree=3	2.70	1.56	2.66	1.15	2.70	1.40	2.25	1.81	2.20	1.71

1D). In typical images  $\frac{a}{s_N}$  is between 5 and 100. The expected statistical error is then below 0.2 pixels, and the maximum error does not exceed  $3E[\xi] = 0.6$  pixels with probability 0.997. Rows 2 and 3 of Table 1 confirm these predictions.

Smoothing of curved boundaries with the PSF results in biased edgel positions. The gradient magnitude of a disc with radius  $\rho$  and contrast  $a$  is [1]

$$g(r) = |a| \frac{\rho}{\sigma^2} e^{-\frac{r^2 + \rho^2}{2\sigma^2}} I_1 \left( \frac{r\rho}{\sigma^2} \right) \quad (4)$$

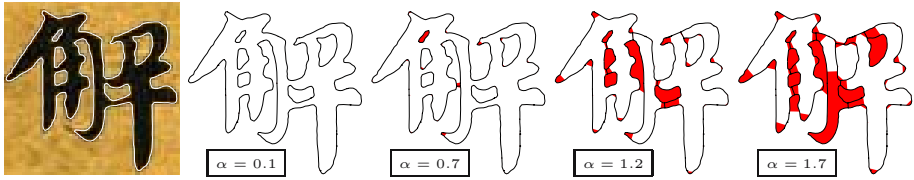
where  $r$  is the distance from the center of the disc,  $I_1$  is the modified Bessel function of order 1, and  $\sigma^2 = \sigma_{\text{PSF}}^2 + \sigma_e^2$  is the combined scale of the PSF and edge operator. The bias depends on the curvature radius  $\rho$  and the scale  $\sigma$ . It is directed towards the concave side of the curve when  $\sigma < 0.8\rho$  (which is true in most practical situations). Row 4 of Table 1 compares theoretical predictions and experimental estimates for  $\rho = 4$ . It can be seen that the best methods (using spline interpolation and Newton iterations) are very close to the theoretical limit.

A bias toward the concave side of the contour is also observed at corners. Its magnitude depends on  $\sigma$  and the corner angle  $\varphi$  and is maximal along the bisector of the corner. The gradient maximum along the bisector (i.e. the estimated edge location) is the solution of the implicit equation [7]

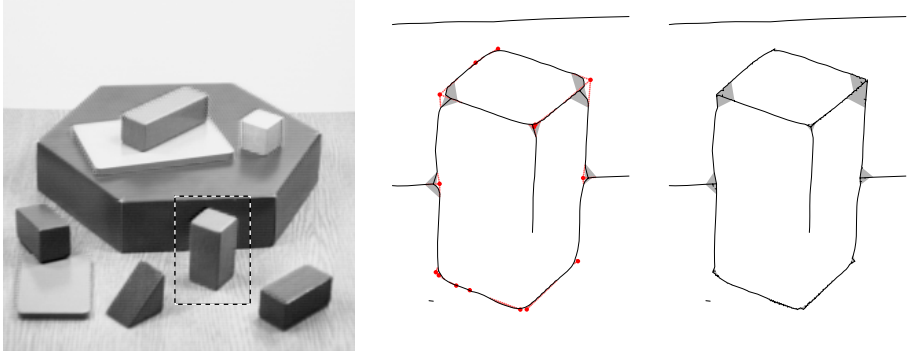
$$\frac{1}{2\pi\sigma^2} e^{-\frac{r^2}{2\sigma^2}} - \left( \tan \left( \frac{\varphi}{2} \right) \right)^2 \frac{r}{2} \left( 1 + \operatorname{erf} \left( \frac{r}{\sqrt{2}\sigma} \right) \right) = 0 \quad (5)$$

where erf is the error function. The sharper the corner, the higher the bias. E.g. for  $\varphi = 90^\circ, 45^\circ, 15^\circ$  it is approximately  $0.5\sigma, 1.2\sigma$ , and  $2.2\sigma$ . Rows 5 and 6 in Table 1 show that actual errors are even higher than theory predicts.

The situation at junctions is even more complicated. The large number of degrees of freedom (angles, intensities) does not allow the error to be described in a compact way. The algorithms considered here are usually unable to close all contours near a junction. The remaining gaps also cause  $p$  to attain quite large values, as row 7 of Table 1 shows.



**Fig. 7.** Chinese character (*white*: contours extracted by levelcontour tracing [13]),  $(\alpha, \beta)$ -boundary reconstructions with increasing values of  $\alpha$  (*red*: before thinning, *black*: minimal boundary reconstruction).



**Fig. 8.** Left: original image and ROI; center:  $(\alpha, \beta)$ -boundary reconstruction from subpixel Canny edgels (black and gray), thinned reconstruction (black only) and additional edgels to be added (red); right: modified reconstruction including new edgels

Fig. 3 and Fig. 7 show results of  $\alpha, \beta$ -reconstruction in two real images. Region topology is correctly recovered when  $\alpha$  and  $\beta$  are properly chosen. Since edgels are considered as isolated points, our new algorithm also facilitates the combination of edgels from different sources, cf. Fig. 8: The edgels computed by Canny's algorithm are not very accurate near corners and junctions, and this requires large  $\alpha$  and  $\beta$  causing the reconstruction to be thick in problematic areas (gray). In a second step, a maximum likelihood junction position is computed from the gradient magnitudes and directions at the edgels in a neighborhood of each thick area, resulting in the red points. These points are simply added to the set of edgels, and the reconstruction from the new set is much more accurate than the original one.

## 5 Conclusions

To our knowledge, this paper exploits the first geometric sampling theorem which explicitly considers measurement errors. We carefully derive the theoretical properties of several well-known edge detectors in order to apply our new theorem and demonstrate theoretically correct edgel linking. The resulting segmentations are similar to what one gets from traditional heuristic linking, but their prop-

erties can now be formally proven thanks to their theoretical basis in Delaunay triangulation. The key to these advancements has been the shift of attention from region-based digitization models to edge based ones: the assumption that no sampling points are in the interior of any region (beyond the known error bound) allows us to reliably recover region and boundary connectivity.

We demonstrated that many known digitization and segmentation methods can be analyzed and applied in the new framework by simply determining their error bounds. We can predict whether a given image will be handled properly by an algorithm with a certain error bound. When the error increases, the performance degrades gracefully: first, the recovered boundary becomes thick when the detailed curve shape or junction connectivity can no longer be unambiguously determined. Then, regions get split at too narrow waists, and finally too small regions will be lost (cf. Fig. 7). When additional edgels are added within the thick part of the  $(\alpha, \beta)$ -boundary reconstruction, the accuracy parameters  $p$  and  $q$  will never increase. This opens up new possibilities for algorithm combination. For example, one could start with an edge detector and a large  $\alpha$  which produces thick boundaries near corners and junctions. Additional edgels can then be computed by a corner detector whose output is confined to these areas, so that it cannot produce false positives within regions. In fact, false positives (large  $q$ ) and false negatives (large  $p$ ) are the major difficulties in our new algorithm. We are currently investigating how these can be recognized and removed.

## References

1. H. Bouma, A. Vilanova, L.J. van Vliet, and F.A. Gerritsen: *Correction for the dislocation of curved surfaces caused by the PSF in 2D and 3D CT images*, IEEE Trans. Pattern Analysis and Machine Intelligence, 27(9):1501-1507, 2005
2. J.-P. Braquelaire, L. Brun: *Image segmentation with topological maps and interpixel representation*, J. Visual Communication and Image Repr. 9(1):62-79, 1998
3. J. Canny: *A Computational Approach to Edge Detection*, IEEE Trans. Pattern Analysis and Machine Intelligence, 8(6):679-698, 1986
4. R. Haralick, L. Shapiro: *Computer and Robot Vision*, vol. 1, Addison Wesley, 1992
5. L.J. Latecki, C. Conrad, A. Gross: *Preserving Topology by a Digitization Process*, J. Mathematical Imaging and Vision, 8:131-159, 1998
6. T. Pavlidis: *Algorithms for Graphics and Image Processing*, Comp. Sc. Press, 1982
7. K. Rohr: *Localization Properties of Direct Corner Detectors*, J. Mathematical Imaging and Vision 4:139-150, 1994
8. J. Serra: *Image Analysis and Mathematical Morphology*, Academic Press, 1982
9. P. Stelldinger, U. Köthe: *Towards a General Sampling Theory for Shape Preservation*, Image and Vision Computing, 23(2):237-248, 2005
10. P. Stelldinger: *Digitization of Non-regular Shapes*, in: C. Ronse, L. Najman, E. Decenciere (Eds.): *Mathematical Morphology*, ISMM, 2005
11. P. Stelldinger, U. Köthe, H. Meine: *Topologically Correct Image Segmentation using Alpha Shapes*, University Hamburg, Technical Report, 2006
12. B. ter Haar Romeny: *Front-End Vision and Multi-Scale Image Analysis*, Kluwer Academic Publishers, Dordrecht, 2003
13. Allgower, E.L., Georg, K.: Numerical path following. In: P.G. Ciarlet, J.L. Lions (Eds.), *Handbook of Numerical Analysis* 5 (1997) 3-207. North-Holland.

# The Edge Preserving Wiener Filter for Scalar and Tensor Valued Images

Kai Krajssek and Rudolf Mester

J.W. Goethe University, Frankfurt, Germany  
Visual Sensorics and Information Processing Lab  
{krajsek, mester}@vsi.cs.uni-frankfurt.de  
<http://www.vsi.cs.uni-frankfurt.de/>

**Abstract.** This contribution presents a variation of the Wiener filter criterion, i.e. minimizing the mean squared error, by combining it with the main principle of normalized convolution, i.e. the introduction of prior information in the filter process via the certainty map. Thus, we are able to optimize a filter according to the signal and noise characteristics while preserving edges in images. In spite of its low computational costs the proposed filter schemes outperforms state of the art filter methods working also in the spatial domain. Furthermore, the Wiener filter paradigm is extended from scalar valued data to tensor valued data.

## 1 Introduction

De-noising of images is still a challenging problem in image processing. Amongst linear image de-noising methods, Wiener filtering[1] is known to be the optimal estimator for the true underlying image. However, Wiener filtering as a linear and shift invariant filter scheme is often assumed to be unsuitable for images containing edges. In order to cope with edges in images, nonlinear image de-noising methods have been extensively examined. We list here diffusion based methods [2], variational methods like total variation [3], mean shift filtering [4], channel smoothing [5], wavelet based methods [6] or combinations of different methods [7,8]. In the context of mixed methods, Wiener filtering has again obtained attention when a combination with wavelet decomposition leads to one of the best image de-noising approaches [7]. However, the method proposed in [7] is rather time consuming due to the transformation and re-transformation step from the image in the wavelet domain and vice versa. We are aiming at a fast and reliable method that directly works in the image domain. In this contribution we show how to combine Wiener filter theory with the idea of normalized convolution [9] such that the assumed disadvantages of Wiener filtering mentioned above do not occur. Furthermore, the proposed filter does not only preserve edges, is easy to implement, fast and has only a few number of free parameter, but it also outperforms other spatial de-noising techniques like nonlinear diffusion [2]. The main idea of almost all edge preserving filter techniques is to smooth the image in its homogenous regions and to reduce or even stop the smoothing at edges. A famous example is the nonlinear diffusion scheme [2]

where the diffusion process is driven by the gradient of the underlying image. This principle should also work for Wiener filtering where in homogenous regions the optimal linear filter is applied and in the presence of edge it is stopped. In this contribution we show how to design such an edge-preserving filter. *Tensor valued image processing* has raised much attention in the last couple of years. Therefore, we restrict our filter approach not only to scalar valued image data but consider tensor valued data as well. In fact, most relevant tensors for image processing are positive (semi-) definite matrices. These occur in orientation or motion estimation (structure tensor) [10,11], image de-noising and scale space theory (structure tensors in coherence enhancing diffusion filtering), in statistical description of images or in diffusion tensor magnetic resonance images. Classical approaches to tensor image regularization are commonly done by smoothing the tensor field by a Gaussian filter. Besides the drawback that this method blurs the edges, it is quite arbitrary. The size of the averaging mask as well as the shape have to be chosen ad hoc and thus are not optimized for the current tensor field. Recent approaches trying to handle the problem of blurring structures are either motivated from a deterministic point of view, e.g. they use nonlinear [12] diffusion PDE's to smooth tensor fields, regularization based approaches [13] or classical statistics, i.e. robust estimation [14] or normalized convolution or adaptive anisotropic Gaussian filtering [15]. The Bayesian estimation approaches to tensor fields which has been proposed so far [16,17,18] differs from our Wiener filter approach in terms of computational costs (the Gibbs sampler as well as the simulated annealing algorithm is applied) and in the ability to preserve edges. The main challenge of all tensor processing methods is that the processed tensors are again positive definite. Since the set of positive definite matrices does not form a vector space, simple operations like subtraction lead out of this space. Different strategies have been proposed to cope with this problem in different methods of processing tensor valued data. In a variational approach [13] to tensor smoothing, the different matrix elements are not different treated as scalar valued data. In order to keep the tensors positive definite, the processed tensors are projected back into the set of positive definite matrices for every iteration step in the iterative solution scheme of the corresponding Euler Lagrange equations. In the same paper [13] another approach has been proposed that incorporates additional constraints into the variational approach forcing the processed tensor valued data to stay within the set of positive definite matrices. Generally, these strategies can be classified into two groups. The first group handles tensor valued images like scalar valued data combined with some projection scheme which maps the processed tensor field back into the set of positive definite tensors [13] or with some sort of test if the processed tensors meet the positive definite requirements [16,17,18]. If not the tensors are processed so long as it fullfils the requirement. The second approach designs the methods such that the processed tensor field is again positive (semi-)definite [12]. Our approach belongs to the second approach, designing the filter such that the processed tensors remain in the set of positive definite matrices.

## 2 Designing the Edge Preserving Wiener Filter

In the following, we present the edge preserving optimal estimator for scalar and tensor valued data. This estimator is a combination of the classical Wiener filter [1] with the main idea of normalized convolution [9], i.e. giving pixels different weights. This idea has already exploited to interpolate missing pixels in images [19]. Besides obtaining an interpolation scheme which is optimized according to the signal and noise characteristics, the approach in [19] is superior to normalized convolution in that known pixel values remain unchanged while in normalized convolution also pixels that are known are affected by blurring effects. In this contribution, we use the advantage of both concepts, the Wiener filtering, i.e. adapting the filter to the signal and noise statistics, and normalized convolution, i.e. the ability to design edge preserving filters, to derive an edge preserving signal and noise adapted filter denoted as the edge preserving Wiener (EPW-) filter in the following. Starting with filtering scalar valued images, we generalize this concept later to tensor valued data.

### 2.1 The Signal and Noise Model

The observed image signal  $z$  at position  $i, j$  in a two-dimensional image is modeled by the sum of the ideal (noise free) signal  $s$  and a noise term  $v$ . For the subsequent steps, it is convenient to arrange  $s, v$ , and  $z$  in vectors  $\mathbf{s} \in \mathbb{R}^N, \mathbf{v} \in \mathbb{R}^N$  and  $\mathbf{z} \in \mathbb{R}^N$ , where  $N$  is the number of pixels. Furthermore, the image could be degraded by a linear transformation  $\mathbf{K}$ , denoted as the observation matrix

$$\mathbf{z} = \mathbf{K}\mathbf{s} + \mathbf{v} . \quad (1)$$

This model is commonly used in estimation theory (see e.g. [20]) to model the linear disturbing of the image by blurring effects (modeled by  $\mathbf{K}$ ) and additive noise (modeled by  $\mathbf{v}$ ). Missing data could be modeled by setting the corresponding entries in the observation matrix  $\mathbf{K}$  to zero [19]. In this contribution, we extend this approach to arbitrary real values between zero and one representing our belief into the corresponding observed image value. For the tensor valued case, we apply the model according to equ.(1) to the components of the corresponding tensors.

### 2.2 The Scalar Valued Edge Preserving Wiener Filter

The estimation of a true underlying image value  $s_j$  at position  $j$  from a linear but not shift invariant filtering of the observable image  $\mathbf{z}$  can be written in the form  $\hat{s}_j = \mathbf{m}_j^T \mathbf{z}$ . Our task is to choose  $\mathbf{m}_j$  in such a way that the filtered output  $\hat{s}_j$  approximates, on an average, the desired output  $s_j$  for the error-free case as closely as possible in the least mean squares sense. Therefore, it is necessary to model the statistical properties of the signal and the noise processes, respectively. Let the noise vector  $\mathbf{v} \in \mathbb{R}^N$  be a zero-mean random vector with covariance matrix  $\mathbf{C}_v \in \mathbb{R}^{N \times N}$  (which is in this case equal to its correlation matrix  $\mathbf{R}_v$ ). Furthermore, we assume that the process that generates the signal

$\mathbf{s} \in \mathbb{R}^N$  can be described by the expectation  $\mathbf{w}_s = \mathbb{E}[\mathbf{s}]$  of the signal vector, and its autocorrelation matrix  $\mathbf{R}_s$ . Furthermore, let  $\mathbf{R}_{\mathbf{s}s_j} \in \mathbb{R}^{1 \times N}$  denote the correlation matrix between the image value  $s_j$  and the whole image  $\mathbf{s}$ . The filter  $\mathbf{m}_j$  is then determined by minimizing the mean squared error between the estimated signal value and the actual one

$$\mathbf{m}_j = \arg \min_{\tilde{\mathbf{m}}_j} \{ \mathbb{E} [ \|\tilde{\mathbf{m}}_j^T \mathbf{z} - s_j\|^2 ] \} . \quad (2)$$

Knowing the second order statistical moments for both the noise and signal as well as the observation matrix, the Gauss-Markov theorem delivers the optimal filter (for a detailed derivative of mean squared error based filters see e.g. [20])

$$\mathbf{m}_j = (\mathbf{K}_j \mathbf{R}_s \mathbf{K}_j^T + \mathbf{R}_v)^{-1} \mathbf{K}_j \mathbf{R}_{\mathbf{s}s_j} . \quad (3)$$

In following, we discuss the extension of this concept to matrix valued data.

### 2.3 The Tensor Valued Edge Preserving Wiener Filter

As already mentioned in the introduction, most important tensors for image processing are square positive (semi-)definite matrices denoted by  $P(n)$  in the following where  $n$  is the size of the matrix. This set of tensors does not form a subspace of the tensor vector space. For example, multiplying a positive definite matrix by  $-1$  yields a negative definite matrix and hence leads out of the set  $P(n)$ . Thus, applying image processing techniques to  $P(n)$  requires additional care since even simple linear operations might destroy the basic structure of the data. In [21] the proposed nonlinear diffusion scheme is shown to preserve positiveness of the processed tensor field. An equivalent proof based on discrete filtering can be found in [12] which uses the fact that the proposed diffusion filters are convex filters. This is also the basis for the design of our tensor valued EPW-filter, i.e. we design the tensor valued EPW-filter as a convex filter. A map  $\mathbf{F} : \mathbb{R}^N \rightarrow \mathbb{R}^N$  is denoted as a convex filter (see e.g. [22]) if for each  $\mathbf{z} \in \mathbb{R}^N$  there are weights  $w_{ij}(\mathbf{z})$  with

$$(\mathbf{F}\mathbf{z})_k = \sum_{t=1}^N w_{kt}(\mathbf{z})z_t, \quad w_{tk}(\mathbf{z}) \geq 0 \quad \forall k, \quad \sum_{t=1}^N w_{kt}(\mathbf{z}) = 1 . \quad (4)$$

If each component of the tensor-field is processed with the same convex filter, it is simple to prove the positiveness of the processed tensor field. Let  $\mathbf{A}_j, j \in (1, 2, \dots, n)$  be a set of positive (semi-) definite matrices  $P(N)$ , i.e.  $\mathbf{z}^T \mathbf{A}_j \mathbf{z} \geq 0$  for all  $j$  and  $\mathbf{z}$  unequal zero. Applying a convex filter to each component of the tensor field  $\{\mathbf{A}_j\}$  yields again a positive (semi-)definite tensor field  $\{\tilde{\mathbf{A}}_t\}$

$$\mathbf{z}^T \tilde{\mathbf{A}}_t \mathbf{z} = \sum_{mn} z_m \tilde{a}_{tmn} z_n = \sum_{smn} z_m w_s a_{smn} z_n \quad (5)$$

$$= \sum_{smn} w_s z_m a_{smn} z_n = \sum_s \underbrace{w_s}_{\geq 0} \underbrace{\mathbf{z}^T \mathbf{A}_s \mathbf{z}}_{\geq 0} \geq 0 . \quad (6)$$



This implies that we have to model each matrix component by the same process and thus use the same statistical model as in the scalar case for each matrix element. We have to design a filter mask whose sum is equal one and where each element is non-negative. The first requirement can easily be obtained by a proper normalization. The second requirement is not guaranteed by (2). In order to keep each element non-negative, further constraints are introduced to the optimization procedure

$$\mathbf{m}_j = \arg \min_{\tilde{\mathbf{m}}_j} \{ \mathbf{E} [ \|\tilde{\mathbf{m}}_j^T \mathbf{z} - s_j\|^2 ] \} \quad \text{such that} \quad (\mathbf{m}_j)_k \geq 0 . \quad (7)$$

In contrast to equ.(2), a closed form solution does not exist for the non-negative least squares problem and numerical methods (chapter 23, pp. 161 in [23]) need to be applied.

## 2.4 Certainty Maps for Scalar and Tensor Valued Images

The certainty of an image value, being in our case either a gray value or a symmetric positive matrix, depends on the prior knowledge about these entities. Take for example the pixels outside the image border. Since no information about the gray values at these positions is available, the most reasonable we can do is to set the corresponding entries in the observation matrix  $\mathbf{K}$  equal zero. For generating a general certainty map for entities inside the image domain we describe a similar approach proposed by Westin et al. [24] for normalized convolution which is sketched in the following. Let  $\tilde{z}_j$  and  $\hat{\mathbf{T}}_j$  denotes a smoothed image value (e.g. by Gaussian filtering) at position  $j$ . For each position  $i$ , we label the certainty of its neighborhood values  $j \in N_i$  ( $N_i$ : neighborhood system of position  $i$ ) by  $c_{ij} = \exp(-\alpha(z_i - \tilde{z}_j)^2)$  where the *contrast parameter*  $\alpha$  is optimized from training data. In case of tensor valued data we have to introduce a scalar product  $\langle \cdot, \cdot \rangle$  in the tensor space in order to measure magnitude as well as orientation deviations between neighborhood tensors. The angular certainty  $c_{aij}$  and the magnitude certainty  $c_{mij}$  are defined by

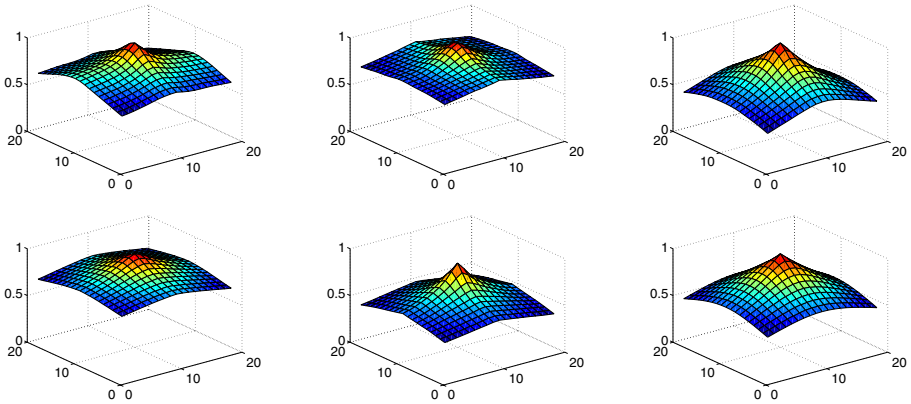
$$c_{aij} = \langle \hat{\mathbf{T}}_i, \hat{\mathbf{T}}_j \rangle^{\beta_1}, \quad c_{mij} = \exp\left(-\beta_2 \left( \|\mathbf{T}_i\| - \|\tilde{\mathbf{T}}_j\| \right)^2\right), \quad (8)$$

where  $\hat{\mathbf{T}}_j = \mathbf{T}_j / \|\mathbf{T}_j\|$  denotes the normalized tensors and  $\beta_1, \beta_2$  are again contrast parameters. The final tensor valued certainty  $c_{tij}$  is then obtained from the product of angular and magnitude certainty  $c_{tij} = c_{aij}c_{mij}$ . The idea of these certainty definitions is the reduction of the influence of outliers as well as the prevention of filter blurring across edges.

## 3 Algorithmic Aspects

In this section several practical issues of the EPW-filter implementation are discussed. Figure 3.1 (upper images and lower left) shows some of the estimated

*autocorrelation functions* (ACFs) of training images<sup>1</sup>. The ACFs of all training images have more or less the same characteristic shape. Thus, modeling all images by a common generating process is appropriate. We choose an isotropic first order autoregressive model with the ACF  $\varphi_{ss}(\mathbf{x}) = \exp(-\gamma|\mathbf{x}|)$  commonly used to characterize the global image statistics. In figure 3.1 the average over 20 ACFs of training images is depicted showing the similarity to the ACF model quite well. The parameter  $\gamma$  of the ACF model is optimized according to the training data. We emphasize that the filters given by equ.(3) as well as equ.(7) do not depend on the absolute powers  $\sigma_s^2$ ,  $\sigma_v^2$  of the signal and noise processes but only on their ratio  $\alpha = \sigma_v^2/\sigma_s^2$ .



**Fig. 3.1.** Upper rows: autocorrelation functions (ACF) estimated from training images; lower left: ACF estimated from a training image; lower middle: average over the ACFs of 20 training images; lower right: model ACF  $\varphi_{ss}(\mathbf{x}) = \exp(-\gamma|\mathbf{x}|)$

Thus, we can normalize the ACF and the filter mask in equ.(3) yields

$$\mathbf{m}_j = \left( \mathbf{K}_j \hat{\mathbf{R}}_s \mathbf{K}_j^T + \alpha \hat{\mathbf{R}}_v \right)^{-1} \mathbf{K}_j \hat{\mathbf{R}}_{ss_j} . \quad (9)$$

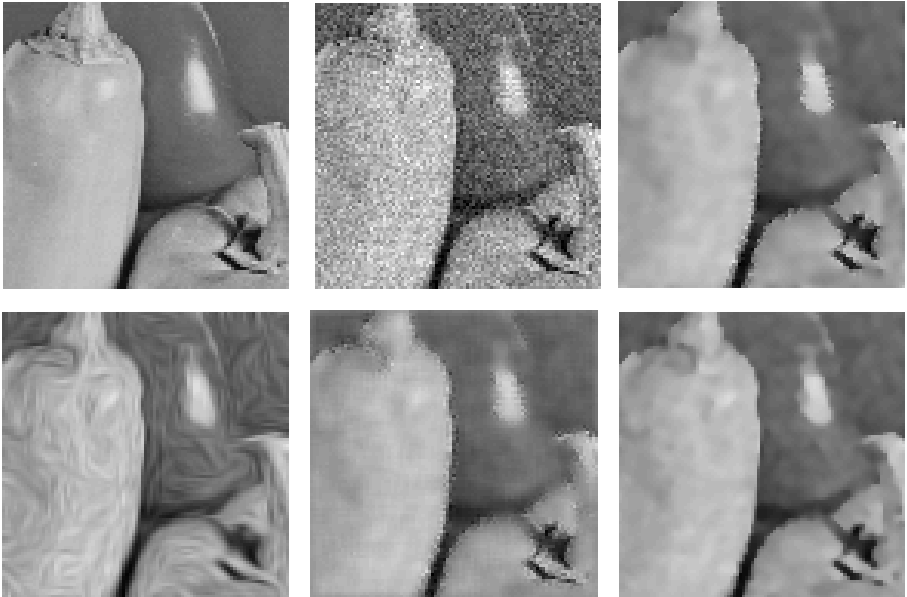
with normalized correlation matrices  $\hat{\mathbf{R}}_s$ ,  $\hat{\mathbf{R}}_v$  and  $\hat{\mathbf{R}}_{ss_j}$ .

## 4 Experimental Results

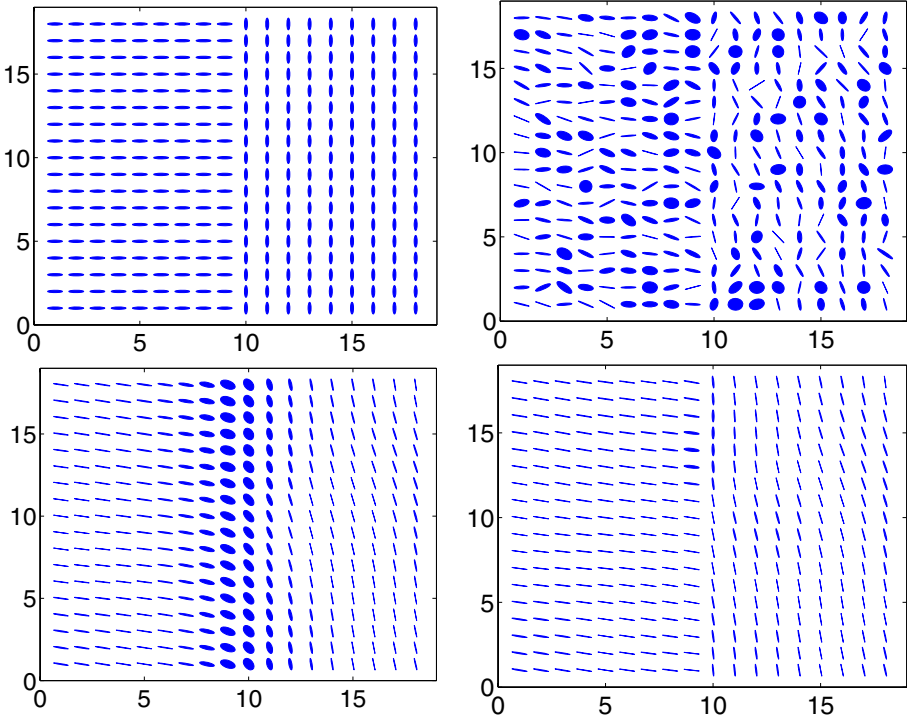
The experiments will demonstrate the performance of the EPW-filter on scalar valued data as well as demonstrate the edge preserving property of the tensor valued filter. In the scalar valued case we compare our method with the locally adaptive Wiener filter (matlab implementation, see also pp. 536-540 in [25]), with isotropic nonlinear and anisotropic (coherence enhancing) nonlinear filtering (We use the matlab toolbox 'nonlinear diffusion toolbox' by Frederico

<sup>1</sup> All 20 training images have been taken from the USC-SIPI image database: 'http://sipi.usc.edu/database/'

D’Almeida). First, we corrupted the test images by additive Gaussian noise with three noise levels ( $\sigma_v = 10, 15, 20$ ). The noise is modeled as identical independent distributed, i.e. the correlation matrix of the noise process is given by  $\mathbf{R}_v = \sigma_v^2 \mathbf{I}$ . Then, for every test image the noise has been estimated according to [25] (see also the Wiener filter matlab implementation). Then, training images have been corrupted by the estimated noise strength and all free parameters of the four different tested methods have been optimized from the training data and afterwards applied to the corresponding test image. Thus, the method for estimating the true underlying images is fully automatic. Table 4.1 shows the result of the experiments where the performance of the individual de-noising methods as well as the amount of noise of the input image have been measured with the peak signal to noise ratio (PSNR). For all three noise levels as well as for all test images our EPW-filter outperforms the other three de-noising methods. Figure 4.1 gives a visual impression of the four different de-noising methods showing a cutout of the original image ‘peppers’, the image corrupted by noise ( $\sigma_v = 20$ ) and the four de-noised images. In order to demonstrate the edge preserving property of the tensor valued EPW-filter, a simple tensor field has been generated. Each component of the positive definite matrices has been corrupted by additive independent normally distributed noise (figure 4.2, upper right). The noise corrupted tensor field has then been filtered with the EPW-filter for the contrast parameters  $\beta_1 = 0, \beta_2 = 0$  (figure 4.2, lower left) and  $\beta_1 = 6, \beta_2 = 1$  (figure 4.2, lower right) showing clearly its edge preserving property.



**Fig. 4.1.** Upper left: original image; upper middle: image corrupted by Gaussian noise; upper right: nonlinear diffusion filtered image; lower left: anisotropic nonlinear filtered image; lower middle: Wiener filtered image; lower right: edge preserving Wiener filtered image



**Fig. 4.2.** Upper left: original tensor field; upper right: left tensor field corrupted by additive Gaussian noise ( $\sigma_v = 0.3$  on each matrix element); lower left: processed tensor field by our EPW-filter with  $\beta_1, \beta_2 = 0$ ; lower right: EPW-filter with  $\beta_1 = 6, \beta_2 = 1$

**Table 4.1.** Results of the de-noising experiment. All test images have been corrupted by additive Gaussian noise with three different noise levels (28.1 dB, 24.6 dB, 22.2 dB). Four different de-noising techniques have been applied: NID: nonlinear isotropic diffusion filter; NAD: nonlinear anisotropic diffusion filter; W: local adaptive Wiener filter; EPW: edge preserving Wiener filter); fingerp.: fingerprint; moons.: moon-surface.

Method	Input PSNR	lena	boat	camera	fingerp.	house	moons.	peppers	bird
NID	22.1	30.4	28.3	27.2	25.6	30.2	30.4	29.3	32.0
NAD	22.1	29.5	28.4	26.6	27.7	29.5	29.1	28.7	29.9
W	22.1	30.1	28.2	27.4	24.3	29.4	30.1	28.9	31.1
<b>EPW</b>	<b>22.1</b>	<b>31.2</b>	<b>29.3</b>	<b>28.2</b>	<b>27.9</b>	<b>31.3</b>	<b>31.0</b>	<b>30.5</b>	<b>32.8</b>
NID	24.6	31.4	28.9	27.7	27.0	31.4	31.1	30.5	33.5
NAD	24.6	31.4	29.8	27.4	28.6	31.2	30.9	30.1	32.0
W	24.6	31.4	29.2	28.4	24.6	30.6	31.1	30.0	32.7
<b>EPW</b>	<b>24.6</b>	<b>32.5</b>	<b>30.0</b>	<b>28.8</b>	<b>28.9</b>	<b>32.5</b>	<b>31.9</b>	<b>31.4</b>	<b>34.5</b>
NID	28.1	32.2	29.5	28.2	28.4	32.5	31.7	31.7	34.9
NAD	28.1	33.4	31.2	28.2	29.5	33.5	32.8	31.6	34.6
W	28.1	32.8	30.2	29.3	24.9	32.1	32.5	31.3	34.6
<b>EPW</b>	<b>28.1</b>	<b>33.5</b>	<b>31.6</b>	<b>29.6</b>	<b>29.9</b>	<b>33.6</b>	<b>33.0</b>	<b>32.5</b>	<b>36.1</b>

## 5 Summary and Conclusion

We combined the edge preserving idea of normalized convolution with the optimal estimation property of Wiener filtering deriving a filter method which considers the signal and noise characteristics as well as preserves edges. Our method is fast (requiring only the computation of the certainty map, the filter design and a sequence of linear filtering afterwards) and outperforms not only the locally adapted Wiener filter but also nonlinear methods like anisotropic and nonlinear diffusion filtering. We combined the principle of convex filtering and optimal linear Bayesian estimation to an optimal linear estimator for tensor valued data.

## References

1. Wiener, N.: The Extrapolation, Interpolation, and Smoothing of Stationary Time Series with Engineering Applications. J. Wiley (1949)
2. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **12** (1990)
3. L. Rudin, S.O., Fatemi, E.: Nonlinear total variation based noise removal algorithms. *Physica D* **60** (1992)
4. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24** (2002)
5. Felsberg, M., Forssén, P.E., Schar, H.: Channel smoothing: Efficient robust smoothing of low-level signal features. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **28** (2006) 209–222
6. Donoho, D.L.: De-noising by soft thresholding. *IEEE Transactions on Inform. Theory* **41** (1995)
7. J. Portilla, V. Strela, M.W., Simoncelli, E.P.: Image denoising using scale mixtures of gaussians in the wavelet domain. *IEEE Trans Image Processing* **12** (2003)
8. Felsberg, M.: Wiener channel smoothing: Robust Wiener filtering of images. In: DAGM 2005. Volume 3663 of LNCS., Springer (2005) 468–475
9. Knutsson, H., Westin, C.F.: Normalized and differential convolution: Methods for interpolation and filtering of incomplete and uncertain data. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition. (1993)
10. Knutsson, H., Granlund, G.: Texture analysis using two-dimensional quadrature filters. In: IEEE Workshop Computer Architecture for Pattern Analysis and Image Data Base Management., Pasadena (CA) (1983)
11. Bigün, J., Granlund, G.H.: Optimal orientation detection of linear symmetry. In: Proc. ICCV, IEEE (1987) 433–438
12. Weickert, J., Brox, T.: Diffusion and regularization of vector- and matrix-valued images. In: M. Z. Nashed, O. Scherzer (Eds.) : Inverse Problems, Image Analysis, and Medical Imaging. Contemporary Mathematics, AMS, Providence (2002) 251–268
13. Tschumperl, D., Deriche, R.: Diffusion tensor regularization with constraints preservation (2001)

14. van den Boomgaard, R., van de Weijer, J.: Robust estimation of orientation for texture analysis. In: Proceedings Texture 2002, 2nd International Workshop on Texture Analysis and Synthesis. (2002)
15. Nagel, H., Gehrke, A.: Spatiotemporally adaptive estimation and segmentation of fields. In Burkhardt, Neumann, eds.: Proc. Europ. Conf. Comp. Vision. Number 1407 in Lecture Notes on Computer Science, Springer Verlag (1998) 305–321
16. Martin-Fernandez, M., San-Jose, R., Westin, C.F., Alberola-Lopez, C.: A novel Gauss-Markov random field approach for regularization of diffusion tensor maps. In Moreno-Diaz, R., Pichler, F., eds.: Ninth International Conference on Computer Aided Systems Theory (EUROCAST'03), Lecture Notes in Computer Science 2809, Las Palmas de Gran Canaria, Spain, Springer Verlag (2003) 506–517
17. Martin-Fernandez, M., Alberola-Lopez, C., Ruiz-Alzola, J., Westin, C.F.: Regularization of diffusion tensor maps using a non-Gaussian Markov random field approach. In Ellis, R.E., Peters, T.M., eds.: Sixth International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI'03), Lecture Notes in Computer Science 2879, Montreal, Canada, Springer Verlag (2003) 92–100
18. Martin-Fernandez, M., Westin, C.F., Alberola-Lopez, C.: 3d bayesian regularization of diffusion tensor MRI using multivariate Gaussian Markov random fields. In: Seventh International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI'04). Lecture Notes in Computer Science, Rennes - Saint Malo, France (2004) 351–359
19. Mühlich, M., Mester, R.: A statistical unification of image interpolation, error concealment, and source-adapted filter design. In: Proc. Sixth IEEE Southwest Symposium on Image Analysis and Interpretation, Lake Tahoe, NV/U.S.A. (2004)
20. Kay, S.M.: Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory. Prentice Hall PTR (1993)
21. T. Brox, J. Weickert, B.B., Mrázek, P.: Nonlinear structure tensors. Revised version of technical report no. 113, Saarland University, Saarbrücken, Germany (2004)
22. Winkler, G.: Image Analysis, Random Fields and Markov Chain Monte Carlo Methods. A Mathematical Introduction. Springer, Berlin (2002)
23. Lawson, C., Hanson, R.: Solving Least-Squares Problems. Prentice-Hall (1974)
24. Westin, C.F., Knutsson, H.: Tensor field regularization using normalized convolution. In Diaz, R.M., Arencibia, A.Q., eds.: Computer Aided Systems Theory (EUROCAST'03), LNCS 2809, Las Palmas de Gran Canaria, Spain, Springer Verlag (2003) 564–572
25. Lim, Jae, D.: Two-Dimensional Signal and Image Processing. Prentice-Hall (1990)

# From Adaptive Averaging to Accelerated Nonlinear Diffusion Filtering

Stephan Didas and Joachim Weickert

Mathematical Image Analysis Group  
Faculty of Mathematics and Computer Science, Building E11  
Saarland University, 66041 Saarbrücken, Germany  
{[didas](mailto:didas@mia.uni-saarland.de), [weickert](mailto:weickert@mia.uni-saarland.de)}@mia.uni-saarland.de  
<http://www.mia.uni-saarland.de>

**Abstract.** Weighted averaging filters and nonlinear partial differential equations (PDEs) are two popular concepts for discontinuity-preserving denoising. In this paper we investigate novel relations between these filter classes: We deduce new PDEs as the scaling limit of the spatial step size of discrete weighted averaging methods. In the one-dimensional setting, a simple weighted averaging of both neighbouring pixels leads to a modified Perona-Malik-type PDE with an additional acceleration factor that provides sharper edges. A similar approach in the two-dimensional setting yields PDEs that lack rotation invariance. This explains a typical shortcoming of many averaging filters in 2-D. We propose a modification leading to a novel, anisotropic PDE that is invariant under rotations. By means of the example of the bilateral filter, we show that involving a larger number of neighbouring pixels improves rotational invariance in a natural way and leads to the same PDE formulation. Numerical examples are presented that illustrate the usefulness of these processes.

## 1 Introduction

Adaptive averaging filters belong to the simplest and most effective tools for image processing. Since taking the average of the grey values of all pixels in a certain spatial neighbourhood is an intuitive concept, already early methods in image processing use averaging filters: For example, in the beginning of the 1980s, Lee presented an averaging filter for image denoising [1]. In the literature, there is a whole variety of methods which use the concept of averaging pixel grey values with weights depending on their tonal<sup>1</sup> and spatial distance. Some examples are *adaptive smoothing* [2] by Saint-Marc et al., *adaptive weights smoothing* [3] by Polzehl and Spokoiny, or the *W-estimator* [4] by Winkler et al. While many averaging filters work iteratively by applying small stencils, the *bilateral filter* of Tomasi and Manduchi [5,6] in its original form is an example for a noniterative averaging method: They proposed to use just one iteration of an averaging scheme with a large stencil. Applications for other tasks than image denoising are investigated by Smith and Brady with the *SUSAN filter* [7].

---

<sup>1</sup> The tonal difference denotes the difference of grey values.

Many local adaptive filters have been introduced in an intuitive manner. Research on finding a systematic theoretical foundation for them started much more recently: Barash [8], for instance, investigated connections between bilateral filtering and nonlinear diffusion with a scalar-valued diffusivity, and Mrázek et al. [9] have introduced a common framework for a number of adaptive filters that is based on minimising suitable energy functions. An overview over several neighbourhood filtering techniques has been given by Buades et al. [10]. They start with integral formulations of neighbouring filters and relate them to methods based on PDEs [10].

In general, PDE approximations of discrete averaging filters can be useful to study the evolution of the results under iterated filtering, to prove equivalence between seemingly different methods, and to investigate why and how a discrete filter deviates from a rotationally invariant behaviour. Last but not least, these scaling limits can also lead to novel PDEs with interesting properties.

The goal of our paper is to perform novel scaling limits of a specific class of discrete adaptive averaging methods. This class includes local filters as well as more global representatives such as bilateral filtering.

This paper is organised as follows: In Section 2 we start with a fully discrete averaging filter and describe how a scaling limit of it can be related to an accelerated variant of the Perona-Malik filter. These ideas are extended to the two-dimensional case in Section 3. They motivate the use of an anisotropic filter similar to the diffusion filter in [11]. In Section 4 we extract the same filter as scaling limit of bilateral filtering. Numerical examples in Section 5 juxtapose the behaviour of the scaling limits to the averaging filters they originate from. Section 6 concludes the paper with a summary.

## 2 Averaging Filters and Scaling Limits in 1-D

**Derivation of the Scaling Limit.** We start with the consideration of an iterative weighted averaging filter of the form

$$u_i^0 = f_i$$

$$u_i^{k+1} = \frac{g\left(\left|\frac{u_{i+1}^k - u_i^k}{h}\right|\right)u_{i+1}^k + g\left(\left|\frac{u_{i-1}^k - u_i^k}{h}\right|\right)u_{i-1}^k}{g\left(\left|\frac{u_{i+1}^k - u_i^k}{h}\right|\right) + g\left(\left|\frac{u_{i-1}^k - u_i^k}{h}\right|\right)} \quad (1)$$

where  $f \in \mathbb{R}^n$  is an initial signal and  $u^k$  denotes the processed signal at iteration  $k \in \mathbb{N}_0$ . For each pixel  $u_i^{k+1}$ , the filter takes the direct neighbours  $u_{i-1}^k$  and  $u_{i+1}^k$  into account for averaging. At the boundaries, we assume mirroring boundary conditions, that means we have two dummy pixels  $u_0^k := u_1^k$  and  $u_{n+1}^k := u_n^k$ . Typically one chooses a decreasing positive function  $g$  such that the denominator cannot be zero. This also implies that we always have convex combinations which guarantees a maximum-minimum principle for the filter. One may use e.g. the same function  $g$  as the diffusivities in nonlinear diffusion filtering [12], for instance  $g(s) = (1 + s^2/\lambda^2)^{-1}$ . We observe that the weights depend on the



tonal distance between the pixel and its direct neighbours divided by the spatial step size  $h > 0$  between the two pixels. We introduce the abbreviations  $g_{i+\frac{1}{2}}^k := g\left(\left|\frac{u_{i+1}^k - u_i^k}{h}\right|\right)$  and rewrite (1) as

$$u_i^{k+1} = \frac{g_{i+\frac{1}{2}}^k u_{i+1}^k + g_{i-\frac{1}{2}}^k u_{i-1}^k}{g_{i+\frac{1}{2}}^k + g_{i-\frac{1}{2}}^k} \quad (2)$$

$$= u_i^k + \frac{g_{i+\frac{1}{2}}^k (u_{i+1}^k - u_i^k) - g_{i-\frac{1}{2}}^k (u_i^k - u_{i-1}^k)}{g_{i+\frac{1}{2}}^k + g_{i-\frac{1}{2}}^k} \quad (3)$$

$$= u_i^k + \frac{\frac{1}{h} \left( g_{i+\frac{1}{2}}^k \frac{u_{i+1}^k - u_i^k}{h} - g_{i-\frac{1}{2}}^k \frac{u_i^k - u_{i-1}^k}{h} \right)}{\frac{1}{h^2} \left( g_{i+\frac{1}{2}}^k + g_{i-\frac{1}{2}}^k \right)} \quad (4)$$

In this last form we notice that the iterative scheme contains finite differences which approximate spatial derivatives of  $u$ . Now we assume that  $u$  and  $g$  are sufficiently smooth to perform a Taylor expansion. For example, there appears  $\frac{u_{i+1}^k - u_i^k}{h} = \partial_x u_{i+\frac{1}{2}}^k + \mathcal{O}(h^2)$  in the scheme. Together with the abbreviations introduced above this yields

$$g_{i+\frac{1}{2}}^k + g_{i-\frac{1}{2}}^k = g\left(\left|\frac{u_{i+1}^k - u_i^k}{h}\right|\right) + g\left(\left|\frac{u_i^k - u_{i-1}^k}{h}\right|\right) = 2g(|\partial_x u_i^k|) + \mathcal{O}(h^2) \quad (5)$$

and thus we can write

$$u_i^{k+1} = u_i^k + \frac{\partial_x (g(|\partial_x u_i^k|) \partial_x u) + \mathcal{O}(h^2)}{\frac{1}{h^2} (2g(|\partial_x u_i^k|) + \mathcal{O}(h^2))} . \quad (6)$$

To understand the iteration indices  $k+1$  and  $k$  as discrete samples of a continuous time variable  $t$  we introduce a temporal step size  $\tau > 0$ . Division of both sides by  $\tau$  leads to the equation

$$\frac{u_i^{k+1} - u_i^k}{\tau} = \frac{\partial_x (g(|\partial_x u_i^k|) \partial_x u) + \mathcal{O}(h^2)}{\frac{\tau}{h^2} (2g(|\partial_x u_i^k|) + \mathcal{O}(h^2))} \quad (7)$$

where the left-hand side is an approximation for the temporal derivative  $\partial_t u$  at time level  $k$  with an error in the order  $\mathcal{O}(\tau)$ . We set the ratio between  $h$  and  $\tau$  such that  $\frac{\tau}{h^2} = \frac{1}{2}$  and let  $h$  tend to zero. Then (7) approximates

$$\partial_t u = \frac{1}{g(|\partial_x u|)} \partial_x (g(|\partial_x u|) \partial_x u) \quad (8)$$

with an error in the order of  $\mathcal{O}(\tau + h^2)$ . This equation is similar to the nonlinear diffusion equation presented by Perona and Malik [12]:

$$\partial_t u = \partial_x (g(|\partial_x u|) \partial_x u) . \quad (9)$$

The only difference is the factor  $\frac{1}{g(|\partial_x u|)}$  on the right-hand side which acts as an acceleration of the Perona-Malik filtering process at edges. To understand this, assume that  $|\partial_x u|$  is relatively small within a region. A classical Perona-Malik diffusivity is close to 1 in this case, and the factor has only a small effect. More interesting is the situation near an edge where  $\partial_x u$  has large absolute value, and backward diffusion can occur for the diffusivities presented by Perona and Malik. In this case,  $g(|\partial_x u|)$  is close to zero, and thus  $\frac{1}{g(|\partial_x u|)}$  leads to an amplification of the backward diffusion behaviour. We can expect such equations to yield sharper results than classical Perona-Malik PDEs. On the other hand, they do not necessarily preserve the average grey value, since they cannot be written in divergence form.

**Stability of an Explicit Discretisation.** Since classical diffusivities  $g$  may be arbitrary close to zero, the fraction  $\frac{1}{g(|\partial_x u|)}$  in (8) is not bounded. This might give rise to concerns regarding stability. However, the weighted averaging scheme (1) inspires also ways how to obtain stable discretisations: An explicit Euler scheme for (8) can be written as

$$\begin{aligned}
 u_i^{k+1} &= u_i^k + \tau \frac{2}{g_{i+\frac{1}{2}}^k + g_{i-\frac{1}{2}}^k} \frac{1}{h} \left( g_{i+\frac{1}{2}}^k \frac{u_{i+1}^k - u_i^k}{h} - g_{i-\frac{1}{2}}^k \frac{u_i^k - u_{i-1}^k}{h} \right) \quad (10) \\
 &= \frac{2\tau}{h^2} \frac{g_{i+\frac{1}{2}}^k}{g_{i+\frac{1}{2}}^k + g_{i-\frac{1}{2}}^k} u_{i+1}^k + \left( 1 - \frac{2\tau}{h^2} \right) u_i^k + \frac{2\tau}{h^2} \frac{g_{i-\frac{1}{2}}^k}{g_{i+\frac{1}{2}}^k + g_{i-\frac{1}{2}}^k} u_{i-1}^k \quad (11)
 \end{aligned}$$

with the same notation as above and with mirroring boundary conditions. We note that the factors in front of  $u_{i+1}^k$ ,  $u_i^k$  and  $u_{i-1}^k$  sum up to 1. For  $\tau \leq \frac{h^2}{2}$  all three factors are nonnegative, and thus  $u_i^{k+1}$  is a convex combination of the three pixels: the scheme is maximum-minimum-stable. Further we see that for the limit  $\tau = \frac{h^2}{2}$  we obtain exactly the averaging filter (1). It should be noted that the stability of our scheme is a consequence of the arithmetic mean used in the fraction in (10) to approximate the diffusivity at the position of the pixel  $x_i$ .

**A Weighted Averaging Variant Involving the Central Pixel.** The filter (1) does not involve the central pixel  $u_i$  itself in the average. This might cause problems for certain initial signals: If we choose  $f$  to be an alternating signal with two different values, then applying the filter will simply exchange the grey values. To avoid this problem one can give the central pixel a nonnegative weight and involve it in the averaging process. For example, such a modified scheme could look like

$$u_i^{k+1} = \frac{g \left( \left| \frac{u_{i+1}^k - u_i^k}{h} \right| \right) u_{i+1}^k + \alpha u_i^k + g \left( \left| \frac{u_{i-1}^k - u_i^k}{h} \right| \right) u_{i-1}^k}{g \left( \left| \frac{u_{i+1}^k - u_i^k}{h} \right| \right) + \alpha + g \left( \left| \frac{u_{i-1}^k - u_i^k}{h} \right| \right)} \quad (12)$$

where we have given the central pixel a fixed weight  $\alpha > 0$ . The same reasoning as presented above relates this averaging filter to the PDE

$$\partial_t u = \frac{1}{\frac{\alpha}{2} + g(|\partial_x u|)} \partial_x (g(|\partial_x u|) \partial_x u) . \quad (13)$$

Here we see that there is still some factor influencing the velocity of the diffusion process, but this factor now is bounded from above to  $\frac{2}{\alpha}$ . Compared to (8), this slows down the evolution in regions with small derivatives of  $u$ .

### 3 Averaging Filters and Scaling Limits in 2-D

In this section we consider filtering of images with a two-dimensional domain with weighted averaging over the direct neighbouring pixels. Let  $\mathcal{N}(i)$  be the set of indices of the maximal four direct neighbours of the pixel with index  $i$ . Then an equivalent of the weighted averaging filter (1) in two dimensions can be written as

$$u_i^{k+1} = \frac{\sum_{j \in \mathcal{N}(i)} g\left(\left|\frac{u_j^k - u_i^k}{h}\right|\right) u_j^k}{\sum_{j \in \mathcal{N}(i)} g\left(\left|\frac{u_j^k - u_i^k}{h}\right|\right)} . \quad (14)$$

Numerator and denominator of this scheme can be understood as the sum of numerators and denominators of two one-dimensional schemes in  $x$ - and  $y$ -direction. Thus the reasoning described in the last section shows that (14) is a consistent approximation for

$$\partial_t u = \frac{\partial_x (g(|\partial_x u|) \partial_x u) + \partial_y (g(|\partial_y u|) \partial_y u)}{g(|\partial_x u|) + g(|\partial_y u|)} . \quad (15)$$

This equation is not rotationally invariant, and thus will lead to artifacts in images with rotational invariant objects. This indicates that also the weighted averaging method (14) leads to such artifacts which is shown with a practical example in Fig. 2.

To circumvent this shortcoming, we understand equation (15) as a crude approximation of the rotationally invariant equation

$$\partial_t u = \frac{1}{\int_0^\pi g(|\partial_{e_\varphi} u|) d\varphi} \cdot \int_0^\pi \partial_{e_\varphi} (g(|\partial_{e_\varphi} u|) \partial_{e_\varphi} u) d\varphi \quad (16)$$

where we write  $e_\varphi = (\cos(\varphi), \sin(\varphi))^T$  for the unit vector in direction  $\varphi$ . In (15) the integrals are approximated as trapez sums where only two evaluation points of the integrands are used. Similar to [13] we introduce a smoothing of the argument of the diffusivity by the convolution of  $u$  with a Gaussian kernel of standard deviation  $\sigma$ , and we write  $u_\sigma = K_\sigma * u$ . This convolution can also

simply be introduced in the arguments of the weights used in the averaging process (14). It does not affect the reasoning leading to the PDE (16).

An equation similar to (16) has been studied in [11] in the context of anisotropic diffusion filtering:

$$\partial_t u = \frac{2}{\pi} \int_0^\pi \partial_{e_\varphi} (g(|\partial_{e_\varphi} u_\sigma|) \partial_{e_\varphi} u) d\varphi. \tag{17}$$

The proofs in [11] can be applied to show that (16) can be transformed into

$$\partial_t u = \frac{1}{\text{trace}(D(\nabla u_\sigma))} \text{div}(D(\nabla u_\sigma) \cdot \nabla u) \tag{18}$$

with the diffusion tensor  $D(\nabla u_\sigma) = \int_0^\pi e_\varphi e_\varphi^\top g(|\partial_{e_\varphi} u_\sigma|) d\varphi$ . In [11] the eigenvectors of this diffusion tensor  $D(\nabla u_\sigma)$  are calculated as  $v_1(\psi) = (-\sin(\psi), \cos(\psi))^T$  and  $v_2(\psi) = (\cos(\psi), \sin(\psi))^T$  where  $\nabla u_\sigma \neq 0$  and  $(r, \varphi)$  are the polar coordinates of  $\nabla u_\sigma$ . That means  $v_1$  is the direction of the isophote of  $u_\sigma$  (along an edge), while  $v_2$  is the direction across the edge. The corresponding eigenvalues are given by

$$\lambda_1(\nabla u_\sigma) = \int_0^\pi \sin^2(\varphi) g(|\partial_{e_\varphi} u_\sigma|) d\varphi \quad \text{and} \tag{19}$$

$$\lambda_2(\nabla u_\sigma) = \int_0^\pi \cos^2(\varphi) g(|\partial_{e_\varphi} u_\sigma|) d\varphi. \tag{20}$$

Equation (18) is the relevant formulation for practical implementations. This equation is rotationally invariant, since the eigenvectors follow a rotation of the input image, and the eigenvalues are invariant under image rotations. Since  $\text{trace}(D(\nabla u_\sigma))$  is always significantly larger than zero, the sharpening of the edges will be less pronounced in this anisotropic case (similar to (13)). Nevertheless, we are going to see with numerical examples that not only preservation of edges, but also sharpening is possible with this filter.

## 4 Larger Neighbourhood and Rotational Invariance

In the last section we have derived an anisotropic PDE filter from a weighted averaging of the direct neighbouring pixels. To circumvent the lack of rotational invariance in (15) we have understood it as a very crude approximation of the rotational invariant approach (16). Nevertheless, there are discrete filters which address the problem of lacking rotational invariance by involving information from pixels in a larger neighborhood.

We consider here the prominent example of the bilateral filter [5,8,6]. Even though this filter is proposed as a noniterative method, it can make sense to perform several filtering steps; thus we understand it as an iterative averaging filter. In one filtering step, not only the direct neighbouring pixels are involved

in the averaging, but an extended neighbourhood  $i + \mathcal{B}_R$ . Here  $\mathcal{B}_R = \{j \in \mathbb{R}^2 : |j| \leq R\} \cap G_h$  denotes the intersection of the disc of radius  $R$  in  $\mathbb{R}^2$  with the pixel grid  $G_h$ . A variant of the bilateral filter then looks like this:

$$u_i^{k+1} = \frac{\sum_{j \in \mathcal{B}_R} g\left(\frac{|u_{i+j}^k - u_i^k|}{|j|}\right) \frac{w(|j|)}{|j|^2} u_{i+j}^k}{\sum_{j \in \mathcal{B}_R} g\left(\frac{|u_{i+j}^k - u_i^k|}{|j|}\right) \frac{w(|j|)}{|j|^2}}. \quad (21)$$

The spatial distance between  $u_i$  and  $u_{i+j}$  results in a usually smaller weight  $w(|j|)/|j|^2$ , where an example for  $w$  is  $w(h) = h^2 \exp(-h^2)$ . In this special example,  $w$  leads to a Gaussian weight depending on the distance of the two pixels.

We now want to imitate the approach presented in Section 2. To this end we only consider one half of the disc  $\mathcal{B}_R^+ = \{(x, y) \in \mathcal{B}_R | x \geq 0\}$  and rewrite the sums in (21) as

$$u_i^{k+1} = \frac{\sum_{j \in \mathcal{B}_R^+} \frac{w(|j|)}{|j|^2} \left( g\left(\frac{|u_{i+j}^k - u_i^k|}{|j|}\right) u_{i+j}^k + g\left(\frac{|u_i^k - u_{i-j}^k|}{|j|}\right) u_{i-j}^k \right)}{\sum_{j \in \mathcal{B}_R^+} \frac{w(|j|)}{|j|^2} \left( g\left(\frac{|u_{i+j}^k - u_i^k|}{|j|}\right) + g\left(\frac{|u_i^k - u_{i-j}^k|}{|j|}\right) \right)}. \quad (22)$$

The novelty in this two-dimensional case is that we have to consider several directional derivatives. We see that there appear directional finite differences in (22). Let  $e_\varphi = \frac{j}{|j|}$  be the unit vector pointing in the direction of  $j \neq 0$ , and  $h = |j|$  be the length of the vector  $j$ . A Taylor expansion of  $u$  around the pixel  $i$  yields

$$u_{i+j} = u_i + \langle \nabla u, j \rangle + \mathcal{O}(h^2) = u_i + (\partial_{e_\varphi} u) \cdot h + \mathcal{O}(h^2)$$

which will be useful in the form

$$\frac{u_{i+j} - u_i}{h} = \partial_{e_\varphi} u + \mathcal{O}(h). \quad (23)$$

Applying the Taylor formula (23) to (22) allows us to write

$$u_i^{k+1} - u_i^k = \frac{\sum_{j \in \mathcal{B}_R^+} w(h) (\partial_{e_\varphi} (g(|\partial_{e_\varphi} u|) \partial_{e_\varphi} u) + \mathcal{O}(h^2))}{\sum_{j \in \mathcal{B}_R^+} \frac{2w(h)}{h^2} (g(|\partial_{e_\varphi} u|) + \mathcal{O}(h^2))} \quad (24)$$

At this point we investigate the scaling limit if we let the spatial step sizes in  $x$ - and  $y$ -direction tend to zero while we keep the size  $R$  of the neighbourhood fixed. This means that the number of grid points in our neighbourhood  $\mathcal{B}_R$  is tending to infinity. Thus we can consider the sums in (24) as Riemann sums which approximate integrals over the set  $\mathcal{B}_R^+$ :

$$u_i^{k+1} - u_i^k = \frac{1}{\int_0^R \frac{2w(h)}{h^2} \int_0^\pi g(|\partial_{e_\varphi} u^k|) d\varphi dh} \cdot \int_0^R w(h) \int_0^\pi \partial_{e_\varphi} (g(|\partial_{e_\varphi} u^k|) \partial_{e_\varphi} u^k) d\varphi dh \quad (25)$$

Since the inner integrals do not depend on the radius  $r$ , the outer ones are just a scaling factor, that means (25) is corresponding up to a constant factor to

$$u^{k+1} - u^k = \frac{1}{\int_0^\pi g(|\partial_{e_\varphi} u^k|) d\varphi} \cdot \int_0^\pi \partial_{e_\varphi} (g(|\partial_{e_\varphi} u^k|) \partial_{e_\varphi} u^k) d\varphi . \quad (26)$$

If we understand the right-hand side as temporal forward difference we can see (26) as an approximation to (16). This provides a novel interpretation of bilateral filtering as an anisotropic PDE.

## 5 Experiments

Now we show some numerical examples to illustrate the practical behaviour of averaging methods and our novel PDE methods. As weight function or diffusivity we use the classical diffusivity  $g(s) = \left(1 + \frac{s^2}{\lambda^2}\right)^{-1}$  by Perona and Malik [12].

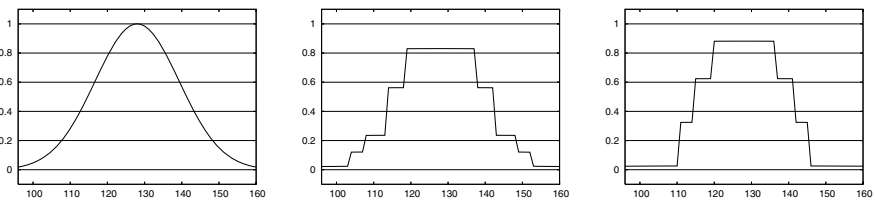
First we display an experiment in the one-dimensional case in Fig. 1. We see that the presence of the acceleration factor allows for sharper edges. With the same evolution time we can achieve a stronger edge enhancement than with a classical nonlinear diffusion equation of Perona-Malik type.

Figure 2 visualises the lack of rotational invariance of local averaging filters and how it can be improved with a larger neighbourhood in the bilateral filter. Even a better effect than extending the neighbourhood can be achieved with the anisotropic nonlinear diffusion equation (16).

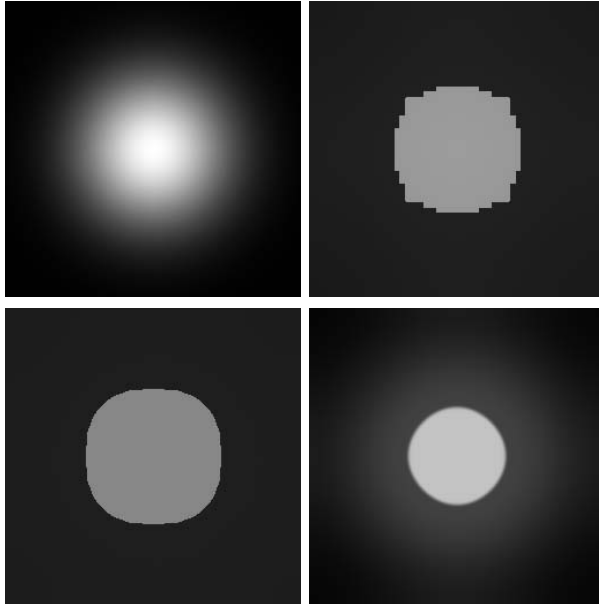
Figure 3 shows the denoising capabilities of the anisotropic diffusion equation (16) for real-world data. The anisotropic behaviour is clearly visible.

## 6 Conclusions

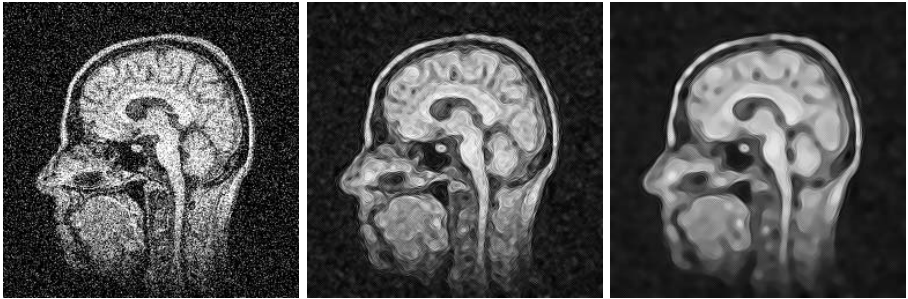
We have described the close relationship between weighted averaging processes and filters based on partial differential equations with an acceleration factor. In



**Fig. 1.** Accelerated Perona-Malik diffusion in 1-D. Left: Original signal (64 point width section of a signal with 256 pixels). Middle: Perona-Malik diffusion ( $\lambda = 0.005$ ,  $t = 5000$ ). Right: Perona-Malik diffusion with additional factor (8) and the same parameters.



**Fig. 2.** Weighted averaging and accelerated diffusion. Top Left: Original image (size: 256 x 256 pixels). Top Right: Weighted averaging (equation (14),  $\lambda = 3.0$ , 15000 iterations). Bottom Left: Iterated bilateral filtering ( $\lambda = 3.0$ , window size  $5 \times 5$  pixels,  $w(h) = h^2 \exp(-h^2/4)$ , 5000 iterations). Bottom Right: Accelerated anisotropic diffusion ( $\lambda = 10$ ,  $\sigma = 2$ ,  $t = 1660$ ).



**Fig. 3.** Accelerated diffusion. Left: Original image (size: 256 x 256 pixels) and additive Gaussian noise with standard deviation 50. Middle: Accelerated anisotropic diffusion ( $\lambda = 2$ ,  $\sigma = 3$ ,  $t = 2$ ). Right: Same, but with  $t = 10$ .

the 1-D setting we have shown that a suitable scaling limit leads to a modification of the nonlinear diffusion filter of Perona and Malik [12]. The modification consists of a factor that accelerates the sharpening of edges and may give an improved edge enhancement. In the two-dimensional setting, choosing only a small neighbourhood for the averaging can lead to lack of rotational invariance.

However, it can be regarded as a crude approximation of a rotationally invariant PDE that resembles the anisotropic diffusion filter of Weickert [11]. We have also derived the same PDE as a scaling limit of bilateral filtering. This provides additional insights in the behaviour of the widely-used bilateral filter, and shows a way how to improve its invariance under rotations. It is our hope that these examples will motivate more people to analyse the fruitful connections between averaging filters and PDE-based methods in the future.

**Acknowledgements.** We gratefully acknowledge partly funding by the *Deutsche Forschungsgemeinschaft (DFG)*, project WE 2602/2-2.

## References

1. Lee, J.S.: Digital image smoothing and the sigma filter. *Computer Vision, Graphics, and Image Processing* **24** (1983) 255–269
2. Saint-Marc, P., Chen, J.S., Medioni, G.: Adaptive smoothing: A general tool for early vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **13** (1991) 514–529
3. Polzehl, J., Spokoiny, V.: Adaptive weights smoothing with applications to image restoration. *Journal of the Royal Statistical Society, Series B* **62** (2000) 335–354
4. Winkler, G., Aurich, V., Hahn, K., Martin, A.: Noise reduction in images: Some recent edge-preserving methods. *Pattern Recognition and Image Analysis* **9** (1999) 749–766
5. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and colour images. In: *Proc. of the 1998 IEEE International Conference on Computer Vision, Bombay, India*, Narosa Publishing House (1998) 839–846
6. Elad, M.: On the origin of the bilateral filter and ways to improve it. *IEEE Transactions on Image Processing* **11** (2002) 1141–1151
7. Smith, S.M., Brady, J.M.: SUSAN – A new approach to low level image processing. *International Journal of Computer Vision* **23** (1997) 43–78
8. Barash, D.: A fundamental relationship between bilateral filtering, adaptive smoothing and the nonlinear diffusion equation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24** (2002) 844–847
9. Mrázek, P., Weickert, J., Bruhn, A.: On robust estimation and smoothing with spatial and tonal kernels. In Klette, R., Kozera, R., Noakes, L., Weickert, J., eds.: *Geometric Properties for Incomplete Data*. Volume 31 of *Computational Imaging and Vision*. Springer, Dordrecht (2006) 335–352
10. Buades, A., Coll, B., Morel, J.M.: A review of image denoising algorithms, with a new one. *Multiscale Modeling and Simulation* **4** (2005) 490–530
11. Weickert, J.: Anisotropic diffusion filters for image processing based quality control. In Fasano, A., Primicerio, M., eds.: *Proc. Seventh European Conference on Mathematics in Industry*. Teubner, Stuttgart (1994) 355–362
12. Perona, P., Malik, J.: Scale space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Anal. Mach. Intell.* **12** (1990) 629–639
13. Catté, F., Lions, P.L., Morel, J.M., Coll, T.: Image selective smoothing and edge detection by nonlinear diffusion. *SIAM Journal on Numerical Analysis* **29** (1992) 182–193



# Introducing Dynamic Prior Knowledge to Partially-Blurred Image Restoration

Hongwei Zheng and Olaf Hellwich

Computer Vision & Remote Sensing, Berlin University of Technology  
Franklinstrasse 28/29, Office FR 3-1, D-10587 Berlin  
{hzheng, hellwich}@cs.tu-berlin.de

**Abstract.** The paper presents an unsupervised method for partially-blurred image restoration without influencing unblurred regions or objects. Maximum a posteriori estimation of parameters in Bayesian regularization is equal to minimizing energy of a dataset for a given number of classes. To estimate the point spread function (PSF), a parametric model space is introduced to reduce the searching uncertainty for PSF model selection. Simultaneously, PSF self-initializing does not rely on supervision or thresholds. In the image domain, a gradient map as *a priori* knowledge is derived not only for dynamically choosing nonlinear diffusion operators but also for segregating blurred and unblurred regions via an extended graph-theoretic method. The cost functions with respect to the image and the PSF are alternately minimized in a convex manner. The algorithm is robust in that it can handle images that are formed in variational environments with different blur and stronger noise.

## 1 Introduction

The challenge of blind image deconvolution (BID) is to uniquely define the optimized signals from degraded images with unknown blur information, which is an ill-posed problem in the sense of Hadamard. However, knowledge of the direct model is not sufficient to determine an existing, unique and stable solution, and it is necessary to regularize the solution using some *a priori* knowledge. Mathematically, the *a priori* knowledge is often expressed through a regularization theory [1] which replaces an ill-posed problem by a well-posed problem with an acceptable approximation to the solution.

In the real world, CCD and CMOS camera images or medical images are often blurred or partially-blurred in a stationary or non-stationary way. BID of partially-blurred images is to restore blurred regions without influencing unblurred regions for achieving better visual perception based on the Gestalt theory, shown in Fig. 1. To achieve this, blurred regions or objects have to be blur identified, segregated and restored respectively. Different characteristic properties [2,3] (gradient, frequency, entropy, etc.) between blurred and unblurred regions or objects endowed with pairwise relationships can be naturally considered as a graph. Thus, we treat BID of partially-blurred images as a combinatorial

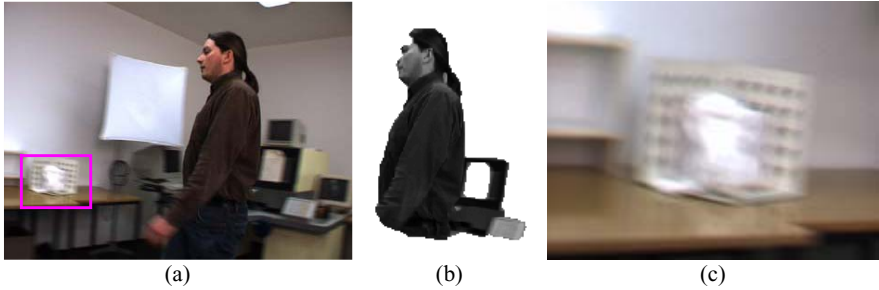
optimization problem including graph partitioning [4,5], blur identification and edge-driven image restoration.

Our work relates to the deterministic edge-preserving image restoration [6,7], and nonlinear filtering techniques incorporated in variational methods [8,9,10]. Since the traditional edge-preserving methods have some limitations of noise processing due to the unique diffusion operator and passively edge-preserving processes. Likewise, most nonlinear diffusion techniques are studied for the well adapted input data with underlying geometric assumptions [11]. Recently, variational regularization for image restoration [12,13,14] have been investigated. These methods are observed in underutilizing prior information. The initialization problem for deterministic optimization is still not progressively solved. Even if a unique solution exists, a proper initialization value is still intractable, e.g., when the cost function is non-convex, convergence to local minima often occurs without proper initialization. [15] have reported that the estimates for the PSF could vary significantly, depending on the initialization. For restoring partially-blurred images, it is also not possible to directly apply these methods. Hence, general strategies are still needed for solving the blur problem as well as achieving better visual perception for partially-blurred image restoration.

In this paper, we propose a Bayesian based variational regularization. In Section 2, it is shown that the Bayesian approach [15,16] provides a structured way for introducing prior knowledge from the image domain and the PSF domain. In Section 3 and 4, alternate blur identification and edge-driven image restoration are discussed. The proposed blur kernel space including most existing PSF parametric models reduces the searching dimension and uncertainty efficiently for PSF self-learning. Thus, the self-learning PSF can be an accurate initial value for yielding a unique solution. Alternately, computed edge gradients are derived as *a priori* knowledge for choosing image diffusion operators [13,17] in a dynamic and continuous processing mechanism. An extended bisection-partitioning method for identifying and segregating blurred and unblurred regions or objects is presented in Section 5. Different edge gradients and blur information in different regions are *a priori* knowledge to achieve a high perceptual quality segmentation according to a clustering criterion [4] without any supervision. Likewise, deconvolution and edge-driven image diffusion improve perceptual quality for restoring partially-blurred images without influencing unblurred regions or objects. The experimental results are shown in Section 6. Conclusions are presented in Section 7.

## 2 Bayesian Estimation Based Variational Regularization

An observed image  $g$  in the image plane is normally an ideal image  $f$  in the object plane degraded by two unknown factors, including linear space-invariant blur kernel (PSF)  $h$  and additive white Gaussian noise  $n$ . It can be formulated in a lexicographic notation,  $g = h * f + n$ , where  $*$  denotes two-dimensional convolution. Following a Bayesian paradigm, the ideal image  $f$ , the PSF  $h$  and the observed image  $g$  fulfill  $p(f, h|g) = p(g|f, h)p(f, h)/p(g) \propto p(g|f, h)p(f, h)$ . Based



**Fig. 1.** (a) Original video. (b) Identified unblurred region. (c) Blurred background.

on this form, our goal is to find the optimal  $\hat{f}$  and  $\hat{h}$  that maximizes the posterior  $p(f, h|g)$ .  $\mathcal{F}(f|h, g) = -\log\{p(g|f, h)p(f)\}$  and  $\mathcal{F}(h|f, g) = -\log\{p(g|f, h)p(h)\}$  express that the energy cost  $\mathcal{F}$  is equivalent to the negative log-likelihood of the data. The priors  $p(f)$  and  $p(h)$  over the parameters are penalty terms added to the cost function to minimize the energy cost in a regularization framework for solving ill-posed problems [1,6,15,16]. To avoid stochastic optimization (longer computing time)[18,19,20], we solve the optimization problem deterministically [6,7,21] in a convex manner with respect to the image and the PSF. The proposed variational double regularized energy functional in a Bayesian framework is formulated according to

$$\mathcal{F}(\hat{f}, \hat{h}) = \underbrace{\int_{\Omega} (g - \hat{h} * \hat{f})^2 dA}_{\text{fidelityTerm}} + \lambda \underbrace{\int_{\Omega} \phi_{\varepsilon}(x, \nabla \hat{f}) dA}_{\text{imageSmoothing}} + \beta \underbrace{\int_{\Omega} |\nabla \hat{h}| dA}_{\text{psfSmoothing}} + \gamma \underbrace{\int_{\Omega} |\hat{h} - \hat{h}_f| dA}_{\text{psfLearning}}$$

where  $dA = dx dy$ . The estimates of the ideal image  $f$  and the PSF  $h$  are denoted by  $\hat{f}$  and  $\hat{h}$  respectively which can be iteratively alternating minimized (AM) [21]. The image smoothing term is a variable exponent, nonlinear diffusion term [13]. The PSF smoothing term represents the regularization of blur kernels. The flexibility of the last term denotes the PSF learning decision of the best-fit parametric model  $\hat{h}_f$ . The primary objective of this learning decision approach is to evaluate the relevance of the parametric structure and integrate the information into the learning scheme accordingly. It can adjust and incorporate the PSF parametric model throughout the process of blur identification and image restoration.

### 3 Simultaneous Model Selection and Self-initializing PSF

Generally, power spectral densities vary considerably from low frequency uniform regions to medium or high frequency discontinuities and texture regions in a real blurred image. Moreover, most PSFs exist in the form of low-pass filters. To a certain degree, PSF models in numerous real blurred images can be represented as parametric models. Blur identification can be based on these characteristic properties. Compared to general priors, e.g., Gibbs distribution [18,19], smoothness

prior [15] or maximum entropy [18], we define a set of primary parametric blur models in *a priori* model space  $\Theta$  for model selection based Bayesian PSF estimation. It consists of the most typical blur models in  $\Theta = \{h_i(\theta), i = 1, 2, 3, \dots, N\}$ .  $h_i(\theta)$  represents the  $i$ th PSF parametric model with its defining parameters  $\theta$ , and  $N$  is the number of blur types such as pillbox blur, Gaussian blur, 1D and 2D linear motion blur and out-of-focus blur, etc.

Based on the model space, an unsupervised self-initializing PSF learning term can learn a PSF parametric model according to the following energy functional,

$$\mathcal{F}(\hat{h}|\hat{f}, g) = \frac{1}{2} \int_{\Omega} (g - \hat{h} * \hat{f})^2 dA + \beta \int_{\Omega} |\nabla \hat{h}| dA + \gamma \int_{\Omega} |\hat{h} - \hat{h}_f| dA \quad (1)$$

where the likelihood is  $p(g|\hat{f}, \hat{h}) \propto \exp\{-\frac{1}{2} \int_{\Omega} (g - \hat{h} * \hat{f})^2 dA\}$ ,  $p(\hat{h})$  is the prior density. Since an image represents intensity distributions that cannot take negative values, the PSF coefficients are always nonnegative,  $h(x) \geq 0$ . Furthermore, since image formation system normally do not absorb or generate energy, the PSF should fulfill  $\sum_{x \in \Omega} h(x) = 1.0$ ,  $x \in \Omega$  and  $\Omega \subset R^2$ .

A MAP estimator is used to determine the best fit model  $h_i(\theta^*)$  for the estimated PSF  $\hat{h}$  in resembling the  $i$ th parametric model  $h_i(\theta)$  in a multivariate Gaussian distribution. The subscript  $i$  denotes the index of blur kernel.  $h_i(\theta^*) = \arg \max_{\theta} \{(2\pi)^{-\frac{LB}{2}} |\sum_{dd}^{-1}|^{-\frac{1}{2}} \cdot \exp[-\frac{1}{2}(h_i(\theta) - \hat{h})^T \sum_{dd}^{-1}(h_i(\theta) - \hat{h})]\}$ . The modeling error  $d = h_i(\theta) - \hat{h}$  is assumed to be a zero-mean homogeneous Gaussian distributed white noise with covariance matrix  $\sum_{dd} = \sigma_d^2 I$  independent of image.  $LB$  is an assumed support size of blur. The PSF learning likelihood is computed based on mahalanobis distance and corresponding model  $l_i(\hat{h}) = \frac{1}{2} \exp[(h_i(\theta^*) - \hat{h})^t \sum_{dd}^{-1}(h_i(\theta^*) - \hat{h})]$ . A best fit model  $h_i(\theta)$  for  $\hat{h}$  is selected according to the Gaussian distribution and a cluster filter. We use a K-NN rule to find the estimated output blur model  $\hat{h}_f$ .  $\hat{h}_f$  is obtained from the parametric blur models using  $\hat{h}_f = [l_0(\hat{h})\hat{h} + \sum_{i=1}^C l_i(\hat{h})h_i(\theta)] / [\sum_{i=1}^C l_i(\hat{h})]$ , where  $l_0(\hat{h}) = 1 - \max(l_i(\hat{h}))$ ,  $i = 1, \dots, C$ . The main objective is to assess the relevance of current estimated blur  $\hat{h}$  with respect to parametric PSF models, and to integrate such knowledge progressively into the computation scheme.

If the current blur  $\hat{h}$  is close to the estimated PSF model  $\hat{h}_f$ , that means  $\hat{h}$  belongs to a predefined parametric model. Otherwise, if  $\hat{h}$  differs from  $\hat{h}_f$  significantly, it means that current blur  $\hat{h}$  may not belong to the predefined priors. The prior space reduces the uncertainty of model selection and largely improve the efficiency for PSF self-initializing in practice.

## 4 Dynamic Edge-Driven Regularization in Image Domain

During the alternate minimization (AM) with respect to the estimates of the PSF and the image, the previous computed PSF is *a priori* knowledge for the next iterative image deconvolution. However, we need diffusion operators to compensate and smooth the “ringing” or “staircase” effects for achieving a restored

image  $\hat{f}$  with more fidelity and high quality visual perception. In the AM, the image energy function is minimized according to the following formulation,

$$\mathcal{F}(\hat{f}|\hat{h}, g) = \frac{1}{2} \int_{\Omega} (g - \hat{h} * \hat{f})^2 dA + \lambda \int_{\Omega} \phi_{\varepsilon}(x, \nabla \hat{f}) dA \quad (2)$$

where the likelihood is  $p(g|\hat{f}, \hat{h}) \propto \exp\{-\frac{1}{2} \int_{\Omega} (g - \hat{h} * \hat{f})^2 dA\}$ . Different from most passively edge-preserving restoration approaches [6,7,11], the smoothing operator  $p(\hat{f}) \propto \exp\{-\int_{\Omega} \phi_{\varepsilon}(x, \nabla \hat{f}) dA\}$  as *a priori* knowledge is extended to a convex nonlinear diffusion functional with variable exponent [13]. The significant advantage of this operator is its robustness with respect noise and actively edge-preserving processes in that the chosen diffusion operators oriented dynamically and continuously. The optimization for the cost function Eq. (2) is numerically solved using its associated Euler-Lagrange equation,  $\lambda \text{div}(\phi_{\varepsilon}(x, \nabla \hat{f})) + (\hat{f} * \hat{h} - g) = 0$ , in  $\Omega \times [0, T]$ . We indicate with *div* the divergence operator, and with  $\nabla$  and  $\Delta$  respectively the gradient and Laplacian operators, with respect to the space variables. The Neumann boundary condition  $\frac{\partial \hat{f}}{\partial n}(x, t) = 0$  on  $\partial \Omega \times [0, T]$  and the initial condition  $\hat{f}(x, 0) = f_0(x) = g$  in  $\Omega$  are used,  $n$  is the direction perpendicular to the boundary. Based on variable exponent, linear growth function [13] and physical simulation [17], the diffusion operator is computed in,  $\text{div}(\phi_{\varepsilon}(x, \nabla \hat{f})) =$

$$\underbrace{|\nabla \hat{f}|^{p(x)-2}}_{\text{Coefficient}} \underbrace{[(p(x) - 1)\Delta \hat{f} + (2 - p(x))|\nabla \hat{f}| \text{div}(\frac{\nabla \hat{f}}{|\nabla \hat{f}})]}_{\text{CurvatureTerm}} + \underbrace{\nabla p \cdot \nabla \hat{f} \log |\nabla \hat{f}|}_{\text{HyperbolicTerm}} \quad (3)$$

where  $p(x) = 1 + \frac{1}{1+k|\nabla G_{\sigma} * I(x)|^2}$ , when  $|\nabla \hat{f}| < \beta$  and  $p(x) = 1$ , otherwise.  $\beta > 0$  is a fixed value.  $I(x)$  is the observed image  $g(x)$ ,  $G_{\sigma}(x) = \frac{1}{\sigma} \exp(\frac{-|x|^2}{2\sigma^2})$  is a Gaussian filter,  $k > 0$ ,  $\sigma > 0$  are fixed parameters. The operator  $\text{div}(\phi_{\varepsilon}(x, \nabla \hat{f}))$  is discretized with a small positive constant  $\varepsilon$  based on *central differences* for the coefficient and isotropic term, *minmod scheme* for the curvature term, and *upwind finite difference scheme* developed by Osher and Sethian for curve evolution [17] for the hyperbolic term which can largely improve the signal-to-noise ratio and human visual perception.

The term  $p(x) \in [1, 2]$  is continuously computed based on the constraints of edge gradients. In homogeneous regions, the differences of intensity between the neighboring pixels are small,  $p(x) \rightarrow 2$ . The isotropic diffusion operator (Laplace) is used in such regions. In non-homogeneous regions (near discontinuities), the anisotropic diffusion filter is chosen continuously based on the gradient values  $1 < p(x) < 2$ . The reason is that the chosen discrete anisotropic operators will hamper the recovery of edges. Simultaneously, the nonlinear diffusion operator for piecewise image smoothing is processed during image deconvolution based on a previously estimated PSF. Finally, coupled estimation of PSFs (blur identification) and images (deblurring + denoising + smoothing) are alternately optimized applying a stopping criteria. Hence, over-smoothing and under-smoothing near discontinuities are avoided on pixel level.

## 5 Segregation of Blurred and Unblurred Regions

Some significant differences (local scale [3,8], sharpness and contrast [2], and piecewise smoothing [14]) between pairwise blurred regions and unblurred regions can be considered as natural prior knowledge for measuring the low-level similarities for segregation using a global graph-clustering criterion.

We set up the vertices of a graph  $\mathcal{G} = (V, E)$  into two sets  $A$  and  $B$  to minimize the number of *cut edges*, i.e., edges with one endpoint in  $A$  and the other in  $B$ , where  $V = \{v_i\}_{i=1}^n$  are the vertices and  $E \subseteq \{(v_i, v_j)\}$  are the edges between these vertices.  $V$  can correspond to pixels in an image or set of connected pixels. The bisection problem can be formulated as the minimization of a quadratic objective function by means of the Laplacian matrix  $L = L(\mathcal{G})$  of the graph  $\mathcal{G}$ . Let  $d(i)$  denote the degree of a vertex  $i$ , i.e., the number of vertices adjacent to  $i$ . The Laplacian matrix  $L$  can be expressed in terms of two matrices associated with a graph as  $L = D - W$  in positive semidefinite [22],  $W = \{w_{ij}\}$  is the adjacency matrix of a graph, and  $D$  is the  $n \times n$  diagonal matrix of the degrees of the vertices in  $\mathcal{G}$ . Let  $x$  be an  $n$ -vector with component  $x_i = 1$  if  $i \in A$  and  $x_i = -1$  if  $i \in B$ , then  $x^T L x = x^T D x - x^T W x = \sum_{i=1}^n d_i x_i^2 - \sum_{(i,j) \in E, i \in A, j \in B} (x_i - x_j)^2$ . Thus the bisection problem is equivalent to the problem of maximizing similarity of the objects within each cluster, or, find a *cut edge* through the graph  $\mathcal{G}$  with minimal weight in the formulation of  $\max(x^T W x) \iff \min(x^T L x)$ .

Since the bisection-partitioning problem is NP-complete, we need to approximate this intractable problem by some relaxing constraints. Likewise, to avoid unnatural bias for partitioning out small sets of points, Shi and Malik [4] proposed a new measure of the disassociation between two groups. Instead of looking at the value of total edge weight connecting the two partitions, the cut cost is computed as a fraction of the total edge connections to all the nodes in the graph. This disassociation measure is called the normalized cut (Ncut):  $Ncut(A, B) = \frac{cut(A, B)}{asso(A, V)} + \frac{cut(B, B)}{asso(B, V)}$ .  $A$  and  $B$  are two initial sets. Different from the weight measurement in [4], we measure the degree of dissimilarity between pairwise blurred and unblurred regions based on the special characteristic properties, i.e., stronger difference of edge gradients, pairwise blur and unblur. The edge weight  $w_{ij}$  between node  $i$  and  $j$  as the product of a feature similarity term and spatial proximity term:  $w_{ij} = \exp\frac{-\|Q(i) - Q(j)\|_2^2}{\sigma_I} * \exp\frac{-\|X(i) - X(j)\|_2^2}{\sigma_X}$ , if  $(\|X(i) - X(j)\|_2) < r$ , and  $w_{ij} = 0$ , otherwise.  $Q(i) = \nabla G_\sigma * I(x)$  is the edge gradients and large differences between pairwise blurred and unblurred regions.  $G_\sigma$  is a Gaussian filter,  $I(x)$  is an input image,  $X(i)$  is the spatial location of node  $i$ . The advantage of the suggested algorithm is its directness and simplicity. The high quality segregation of blurred and unblurred regions or objects in Fig. 2 is guided dynamically by computed prior values without any supervision.

## 6 Experiments and Discussion

**Alternate Minimization (AM) of PSF and Image Energy.** To avoid the scale problem between the minimization of the PSF and image via



**Fig. 2.** *Left:* Identified unblurred foreground walking man from blurred background. *Right:* Identified blurred foreground walking man from unblurred background.

steepest descent, an AM method [12,21] following the idea of coordinate descent is applied. The AM algorithm decreases complexity. The choice of regularization parameters is crucial. We use L-curve [23] due to its robustness with respect to correlated noise. The global convergence of the algorithm to the local minima of cost functions can be established by noting the two steps in the algorithm. Since the convergence with respect to the PSF and the image are optimized alternatively, the flexibility of this proposed algorithm allows us to use conjugate gradient algorithm for computing the convergence. Conjugate gradient methods utilize the conjugate direction instead of local gradient to search for the minima. Therefore, it is faster and also requires less memory when compared with the other methods. A meaningful measure called normalized mean square-error (NMSE) is used to evaluate the performance of the identified blur,  $NMSE = (\sum_x \sum_y (h(x,y) - \hat{h}(x,y))^2)^{1/2} / (\sum_x \sum_y h(x,y))$ , and the restored images are measured by peak signal-to-noise ratio (PSNR) in decibels (dB) as  $PSNR = 10 \log_{10}(255^2/MSE)$  with  $MSE = |\Omega|^{-1} \sum_{x_i \in \Omega} [f(x_i) - \hat{f}(x_i)]^2$ , where  $f$  is the noise-free ideal image and  $\hat{f}$  is the restored image.

### **Denosing and Blind Deconvolution for Noisy and Blurred Images.**

Firstly, we have studied the importance of diffusion in the regularization based image deconvolution, shown in Fig. 3. The second experiments demonstrate the efficiency of the suggested edge-driven diffusion method. From visual perception and denoising viewpoint, our unsupervised edge-driven method favorably compares to some state-of-the-art methods: the TV [9], a statistic-wavelet method (GSM) [24] and a Markov random field based filter learning method (FoE) [20] using a PIII 1.8GHz PC. In Fig. 4, the structure of the restored fingerprint is largely enhanced than the original image in our method and more recognizable than the restored image using the GSM method [24]. Fig. 5 shows the advantage of our method, while the TV method [9] has some piecewise constant effects during the denoising. Table 1 shows the different properties of different methods and also shows our method outperforms most of these methods. To achieve similar results, FoE [20] needs more time. Our method (100 iter.) is faster than the TV method (30 iter.) in that our method does not over-smooth and generate redundant image discontinuities. The GSM [24] method is relatively faster due to the computation in the Fourier domain. However, the GSM is only designed for denoising. The dual-purpose edge-driven method is not only for denoising but also for compensating the “ringing” and “staircase” effects and for protecting the image structure and textures during the image deconvolution.



**Fig. 3.** Deconvolution with known PSF without using diffusion operator. *Left:* Blurred image. *Middle:* Deconvolution of Gaussian blur. *Right:* Deconvolution of motion blur.



**Fig. 4.** Fingerprint denoising. *Left:* Cropped noisy image,  $SNR = 8$  dB. *Middle:* GSM method[24]  $PSNR=27.8$ . dB *Right:* The suggested method  $PSNR= 28.6$  dB.

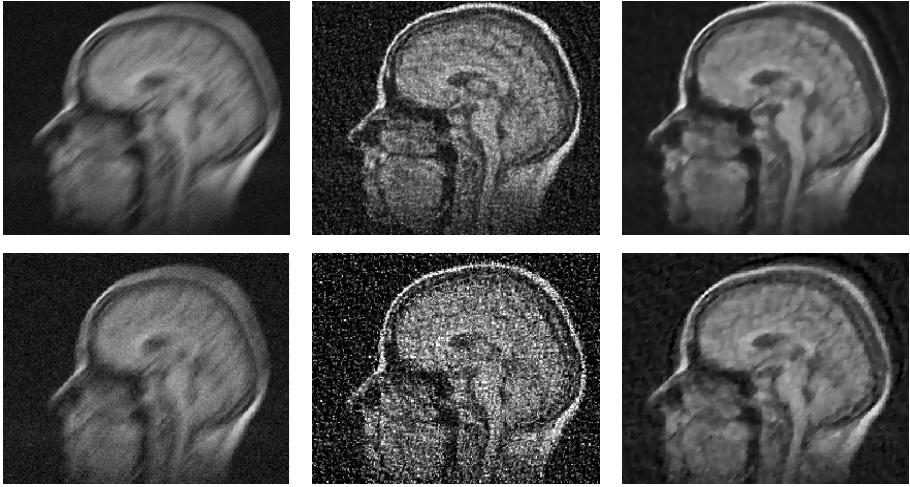


**Fig. 5.** Denoising. *Left:* Unblurred noisy image,  $SNR=8$ dB, size: [256, 256]. *Middle:* TV method,  $PSNR = 27.1$  dB. *Right:* Edge-driven diffusion,  $PSNR = 30.2$  dB.

**Table 1.** Denoising performance of different methods on PSNR (dB)

PSNR	$\sigma = 17.5, SNR \approx 8.7$ dB, size [512,512]						Iter(n)	Time(s)
(dB)	Lena	Barbara	Boats	House	Pepper	fingerprint	Number	Second
Our Met.	32.26	31.25	31.01	31.85	30.61	28.81	100	600 ~ 650
TV.[9]	31.28	26.33	29.42	31.33	24.57	27.29	30	800 ~ 820
FoE[20]	32.11	27.65	30.26	32.51	30.42	26.41	$1 \sim 3 \times 10^3$	$3 \sim 9 \times 10^3$
GSM[24]	32.72	30.12	30.58	32.69	30.78	28.59	100	140 ~ 180





**Fig. 6.** Deconvolution and denoising. *Left:* From top to bottom:  $SNR = 20\text{dB}$  and  $12\text{dB}$ , size:  $[256, 256]$ . *Middle:* L-R method with known PSF. *Right:* The suggested method with unknown PSF.

For blind deconvolution, we compare the classical Lucy-Richardson (L-R) deconvolution method with known PSF to the suggested method with unknown PSF. A MRI image is heavily blurred with two level of noise 20 dB and 12 dB, shown in the first column of Fig. 6. The noise is amplified during the L-R deconvolution with known PSF, shown in the middle column. In the suggested method, the self-initialized PSF is iteratively parametric optimized in the AM algorithm. Diffusion operators vary with the coefficient  $p(x)$  in the interval  $[1, 2]$  continuously. The estimated PSF supports the image smoothing coefficients progressively till the best recovered image is reached, shown in the right column. From the restored images, we can observe that the low frequency regions are more smooth while the fine details of discontinuities (high frequency regions) are preserved during the image deconvolution. The experiment demonstrates the flexibility of Bayesian based double regularization method which can accurately identify the blur and restore images using edge-driven nonlinear image operators. The results also show that the denoising and deblurring can be achieved simultaneously even under the presence of stronger noise and blur.

Identification and segregation of partially-blurred, noisy images or video sequences have good performance using the suggested method, shown in Fig. 1 and Fig. 2. Cluttering blurred or unblurred background does not influence the segmentation and identification of unblurred or blurred objects. These object boundaries with different computing weights are grouped into different groups via the extended global cluster criterion with blur and edge priors. The segmentation results are labeled and color filled following the partitioned regions. The experimental results show that the method yields encouraging results under different kinds and amounts of noise and blur. We applied our method to all

the related images and video data, but due to space limitations we refer the interested reader to our web-page [www.cv.tu-berlin.de/~hzheng/dagm06.html](http://www.cv.tu-berlin.de/~hzheng/dagm06.html).

## 7 Conclusions

This paper validates the hypothesis that the challenging task of nonlinear diffusion and BID are tightly coupled in a variational regularized Bayesian estimation. Firstly, it provides a statistic self-initializing value in regularization for blur identification. Secondly, it shows a theoretically and experimentally sound way of how local diffusion operators are changed dynamically via *a priori* knowledge of edge gradients. The estimated PSF is the prior knowledge for the next iteration of image estimation in the alternating minimization, and vice versa. Finally, a graph-theoretical approach is extended to segregate and identify blurred and unblurred regions or objects. The integrated approach also demonstrates that the mutual supports between natural prior knowledge and low-level image processing have great potential role to improve the results in early vision.

## References

1. Tikhonov, A., Arsenin, V.: Solution of Ill-Posed Problems. Wiley, Winston (1977)
2. Luxen, M., Förstner, W.: Characterizing image quality: Blind estimation of the point spread function from a single image. In: PCV02. (2002) 205–211
3. Elder, J.H., Zucker, S.W.: Local scale control for edge detection and blur estimation. IEEE Trans. on PAMI **20** (1998) 699–716
4. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. on PAMI **8** (2000) 888–905
5. Keuchel, J., Schnörr, C., Schellewald, C., Cremers, D.: Binary partitioning, perceptual grouping, and restoration with semidefinite programming. IEEE Trans. on PAMI **25** (2003) 1364–1379
6. Geman, S., Reynolds, G.: Constrained restoration and the recovery of discontinuities. IEEE Trans. on PAMI **14** (1995) 932–946
7. Charbonnier, P., Blanc-Feraud, L., Aubert, G., Barlaud, M.: Deterministic edge-preserving regularization in computed imaging. IEEE Tr. I.P. **6** (1997) 298–311
8. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. IEEE Trans. on PAMI **12** (1990) 629–639
9. Rudin, L., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithm. Physica D **60** (1992) 259–268
10. Weickert, J.: Coherence-enhancing diffusion filtering. IJCV **31** (1999) 111–127
11. Romeny, B.M.: Geometry-Driven Diffusion in Computer Vision. Kluwer Academic Publishers (1994)
12. Bar, L., Sochen, N., Kiryati, N.: Variational pairing of image segmentation and blind restoration. In Pajdla, T., ed.: ECCV 2004. (LNCS 3022) 166–177
13. Chen, Y., Levine, S., Rao, M.: Variable exponent, linear growth functionals in image restoration. SIAM Journal of Applied Mathematics **66** (2006) 1383–1406
14. Mumford, D., Shah, J.: Optimal approximations by piecewise smooth functions and associated variational problems. Communications on Pure and Applied Mathematics **42** (1989) 577–684

15. Molina, R., Katsaggelos, A., Mateos, J.: Bayesian and regularization methods for hyperparameters estimate in image restoration. *IEEE on S.P.* **8** (1999) 231–246
16. Bishop, C.M., Tipping, M.E.: Bayesian regression and classification. In: *Advances in Learning Theory: Methods, Models and Applications*. (2003) 267–285
17. Osher, S., Sethian, J.A.: Front propagation with curvature dependent speed: algorithms based on Hamilton-Jacobi formulations. *J. Comp. Phy.* **79** (1988) 12–49
18. Geman, S., Geman, D.: Stochastic relaxation, Gibbs distribution and the Bayesian restoration of images. *IEEE Trans. on PAMI* **6** (1984) 721–741
19. Zhu, S., Mumford, D.: Prior learning and Gibbs reaction-diffusion. *IEEE Trans. on PAMI* **19** (1997) 1236–1249
20. Roth, S., Black, M.: Fields of experts: A framework for learning image priors. In: *CVPR*, San Diego (2005) 860–867
21. Zheng, H., Hellwich, O.: Double regularized Bayesian estimation for blur identification in video sequences. In Nara., P., ed.: *ACCV 2006*. (LNCS 3852) 943–952
22. Pothén, A., Simon, H., Liou, K.: Partitioning sparse matrices with eigenvectors of graphs. *SIAM. J. Matrix Anal. App.* **11** (1990) 435–452
23. Hansen, P., O’Leary, D.: The use of the L-curve in the regularization of discrete ill-posed problems. *SIAM J. Sci. Comput.* **14** (1993) 1487–1503
24. Portilla, J., Strela, V., Wainwright, M., Simoncelli, E.: Image denoising using scale mixtures of Gaussians in the wavelet domain. *IEEE Trans. on Image Processing* **12** (2003) 1338–1351

# On-Line, Incremental Learning of a Robust Active Shape Model

Michael Fussenegger<sup>1</sup>, Peter M. Roth<sup>2</sup>, Horst Bischof<sup>2</sup>, and Axel Pinz<sup>1</sup>

<sup>1</sup> Institute of Electrical Measurement and Measurement Signal Processing  
Graz University of Technology

Kopernikusgasse 24/IV, 8010 Graz, Austria

{fussenegger, axel.pinz}@tugraz.at

<sup>2</sup> Institute for Computer Graphics and Vision

Graz University of Technology

Inffeldgasse 16/II, 8010 Graz, Austria

{pmroth, bischof}@icg.tu-graz.ac.at

**Abstract.** Active Shape Models are commonly used to recognize and locate different aspects of known rigid objects. However, they require an off-line learning stage, such that the extension of an existing model requires a complete new re-training phase. Furthermore, learning is based on principal component analysis and requires perfect training data that is not corrupted by partial occlusions or imperfect segmentation. The contribution of this paper is twofold: First, we present a novel robust Active Shape Model that can handle corrupted shape data. Second, this model can be created on-line through the use of a robust incremental PCA algorithm. Thus, an already partially learned Active Shape Model can be used for segmentation of a new image in a level set framework and the result of this segmentation process can be used for an on-line update of the robust model. Our experimental results demonstrate the robustness and the flexibility of this new model, which is at the same time computationally much more efficient than previous ASMs using batch or iterated batch PCA.

## 1 Introduction

Prior knowledge of the object contour/shape is used to improve the result in many computer vision approaches dealing with segmentation, object detection or tracking. A common approach, to model different aspects of rigid objects in a shape prior formalism, is the use of Active Shape Models (ASMs) proposed by Cootes et al. [4, 5]. The standard ASM framework consists of two stages: (1) the modeling/learning and (2) the segmentation/detection stage.

In this paper we use a level set segmentation framework. Level set representation [14] is an established technique for image segmentation. Over the years several different level set models of ASMs have been presented (e.g., [16, 6]). In particular, we use the level set representation of Rousson and Paragios [16]. To avoid unnecessary computation and numerical errors, we work with the level set shape representation and avoid a conversion to the classical landmark representation used in [5].

In the learning stage, a set of registered training shapes is used to model different aspects of a model. Generally ASM approaches use a batch (off-line) version of Principal Component Analysis (PCA) [11] for learning, that has two main disadvantages: (1) the approach is not robust in the recognition nor in the training [12, 18] (but, we might receive data that is corrupted by partial occlusions or imperfect segmentation) and (2) all training data has to be provided a priori, thus, hand segmentation is required and the shape model cannot be extended as new segmentation results become available.

Various approaches have been proposed to introduce robustness in the recognition stage (e.g., [15, 1, 12]). For these methods it is assumed that the samples in the learning stage are undisturbed. Robust learning is a more difficult problem, since there is no previous knowledge that can be used to estimate outliers. Several methods have been proposed to robustly extract the principal axes in the presence of outliers [7, 21]. Other approaches use robust M-estimator [7] or are based on the EM formulation of PCA [17, 20, 18]. Using a robust approach in our framework has two advantages: (1) the robust reconstruction from the ASM allows a much better segmentation of occluded objects and (2) robust learning on the improved segmentation results provides a better shape representation.

We use an incremental PCA approach in our ASM. Applying an incremental method, we can efficiently build and update an ASM that is used for the segmentation process, i.e., we can use the partially learned ASM to perform segmentation and use the segmentation result to retrain the ASM. Different incremental PCA approaches have been proposed that are based on incremental SVD-updating (e.g., [9, 2]). Recently even robust and incremental [13, 19] approaches have been proposed. In particular we apply a simplified version of the approach of Skočaj and Leonardis [19] to learn the ASM that will be explained in Section 2.2.

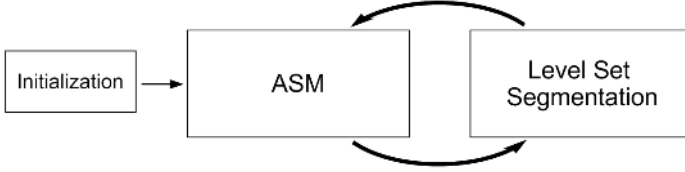
Applying this incremental and robust PCA method, we need a priori only a small hand segmented data set to initialize our ASM. This first model provides shape priors for the segmentation process. Furthermore, the Active Shape Model can be successively updated with new data from the segmentation process.

The outline of the paper is as follows: Section 2 explains our system and gives a short description of its components. In Section 2.1, we describe the shape registration. In Section 2.2, we introduce in detail the Robust Incremental PCA. Experiments are presented in Section 3 and finally, conclusions are drawn in Section 4.

## 2 Incremental Robust Active Shape Model

Fig. 1 depicts our proposed method, which is split into two components: (i) the segmentation module and (ii) our novel ASM module. For the segmentation, we use the approach proposed by Fussenegger et al. [8] which is based on the level set formulations by Rousson and Paragios [16] and Brox and Weickert [3].

The output of the segmentation module is the distance function  $\Phi(\mathbf{x})$ ,  $\Phi : \Omega \rightarrow \mathbb{R}$ , with  $\Phi(\mathbf{x}) > 0$ , if  $\mathbf{x}$  lies in the shape and  $\Phi(\mathbf{x}) < 0$ , if  $\mathbf{x}$  lies out of the shape.  $\Omega$  is the image domain and  $\mathbf{x}$  denotes a pixel in  $\Omega$ . In order to avoid unnecessary computation and numerical errors, we use directly this distance function as the shape representation



**Fig. 1.** Our System consisting of two interacting components: The level set segmentation [8] and our novel Active Shape Model

instead of the landmark representation used in [4, 5]. However, our ASM can be easily adapted to alternative shape representations for use with other segmentation approaches.

In a first step, the ASM module is initialized with a training set of non corrupted, aligned shapes characterizing different aspects of an object to learn a first Active Shape Model. This learned ASM is then used in the segmentation process. After each level set iteration step the current result  $\Phi_i$  is passed from the segmentation module to the ASM module. A registration process (Section 2.1) applies a similarity transformation  $\mathcal{A}$  on  $\Phi_i$  to map it with  $\Phi_M$  in the best way.  $\Phi_M$  is the mean shape calculated over the already learned shapes.  $\Phi_i$  is then projected to the eigenspace and robustly reconstructed (Section 2.2). The reconstruction  $\tilde{\Phi}_i$  is passed to the segmentation module and used as a shape prior in the next level set iteration step. This is repeated until the segmentation process ends. The final result is used to update and improve our ASM.

## 2.1 Shape Registration

For the shape registration, we assume a global deformation  $\mathcal{A}$  between  $\Phi_M$  (the mean shape) and  $\Phi$  (the new shape) that involves the parameters  $[\mathcal{A} = (s; \theta; \mathbf{T})]$  with a scale factor  $s$ , a rotation angle  $\theta$  and a translation vector  $\mathbf{T}$  [16]. The objective function

$$E(\Phi_M, \Phi(\mathcal{A})) = \int_{\Omega} (s\Phi_M - \Phi(\mathcal{A}))^2 dx \quad (1)$$

can be used to recover the optimal registration parameters. The rigid transformation  $\mathcal{A}$  is dynamically updated to map  $\Phi_M$  and  $\Phi$  in the best way. Thus, the calculus of variations for the parameters of  $\mathcal{A}$  yields the system

$$\begin{aligned} \frac{\partial s}{\partial t} &= 2 \int_{\Omega} (s\Phi_M - \Phi(\mathcal{A}))(\Phi_M - \nabla\Phi(\mathcal{A})) \frac{\partial}{\partial s} \mathcal{A} dx \\ \frac{\partial \theta}{\partial t} &= 2 \int_{\Omega} (s\Phi_M - \Phi(\mathcal{A}))(-\nabla\Phi(\mathcal{A})) \frac{\partial}{\partial \theta} \mathcal{A} dx \\ \frac{\partial \mathbf{T}}{\partial t} &= 2 \int_{\Omega} (s\Phi_M - \Phi(\mathcal{A}))(-\nabla\Phi(\mathcal{A})) \frac{\partial}{\partial \mathbf{T}} \mathcal{A} dx. \end{aligned} \quad (2)$$

Fig. 2(a-c) shows three example shapes. Fig. 2(d) and 2(e) show all three shape contours before and after the registration process.

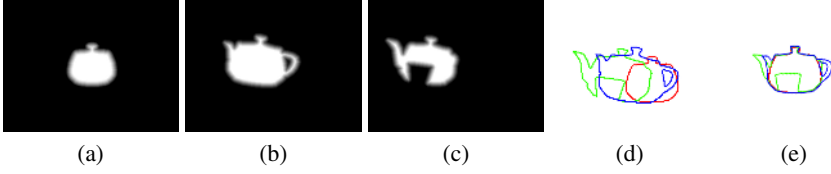


Fig. 2. Three example shapes before and after registration

## 2.2 Robust Incremental PCA

For batch PCA all training images are processed simultaneously. A fixed set of input images  $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n] \in \mathbb{R}^{m \times n}$  is given, where  $\mathbf{x}_i \in \mathbb{R}^m$  is an individual image represented as a vector. It is assumed that  $\mathbf{X}$  is mean normalized. Let  $\mathbf{Q} \in \mathbb{R}^{m \times m}$  be the covariance matrix of  $\mathbf{X}$ , then the subspace  $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_n] \in \mathbb{R}^{m \times n}$  can be computed by solving the eigenproblem for  $\mathbf{Q}$  or more efficiently by solving SVD of  $\mathbf{X}$ .

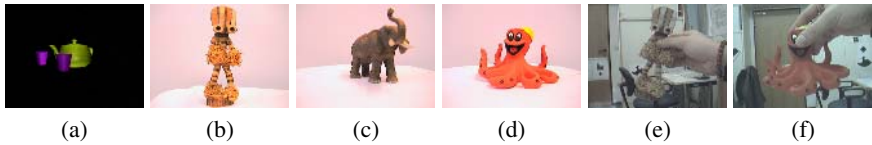
For incremental learning, the training images are given sequentially. Assuming that an eigenspace was already built from  $n$  images, at step  $n + 1$  the current eigenspace can be updated in the following way [19]: First, the new image  $\mathbf{x}$  is projected in the current eigenspace  $\mathbf{U}^{(n)}$  and the image is reconstructed:  $\tilde{\mathbf{x}}$ . The residual vector  $\mathbf{r} = \mathbf{x} - \tilde{\mathbf{x}}$  is orthogonal to the current basis  $\mathbf{U}^{(n)}$ . Thus, a new basis  $\mathbf{U}'$  is obtained by enlarging  $\mathbf{U}^{(n)}$  with  $\mathbf{r}$ .  $\mathbf{U}'$  represents the current images as well as the new sample. Next, batch PCA is performed on the corresponding low-dimensional space  $\mathbf{A}'$  and the eigenvectors  $\mathbf{U}''$ , the eigenvalues  $\lambda''$  and the mean  $\mu''$  are obtained. To update the subspace the coefficients are projected in the new basis  $\mathbf{A}^{(n+1)} = \mathbf{U}'^T (\mathbf{A}' - \mu'' \mathbf{1})$  and the subspace is rotated:  $\mathbf{U}^{(n+1)} = \mathbf{U}' \mathbf{U}''$ . Finally, the mean  $\mu^{(n+1)} = \mu^{(n)} + \mathbf{U}' \mu''$  and the eigenvalues  $\lambda^{(n+1)} = \lambda''$  are updated. In each step the dimension of the subspace is increased by one. To preserve the dimension of the subspace the least significant principal vector may be discarded [10]. To obtain an initial model, the batch method may be applied to a smaller set of training images. Alternatively, to have a fully incremental algorithm, the eigenspace may be initialized using the first training image  $\mathbf{x}$ :  $\mu^{(1)} = \mathbf{x}$ ,  $\mathbf{U}^{(1)} = \mathbf{0}$  and  $\mathbf{A}^{(1)} = \mathbf{0}$ .

This method can be extended in a robust manner, i.e., corrupted input images may be used for incrementally updating the current model. To achieve this, outliers in the current image are detected and replaced by more confident values: First, an image is projected to the current eigenspace using the robust approach [12] and the image is reconstructed. Second, outliers are detected by pixel-wise thresholding (based on the expected reconstruction error) the original image and its robust reconstruction. Finally, the outlying pixel values are replaced by the robustly reconstructed values.

## 3 Experiments

For the experiments, we have created several different data sets: *teapot*, *African man*, *elephant* and *octopus* (Fig. 3). The first one (Fig. 3(a)) was created artificially by using 3D-MAX. The others are representing real world objects where the images were

acquired in two different ways: a smaller number of images was obtained using a turntable and a homogeneous background, such that we can use the level set segmentation without any shape-prior (Fig. 3(b)-(d)). These views are used to build the initial consistent model. Depending on the complexity of the object, i.e., the complexity of the object’s shape, 10 up to 80 views are needed. The consistent model is necessary for robust reconstruction of outlying values in the input shapes resulting from over-segmentation and under-segmentation in further steps. Additionally, more complex images (hand held presentations of the objects with cluttered background) are acquired and used to demonstrate the incremental update and robustness of the method (Fig. 3(e)-(f)).



**Fig. 3.** Examples of our data sets: *teapot*, *African man*, *elephant* and *octopus*

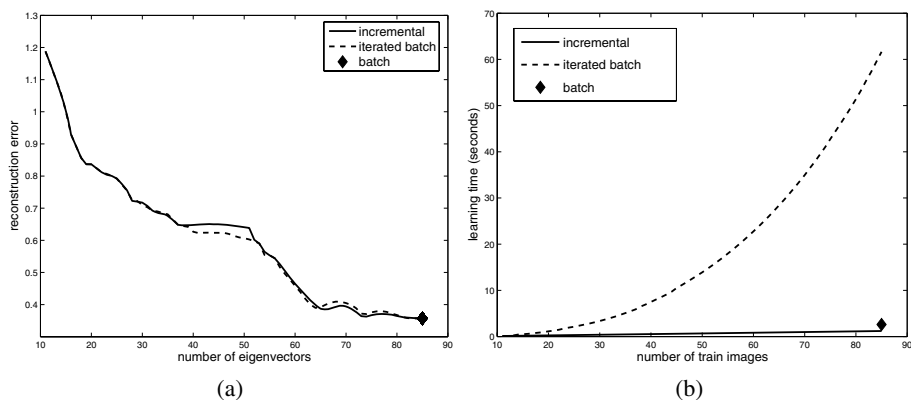
To show the benefit of the incremental method we trained classifiers using different PCA approaches on the *teapot* data set. In total 85 images were processed in the training stage where only 10 eigenvectors were used for reconstruction. The obtained classifiers were evaluated on an independent test set of 10 images.

First, Fig. 4(a) shows that the incremental on-line method yields similar results as the “iterated batch” method (batch PCA is applied when a new image arises) that is applied in most applications. The reconstruction errors of the incremental PCA, the “iterated batch” PCA and the batch PCA are compared for an increasing number of training shapes and eigenvectors. The reconstruction error is similar for both, the incremental and the iterated batch method. Both error curves are continuously decreasing when the number of training shapes is increased and they are approaching the results of the batch method (trained from the full set of training shapes). Thus, there is no real loss of accuracy (for our application) in using the incremental approach. But as can be seen in Fig. 4(b) there are huge differences in the computational costs for the different methods; the learning times were obtained by evaluating the training in MATLAB on a 3GHz machine. The results for the whole training set containing 85 images are summarized in Table 1. Since the matrix operations are performed on smaller matrices only (less memory has to be allocated) for this data set the incremental method is even computationally cheaper than the batch method. But, as the main point, compared to the iterated batch “incremental” approach the computational costs of the incremental method are only approximately 1/40!

**Table 1.** Performance evaluation

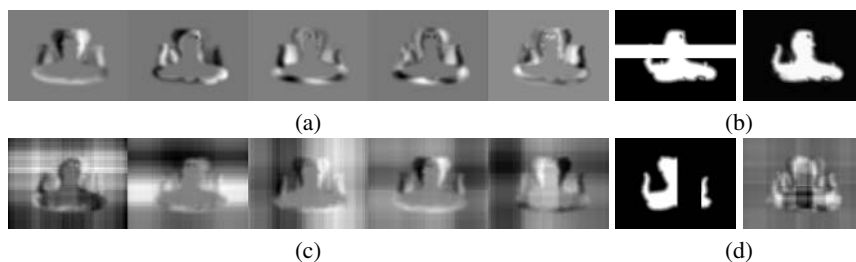
method	incremental PCA	batch PCA	iterated batch PCA
time	4.72s	6.55s	205.38s





**Fig. 4.** Incremental PCA approach evaluated on the *teapot* data set: (a) incremental PCA performs similar as incremental batch PCA, (b) incremental PCA is computationally much cheaper than incremental batch PCA

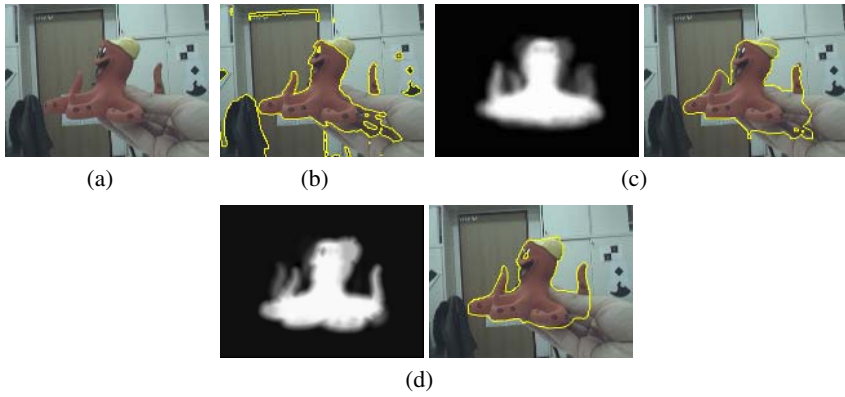
As the applied incremental PCA method can easily be extended in a robust manner we want to show the advantages of robust incremental learning. For this purpose an initial classifier for the octopus is trained using only 15 clean shapes to get a consistent starting model. Later on, the training is continued from corrupted data. To simulate over-segmented and under-segmented shapes the corrupted data is created by randomly adding black and white bars occluding 25% of the image. By adding these shapes the non-robust model gets more and more corrupted (see Fig. 5(c) for the first 5 eigenimages) while a stable model is estimated by using the robust approach (see Fig. 5(a) for the first 5 eigenimages). Examples of reconstructions are shown in Fig. 5(b) (robust eigenspace) and Fig. 5(d) (non-robust eigenspace).



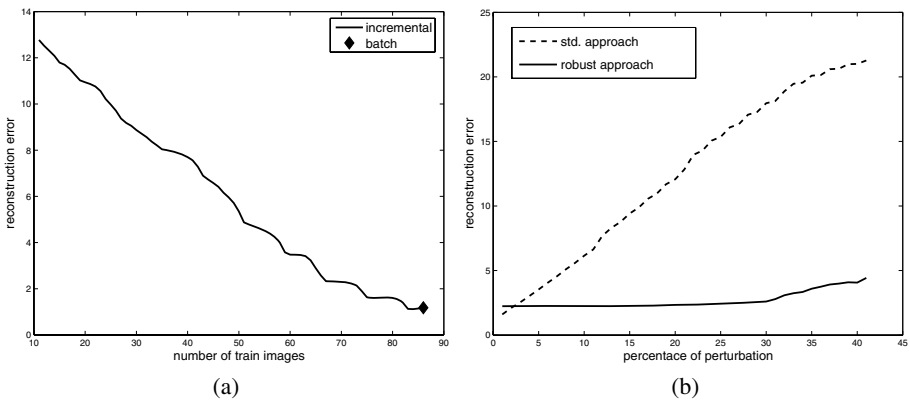
**Fig. 5.** Robust incremental vs. plain incremental approach: (a) eigenspace obtained by robust incremental learning from noisy data, (b) reconstruction from robust eigenspace, (c) eigenspace obtained by incremental learning from noisy data, (d) reconstruction from non-robust eigenspace

To show the increasingly better segmentation results when incrementally updating the current model with newly obtained shapes, the more complex *octopus* data set was evaluated. In total 85 training shapes were processed where 35 eigenvectors were used for reconstruction. Fig. 6 shows different level set segmentation results using our ASM

in different training stages. For Fig. 6(b), the segmentation is done without a trained ASM. In this case the segmentation fails completely. In Fig. 6(c), we show the final shape prior provided from the initialized ASM (40 “off-line” training shapes) and the corresponding segmentation. The segmentation has been improved significantly but still some errors are present. Afterwards, our ASM is incrementally updated with new “on-line” training shapes. Fig. 6(d) shows the results after 40 additional incrementally obtained shapes. The segmentation is perfect and the segmentation result depicted in Fig. 6 can then be used to add a new aspect to our ASM.

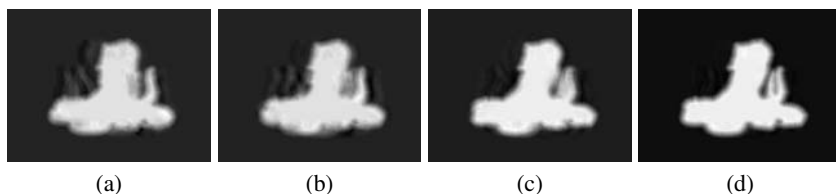


**Fig. 6.** Original image (a) and level set segmentation without an ASM (b). Estimated shape prior, with an ASM learned from 40 training shapes and corresponding level set segmentation (c). Estimated shape prior, with an ASM learned from 80 training shapes and corresponding level set segmentation (d).

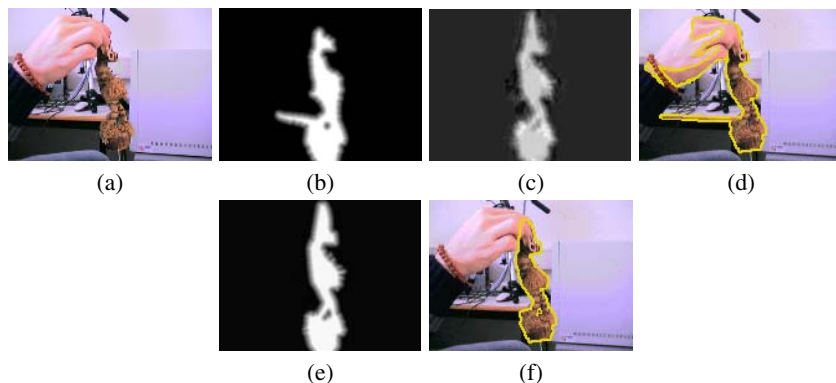


**Fig. 7.** Better segmentation results by robust incremental ASM: (a) increasing the size of the model decreases the reconstruction error, (b) smaller reconstruction error when applying the robust method even for corrupted data

For a more general evaluation different models (varying the number of processed training shapes) were evaluated on a set of 10 independent test images. For this purpose the shapes in the test set were reconstructed using the previously trained models and the reconstruction errors were analyzed. According to the number of processed shapes (10 to 85) up to 35 eigenvectors were used for reconstruction. The results are shown in Fig. 7(a). It can be seen that the reconstruction error is decreasing if the number of learned shapes (and thus the number of eigenimages used for reconstruction) is increased; a better model is obtained! Examples of continuously improving ASMs are shown in Fig. 8.



**Fig. 8.** Improving ASM by updating with new shapes: (a) ASM learned from 10 frames, (b) ASM learned from 20 frames, (c) ASM learned from 40 frames, (d) ASM learned from 80 frames



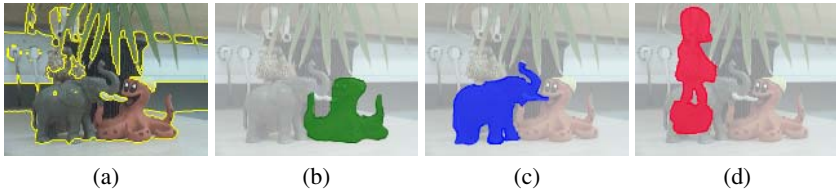
**Fig. 9.** Robust PCA on *African man* data set: (a) original data, (b) segmentation and registration, (c) reconstruction, (d) segmentation result with (c) as shape prior, (e) robust reconstruction, (f) segmentation result using (e) as shape prior

Furthermore, the robust extension of the approach was evaluated on the *African man* data set. The reconstruction error<sup>1</sup> was analyzed when the portion of noise is increased. The results are shown in Fig. 7(b). While the reconstruction error is continuously growing for the standard approach the performance of the robust method is not decreased

<sup>1</sup> The reconstruction error was computed from the undisturbed original images. Alternatively, the distance between the learned eigenspace and the projected shape may be used as measurement (which will yield similar curves).

even in cases of up to 25% occlusion. Thus, these robustly obtained reconstructions can be used to update the current ASM as it can be seen in Fig. 9. The object was presented in a realistic scenario with background clutter by hand from varying views and under different illumination conditions. The registered segmentation without using a shape prior in Fig. 9(b) contains holes and over-segmentations. Thus, the standard reconstruction depicted in Fig. 9(c) is corrupted. But as shown in Fig. 9(e) the robust approach provides a perfect reconstruction that can be used to update the ASM. In addition, Fig. 9(d) shows the corrupted segmentation result obtained by using the standard reconstruction as shape prior while the segmentation result obtained by using the robust reconstruction shown in Fig. 9(f) is perfect.

Finally, we show some examples of segmentations using previously trained Active Shapes Models. In Fig. 10(a), we use the level set segmentation based on [8] without any shape information. The other three Figures 10(b)-(d) are segmented with three different ASMs. On all three objects, we achieve excellent segmentation results, even for Fig. 10(d), where the lower part of the object is highly occluded, our robust ASM is able to segment the object correctly.



**Fig. 10.** Level set segmentation results based on [16]: (a) segmentation without a shape prior; (b)-(d) segmentation using different ASMs

## 4 Conclusion

We have introduced a novel robust Active Shape Model, that can be updated on-line. Using a robust, incremental PCA allows a successive update of our ASMs even with non perfect data (i.e., corrupted data by partial occlusions or imperfect segmentation). For the segmentation and shape representation, we use the work of Rousson et al. [16] but our ASM can easily be adapted to other segmentation approaches and shape representations. We performed experiments on various data sets with different objects. The advantages of the robust, incremental PCA over the standard batch PCA were shown, and we also showed excellent results using different ASMs for segmentation. Even highly occluded objects in a cluttered background can be segmented correctly. Compared to the standard approach the computational costs can be dramatically reduced. Moreover the user interaction is reduced to taking a smaller number of images on a turn table under perfect conditions, i.e., manual segmentation can be completely avoided!

## Acknowledgment

This work has been supported by the Austrian Joint Research Project Cognitive Vision under projects S9103-N04 and S9104-N04.

## References

1. M. J. Black and A. D. Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. In *Proc. European Conf. on Computer Vision*, pages 329–342, 1996.
2. M. Brand. Incremental singular value decomposition of uncertain data with missing values. In *Proc. European Conf. on Computer Vision*, volume I, pages 707–720, 2002.
3. T. Brox and J. Weickert. Level set based image segmentation with multiple regions. In *Proc. DAGM Symposium*, pages 415–423, 2004.
4. T. F. Cootes, D. H. Cooper, C. J. Taylor, and J. Graham. A trainable method of parametric shape description. pages 289–294, 1992.
5. T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Gaham. Active shape models - their training and application. 61(1):38–59, 1995.
6. D. Cremers, N. Sochen, and C. Schnoerr. Towards recognition-based variational segmentation using shape priors and dynamic labeling. In *Proc. of Scale-Space*, pages 388–400, 2003.
7. F. de la Torre and M. J. Black. Robust principal component analysis for computer vision. In *Proc. IEEE Intern. Conf. on Computer Vision*, volume I, pages 362–369, 2001.
8. M. Fussenegger, R. Deriche, and A. Pinz. A multiphase level set based segmentation framework with pose invariant shape priors. In *Proc. of the Asian Conference on Computer Vision*, pages 395 – 404, 2006.
9. P. Hall, D. Marshall, and R. Martin. Incremental eigenanalysis for classification. In *Proc. British Machine Vision Conf.*, volume I, pages 286–295, 1998.
10. P. Hall, D. Marshall, and R. Martin. Merging and splitting eigenspace models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(9):1042–1049, 2000.
11. H. Hotelling. Analysis of a complex of statistical variables with principal components. *Journal of Educational Psychology*, 24:417–441, 1933.
12. A. Leonardis and H. Bischof. Robust recognition using eigenimages. *Computer Vision and Image Understanding*, 78(1):99–118, 2000.
13. Y. Li. On incremental and robust subspace learning. *Pattern Recognition*, 37(7):1509–1518, 2004.
14. S. J. Osher and J. A. Sethian. Fronts propagation with curvature depend speed: Algorithms based on Hamilton-Jacobi formulations. In *Journal of Comp. Phys.*, volume 79, pages 12–49, 1988.
15. R. Rao. Dynamic appearance-based recognition. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 540–546, 1997.
16. M. Rousson and N. Paragios. Shape priors for level set representations. In *Proc. of European Conf. of Computer Vision*, volume 2351 of LNCS, pages 78–92, 2002.
17. S. Roweis. EM algorithms for PCA and SPCA. In *Proc. Conf. on Neural Information Processing Systems*, pages 626–632, 1997.
18. D. Skočaj, H. Bischof, and A. Leonardis. A robust PCA algorithm for building representations from panoramic images. In *Proc. European Conf. on Computer Vision*, volume IV, pages 761–775, 2002.
19. D. Skočaj and A. Leonardis. Weighted and robust incremental method for subspace learning. In *Proc. IEEE Intern. Conf. on Computer Vision*, volume II, pages 1494–1501, 2003.
20. M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society B*, 61:611–622, 1999.
21. L. Xu and A. L. Yuille. Robust principal component analysis by self-organizing rules based on statistical physics approach. *IEEE Trans. on Neural Networks*, 6(1):131–143, 1995.

# Using Irreducible Group Representations for Invariant 3D Shape Description

Marco Reisert and Hans Burkhardt

University of Freiburg, Computer Science Department,  
79110 Freiburg i.Br., Germany  
`reisert@informatik.uni-freiburg.de`

**Abstract.** Invariant feature representations for 3D objects are one of the basic needs in 3D object retrieval and classification. One tool to obtain rotation invariance are Spherical Harmonics, which are an orthogonal basis for the functions defined on the 2-sphere. We show that the irreducible representations of the 3D rotation group, which acts on the Spherical Harmonic representation, can give more information about the considered object than the Spherical Harmonic expansion itself. We embed our new feature extraction methods in the group integration framework and show experiments for 3D-surface data (Princeton Shape Benchmark).

## 1 Introduction

In many fields researchers deal with a huge amount of three dimensional data. In medical and biological applications one usually has to do with volumetric scans of various types. There is a need for fast and reliable feature extraction methods to handle and classify such huge amounts of data. The development of 3D modeling software has increased the number of freely available 3D-surface models, fast retrieval systems are necessary to browse and search for 3D-models in a user-friendly way. As the representation of 3D objects is not canonical and objects often occur at different spatial position and in different rotational poses, the question arises how to compare and classify the objects. One way is to use invariant features.

There are basically two ways to obtain invariance: Group integration and Normalization techniques. Normalization techniques obtain invariance by computing features relative to a global reference frame. The determination of the reference frame makes Normalization techniques extremely sensitive to noise. Whereas Group Integration (GI) is known to be very robust to many kinds of noise. In [1] a detailed overview over GI-techniques is given.

In this work we want to concentrate on 3D surface models. The Princeton Shape Benchmark (PSB) [9] offers a possibility to evaluate 3D-feature extraction techniques. It consists of approx. 1800 surface models collected from the Web. There is already a huge amount of work concerning feature extraction for 3D surface models by the use of Spherical Harmonics (SH). Vranic et al. [10] compute a so-called spherical extent function of the model-surface and make a spherical

harmonic transform of this function, but the rotational invariance is obtained by normalization. Kazhdan et al. [4] eliminate the rotational dependency by taking the magnitude of the invariant subspaces of the Spherical Harmonic transform. They show, that in most cases this is the better alternative than a normalizing approach. In [8] Reisert et al. enhanced the Shape Distribution introduced by Osada [6] by SH-expansion. The currently best performing methods on the PSB are the so-called Light Field Descriptors (LFD) [2]. LFD is an appearance based methods. The shape is rendered from several views and features for each view are computed. Two models are compared by searching the best matching pair of views.

The work is organized as follows: in Section 2 we introduce the basic algebra concerning rotations in 3D and introduce the so called *D-Wigner* matrices, the irreducible representation of the 3D rotation group. Further we give a relation for fast computation. In Section 3 we shortly review the group integration framework and show how the D-Wigner matrices can be used to enhance invariant features and keep more discriminative power. Then in Section 4 we show how this can be applied to extract features from 3D-surface models and show in Section 5 experiments on the Princeton Shape Benchmark. Finally we give a conclusion and an outlook for future work.

## 2 Life in $SO(3)$

First, some preliminaries about the notation. We always assume complex-valued vector spaces. Finite dimensional vectors  $\mathbf{x}$  are printed bold face, their  $i$ th component  $x_i$  in normal face. Transposition is denoted by  $\mathbf{x}^T$ , so scalar products are written by  $\mathbf{x}^T \mathbf{x}'$ , complex conjugation by  $x_i^*$ . Functions  $X$  (infinite dimensional vectors) are printed in capital letters and scalar products between functions are denoted by  $\langle X|X' \rangle$ .

### 2.1 Spherical Harmonics

The Spherical Harmonic expansion is a basic tool for 3D shape representation. We first want to give a short review about its basic properties. The complex functions defined on the two-sphere  $S^2$  form a Hilbert-space with the inner-product  $\langle X|X' \rangle = \int_{S^2} X^*(\mathbf{s}) X'(\mathbf{s}) d\mathbf{s}$ , where  $d\mathbf{s}$  denotes the natural measure on  $S^2$  and  $\mathbf{s} \in S^2$  is some unit-vector on the two-sphere. The Spherical Harmonics form an orthonormal basis in this space. They are commonly denoted by  $Y_m^l(\mathbf{s})$ , where  $l \geq 0$  is a spectral index and  $-l \leq m \leq l$ . Basically the  $Y_m^l$  are polynomials of degree  $l$  in the normalized coordinates  $\mathbf{s} = (x, y, z)^T$ . We can expand any function  $X$  in our Hilbert space in terms of Spherical Harmonics where the expansion coefficients are simply the projections on the basis functions  $\mathbf{a}^l = \langle \mathbf{Y}^l|X \rangle$ , where this abbreviates  $a_m^l = \langle Y_m^l|X \rangle$ . The main property of a Spherical Harmonic expansion is its behavior under rotations. Let  $g \in SO(3)$  an element of the rotation group acting on functions  $X$  by  $(gX)(\mathbf{s}) \mapsto X(\mathbf{R}^T \mathbf{s})$ , where  $\mathbf{R} \in \mathbb{R}^{3 \times 3}$  is the corresponding rotation matrix, then the expansion coefficients have the following property

$$\sum_{m'=-l}^{m'=l} D_{mm'}^l(g) a_{m'}^l = \langle Y_m^l | gX \rangle,$$

or shortly  $\mathbf{D}^l(g)\mathbf{a}^l = \langle \mathbf{Y}^l | gX \rangle$ . Note the close relation  $D_{m0}^l = Y_m^l$ . The  $\mathbf{D}^l(g)$  are unitary transformation matrices depending on the rotation  $g$ . Note that only coefficients  $a_m^l$  with the same index  $l$  mix with each other, i.e. the subspaces with fixed index  $l$  stay invariant. This property is often used to obtain invariance against rotations. Due to the unitarity of  $\mathbf{D}^l(g)$  the energy within a subspace is preserved. One can easily obtain invariant features of the function  $X$  by taking the magnitudes  $\|\mathbf{a}^l\| = \sqrt{\sum_{m=-l}^{m=l} |a_m^l|^2}$ , which are sometimes called SH-descriptors, analog to the Fourier-Descriptors in 2D.

### 2.2 D-Wigner Matrix

Let us have a closer look on the  $\mathbf{D}^l(g)$  itself. For  $l = 1$  there is a close relation to the real-valued ordinary rotation matrix  $\mathbf{R}$  by a special linear unitary transformation  $\mathbf{U}$ , i.e.  $\mathbf{D}^1(g) = \mathbf{U}^T \mathbf{R} \mathbf{U}$ . The general  $\mathbf{D}^l(g)$  are called the *D-Wigner matrices* and they are the irreducible representations of the three dimensional rotation group. Irreducibility means that there is no further linear decomposition of the  $a_m^l$  such that the corresponding subspaces do not mix up during rotations (for references concerning the related group theory see e.g. [5] and references therein) The irreducibility has several important consequences: in fact, the  $\mathbf{D}^l(g)$  are an orthogonal basis for the functions defined on the rotation group itself. Before going into detail let us introduce some basics. We obtain a Hilbert-space whose elements are the functions defined on  $SO(3)$  by introducing an inner product via the group integral. Let  $Z, Z' : SO(3) \mapsto \mathbb{C}$  two functions defined on the rotation group, then

$$\langle Z | Z' \rangle = \int_{SO(3)} Z^*(g) Z'(g) dg$$

defines a regular inner product, where  $dg$  is the natural group measure on  $SO(3)$ , which is left- and right-invariant (in Euler angles  $dg = d\psi d\varphi \sin \theta d\theta$ ). Now we are able to state the orthogonality relation for the irreducible representations

$$\langle D_{m_1' m_1}^{l_1} | D_{m_2' m_2}^{l_2} \rangle = \delta_{l_1 l_2} \delta_{m_1' m_2'} \delta_{m_1 m_2} \frac{8\pi^2}{2l_1 + 1},$$

i.e. any components of the representation matrices  $\mathbf{D}^l(g)$  are orthogonal with respect to the given inner product. Now, given a function  $Z$  we can expand this function in terms of D-Wigner matrices as follows

$$Z(g) = \sum_{l=0}^{\infty} \sum_{m=-l}^l \sum_{m'=-l}^l b_{m'm}^l D_{m'm}^l(g),$$

where the  $\mathbf{b}^l$  are expansion-'matrices' obtained by the projections on the basis-functions  $b_{m'm}^l = \frac{2l+1}{8\pi^2} \langle D_{m'm}^l | Z \rangle$ .



Now the question arises how the  $\mathbf{D}^l(g)$  look like explicitly. There is an expression involving the Jacobi-polynomials depending on the Euler-angles corresponding to  $g$ . The direct computation of this expression is a little bit cumbersome and of high computational complexity. Moreover the parameterization in Euler-angles is not advantageous for our purposes. We need a formulation in terms of the corresponding three-dimensional rotation matrix itself. In the following we want to point out the alternative way.

### 2.3 Product Relations and Clebsch-Gordan Coefficients

As we know that any function on the sphere can be expanded in terms of Spherical Harmonics, then also products of two Spherical Harmonics must have such a representation. The corresponding expansion coefficients are the so called Clebsch-Gordan coefficients.

$$Y_{m_1}^{l_1}(\mathbf{s})Y_{m_2}^{l_2}(\mathbf{s}) = \sum_{l=0}^{l_1+l_2} \sum_{m=-l}^{m=l} \langle lm|l_1m_1l_2m_2\rangle Y_m^l(\mathbf{s}) \tag{1}$$

The Clebsch-Gordan coefficients  $\langle lm|l_1m_1l_2m_2\rangle$  fulfill two selection rules. They only give a contribution when  $|l_1-l_2| \leq l \leq l_1+l_2$  and  $m = m_1+m_2$ . Additionally the Clebsch-Gordan coefficients themselves fulfill several orthogonality relation by what we can reformulate equation (1) as follows

$$Y_m^l(\mathbf{s}) = \sum_{m_1=-l_1}^{m_1=l_1} \sum_{m_2=-l_2}^{m_2=l_2} \langle lm|l_1m_1l_2m_2\rangle Y_{m_1}^{l_1}(\mathbf{s})Y_{m_2}^{l_2}(\mathbf{s}),$$

where  $l_1$  and  $l_2$  have to be chosen such that  $l = l_1 + l_2$  due to the selection rules of the Clebsch-Gordan coefficients. By choosing  $l_1 = l - 1$  and  $l_2 = 1$  we have an iterative way to compute the Spherical Harmonics. The computation of the  $\mathbf{Y}^l(\mathbf{s})$  only involves  $\mathbf{Y}^{l-1}(\mathbf{s})$  and  $\mathbf{Y}^1(\mathbf{s})$ . For a fast implementation the Clebsch-Gordan coefficients can be precomputed and stored in a lookup-table. Considering the selection rules the actual algorithm to compute the Spherical Harmonics is a convolution-like method. In fact, the overall computational complexity is linear in the number of coefficients to be computed.

The computation of the D-Wigner matrices is very much the same as above. Products of D-Wigner matrix elements show nearly the same behavior, but needing products of Clebsch-Gordan coefficients. The basis for the iteration is now

$$D_{m'm}^l(g) = \sum_{\substack{m'_1=l_1 \\ m_1=l_1 \\ m'_1=-l_1 \\ m_1=-l_1}}^{m'_1=l_1} \sum_{\substack{m'_2=l_2 \\ m_2=l_2 \\ m'_2=-l_2 \\ m_2=-l_2}}^{m'_2=l_2} \langle lm'|l_1m'_1l_2m'_2\rangle \langle lm|l_1m_1l_2m_2\rangle D_{m'_1m_1}^{l_1}(g)D_{m'_2m_2}^{l_2}(g),$$

and we again choose  $l_1 = l - 1$  and  $l_2 = 1$ . The computational complexity is again linear in the number of computed matrix elements.

### 3 Group Integration Features

Group Integration is a well known tool to gain invariance for object representations. Suppose a given 3D-object  $X$  which has to be represented invariant against rotations. The easiest way to get an invariant feature is to extract some non-invariant non-linear 'kernel'-feature  $f(X)$  and sum up this feature for all rotational poses

$$I_f(X) = \int_{SO(3)} f(gX) dg$$

Due to the integration lots of information about the original object is getting lost. In previous work [7] we have shown how to use Spherical Harmonics to keep more discriminative power. We will shortly review how this was done. The integral over a 3D rotation can be decomposed into an integration over the sphere  $S^2$  and the circle:

$$I_f(X) = \int_{S^2} \int_0^{2\pi} f(g_{\mathbf{s},\varphi}X) d\varphi ds,$$

where  $g_{\mathbf{s},\varphi}$  is a rotation around the axis  $\mathbf{s}$  with angle  $\varphi$ . Now, instead of just integrating over the sphere we projected the inner integral  $F_X(\mathbf{s}) = \int_0^{2\pi} f(g_{\mathbf{s},\varphi}X) d\varphi$  on the  $\mathbf{Y}^l(\mathbf{s})$  by  $\mathbf{I}_f^l(X) = \langle \mathbf{Y}^l | F_X \rangle$  and obtained invariance by taking the band-wise magnitudes  $\|\mathbf{I}_f^l(X)\|$ . Doing this we already retained much information, but we still lose information about the  $\phi$ -angle.

The D-Wigner matrices offer a much more natural way to extend the discriminative power of the features. Instead of an artificial decomposition of the integral we can simply project the function  $F_X(g) := f(gX)$  on the irreducible group representation itself

$$\mathbf{I}_f^l(X) = \langle F_X | \mathbf{D}^l \rangle = \int_{SO(3)} f^*(gX) \mathbf{D}^l(g) dg \tag{2}$$

But what effect does have a rotation of the object  $X \mapsto g'X$  on the  $\mathbf{I}_f^l(X)$ ? Since the  $\mathbf{D}^l(g)$  are unitary representations of the rotation group, i.e.  $\mathbf{D}^l(gg') = \mathbf{D}^l(g)\mathbf{D}^l(g')$ , we can show that

$$\mathbf{I}_f^l(gX) = \mathbf{I}_f^l(X) \mathbf{D}^l(g^{-1}),$$

i.e. a rotation of the object  $X$  leads to right-multiplication of the features matrices  $\mathbf{I}_f^l$  with  $\mathbf{D}^l$ . This means that the magnitudes of the columns are preserved during rotations and hence form an invariant feature. The final invariant features we use look as follows

$$\mathbf{I}_f^{l,m}(X) = \sqrt{\sum_{m'=-l}^{m'=l} |I_{f,mm'}^l(X)|^2},$$

where  $I_{f,mm'}^l(X)$  denote the components of the feature matrices  $\mathbf{I}_f^l(X)$ .

## 4 Application to Surface Models

In [8] the group integration framework was already applied to surface models. The surface model is represented by a function  $\mathbf{X} : \mathbb{R}^3 \mapsto \mathbb{R}^3$  giving only contribution on the surface  $\mathcal{S}$  of the shape, where the function value on the surface is given by the surface-normal at this position.

### 4.1 First-Order

To obtain translation invariance we shift the origin of the coordinate system into the center of gravity of the shape  $\mathbf{X}$ , i.e. we first use a simple normalization approach. For the group integration we will use the following kernel-function

$$f_{\mathbf{h},\mathbf{r}}(\mathbf{X}) = \delta(\mathbf{h}^T \mathbf{X}(\mathbf{r}) - 1), \quad (3)$$

where  $\mathbf{r}, \mathbf{h} \in \mathbb{R}^3$  are parameters with  $\|\mathbf{h}\| = 1$  and  $\delta$  is the Delta-Distribution. The kernel-function gives contribution whenever at position  $\mathbf{r}$  the surface is present and its normal is parallel to  $\mathbf{h}$ . Inserting the kernel into (2) gives

$$\begin{aligned} \mathbf{I}_{\mathbf{h},\mathbf{r}}^l(\mathbf{X}) &= \int_{SO(3)} \delta(\mathbf{h}^T \mathbf{R} \mathbf{X}(\mathbf{R}^T \mathbf{r}) - 1) \mathbf{D}^l(g) dg \\ &= \int_{SO(3)} \int_{\mathbb{R}^3} \delta(\mathbf{h}^T \mathbf{R} \mathbf{X}(\mathbf{r}') - 1) \delta(\mathbf{R}^T \mathbf{r} - \mathbf{r}') \mathbf{D}^l(g) dg d\mathbf{r}'. \end{aligned}$$

We see that the integral only gives contribution, whenever  $\|\mathbf{r}\| = \|\mathbf{r}'\|$  and  $\mathbf{r}' \parallel \mathbf{R}^T \mathbf{r}$  and  $\mathbf{h} \parallel \mathbf{R} \mathbf{X}(\mathbf{r}')$ . The last two conditions are only satisfiable, if  $\mathbf{r}^T \mathbf{h} = \mathbf{r}'^T \mathbf{X}(\mathbf{r}')$ . If they are satisfied they determine the rotation matrix  $\mathbf{R}$  uniquely. So the group integral disappears and only the  $\mathbf{r}'$ -integral is left over. As the function  $\mathbf{X}$  gives contribution on the surface of the shape, the volume-integral is actually a surface-integral and we arrive at

$$\mathbf{I}_{\mathbf{h},\mathbf{r}}^l(\mathbf{X}) = \int_{\mathbf{r}' \in \mathcal{S}} \delta(\|\mathbf{r}'\| - r) \delta\left(\frac{\mathbf{r}'^T \mathbf{X}(\mathbf{r}')}{\|\mathbf{r}'\|} - \alpha\right) \mathbf{D}^l(g^*) d\mathbf{r}', \quad (4)$$

where  $r = \|\mathbf{r}\|$  and  $\alpha = \frac{\mathbf{r}^T \mathbf{h}}{\|\mathbf{r}\|}$  (for illustration see Figure 1). The  $g^*$  denotes the rotation which turns  $\mathbf{r}', \mathbf{X}(\mathbf{r}')$  into  $\mathbf{r}, \mathbf{h}$ . One can see that  $g^*$  is the only part that depends on the actual values of  $\mathbf{r}$  and  $\mathbf{h}$ , the rest on their relative directions and the length of  $\mathbf{r}$ . A joint rotation of the two parameters leads to a left-multiplication of  $\mathbf{I}_{\mathbf{r},\mathbf{h}}^l$  with  $\mathbf{D}^l$ . As this is only a unitary transformation of the features, we can restrict us for computation to one standard pose of  $\mathbf{h}$  and  $\mathbf{r}$  and hence the features depend on  $r$  and  $\alpha$  only. But the actual invariant features depend on the actual choice of the standard pose. Following expression (4) for the integral the algorithm looks as follows:

Start with result array  $\mathbf{I}_{r,\alpha}^l$  initialized with zeros.

For all points  $\mathbf{r}'$  on the surface of the object

    Compute  $\alpha = \frac{\mathbf{r}'^T \mathbf{X}(\mathbf{r}')}{\|\mathbf{r}'\|}$ ,  $r = \|\mathbf{r}'\|$

    Compute  $g^*$ , which turns  $\mathbf{r}', \mathbf{X}(\mathbf{r}')$  into  $\mathbf{r}_{\text{norm}}, \mathbf{h}_{\text{norm}}$

    Update  $\mathbf{I}_{r,\alpha}^l = \mathbf{I}_{r,\alpha}^l + \mathbf{D}^l(g^*)$  for all  $l \leq l_{\text{max}}$

The vectors  $\mathbf{r}_{\text{norm}}, \mathbf{h}_{\text{norm}}$  are the parameter-vectors in normalized pose such that  $\alpha = \frac{\mathbf{r}_{\text{norm}}^T \mathbf{h}_{\text{norm}}}{\|\mathbf{r}_{\text{norm}}\|}$  and  $\|\mathbf{r}_{\text{norm}}\| = r$ . Actually we compute for  $l = 0$  something like a histogram. We count how often a point on the surface of the shape occur within a distance  $r$  to the COG and with angle  $\arccos(\alpha)$  between the surface-normal and the vector connecting the point with the COG.

## 4.2 Second-Order

Now we assume an unnormalized model and obtain translation invariance by group-integration. We first perform a group integration over the three dimensional translation group  $\mathcal{T}$  and treat the result as above by projecting it on the irreducible representations. Equation (2) becomes

$$\mathbf{I}_l(\mathbf{X}) = \int_{SO(3)} \left( \int_{\mathcal{T}} f^*(\tau g \mathbf{X}) d\tau \right) \mathbf{D}^l(g) dg.$$

In this case the kernel-function (3) is too simple, the results would be undiscriminative. We need a more complex kernel-function. A simple generalization of (3) is

$$f_{\mathbf{h}, \mathbf{h}', \mathbf{d}}(\mathbf{X}) = \delta(\mathbf{h}^T \mathbf{X}(0) - 1) \delta(\mathbf{h}'^T \mathbf{X}(\mathbf{d}) - 1).$$

We do not want to give the complete derivation again since it is very much the same like in the first case, so we outline the resulting algorithm directly.

Start with result array  $\mathbf{I}_{d, \alpha, \beta, \gamma}^l$  initialized with zeros.

For all pairs of points  $\mathbf{r}_1, \mathbf{r}_2$  on the surface of the object

Let  $\mathbf{d} = \mathbf{r}_1 - \mathbf{r}_2$  and  $d = \|\mathbf{d}\|$

Compute  $\alpha = \mathbf{d}^T \mathbf{X}(\mathbf{r}_1)/d$ ,  $\beta = \mathbf{d}^T \mathbf{X}(\mathbf{r}_2)/d$

Compute  $\gamma = \frac{\mathbf{P}_d \mathbf{X}(\mathbf{r}_1)^T \mathbf{P}_d \mathbf{X}(\mathbf{r}_2)}{\|\mathbf{P}_d \mathbf{X}(\mathbf{r}_1)\| \|\mathbf{P}_d \mathbf{X}(\mathbf{r}_2)\|}$

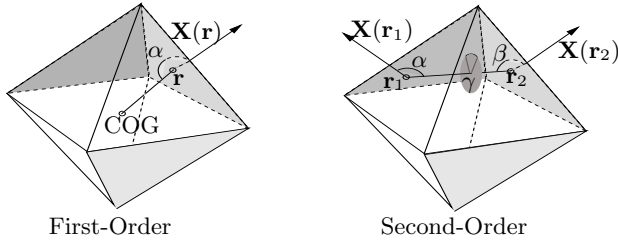
Compute  $g^*$ , which turns  $\mathbf{d}, \mathbf{X}(\mathbf{r}_1), \mathbf{X}(\mathbf{r}_2)$  into  $\mathbf{d}_{\text{norm}}, \mathbf{h}_{\text{norm}}, \mathbf{h}'_{\text{norm}}$

Update  $\mathbf{I}_{d, \alpha, \beta, \gamma}^l = \mathbf{I}_{d, \alpha, \beta, \gamma}^l + \mathbf{D}^l(g^*)$  for all  $l \leq l_{\text{max}}$

The matrix  $\mathbf{P}_d$  denotes a projection matrix, projecting on the plane given by the vector  $\mathbf{d}$ . For illustration of the variables see Figure 1. Again  $\mathbf{d}_{\text{norm}}, \mathbf{h}_{\text{norm}}, \mathbf{h}'_{\text{norm}}$  are the parameters in standard pose. For  $l = 0$  the feature can also be interpreted as a histogram, the frequency of occurrences of two points within a specific distance  $d$  and a specific surface-normal configuration determined by the parameters  $\alpha, \beta, \gamma$ .

## 4.3 Implementation Details

For evaluation of our features we use the Princeton Shape Benchmark [9]. It consists of approx. 1800 polygonal mesh models. We represent the models in  $256^3$  voxel grid. Additionally, each voxel gets a reference to the triangle it is stemming from to enable us to incorporate the original surface normals in our calculations. To exclude non visible constructional artefacts inside the closed surface we use a floodfill operation.



**Fig. 1.** Explanation of variables. For first-order just the length of  $\mathbf{r}$  and the angle between  $\mathbf{r}$  and  $\mathbf{X}(\mathbf{r})$  are relevant. For second-order we have four parameters,  $\alpha, \beta, \gamma$  and distance between  $\mathbf{r}_1$  and  $\mathbf{r}_2$ . The  $\gamma$ -angle is the angle between  $\mathbf{X}(\mathbf{r}_1)$  and  $\mathbf{X}(\mathbf{r}_2)$  after projection onto the plane orthogonal to  $\mathbf{r}_1 - \mathbf{r}_2$ .

The implementation of the first-order features is straight-forward, because each voxel has to be accessed only once. The time for a computation of one feature set is less than a second (*Pentium4* 2.8Ghz), where most of the time is spend on voxelization. For the second-order features the complexity is quadratic in the number of voxels. Typical models consist of several tens of thousand voxels, hence an exhaustive computation is not possible. We use a randomized approach to keep the computation time low. Computation times are varying from one to several seconds, depending on the accuracy of the computation and the number of D-Wigner coefficients.

For the computation of Clebsch-Gordan coefficients we use *Matpack*. Due to the precomputation of the coefficients the running time does not depend on their implementation. We tried several discretizations of the parameters  $r, \alpha$  and  $d, \alpha, \beta, \gamma$ . In the experiments we always give the results for the best quantizations, which always depend on the type of feature. To obtain invariance we take the magnitudes of the rows of the 'matrix'-features  $\mathbf{I}^l$ . As already mentioned the features depend on the absolute pose of the parameter-vectors  $\mathbf{r}_{\text{norm}}$  and  $\mathbf{h}_{\text{norm}}$  ( $\mathbf{d}_{\text{norm}}, \mathbf{h}_{\text{norm}}, \mathbf{h}'_{\text{norm}}$  for second-order). But which pose one should choose? The most simple representation  $\mathbf{h}_{\text{norm}} = (\alpha, \sqrt{1 - \alpha^2}, 0)^T$  and  $\mathbf{r}_{\text{norm}} = (r, 0, 0)^T$  has the disadvantage that the magnitude of the rows of  $\mathbf{D}^l$  are the same for  $m$  with the same absolute value. We have found that the complex representation  $\mathbf{h}_{\text{norm}} = (\sqrt{\frac{1-\alpha^2}{2}}e^{i\frac{\pi}{4}}, \alpha, \sqrt{\frac{1-\alpha^2}{2}}e^{-i\frac{\pi}{4}})^T$  and  $\mathbf{r}_{\text{norm}} = (0, r, 0)^T$  keeps the magnitudes of the rows of  $\mathbf{D}^l$  more independent. And it actually performs better than the first simple one.

## 5 Experiments

In order to keep the results comparable to experiments given in [9] we conducted our experiments only on the test set of the PSB at the finest granularity. To show the superiority of the D-Wigner matrices over the Spherical Harmonics we also give the results for the best corresponding SH-feature (see [8]). As distance measure between the features we use the  $L_1$ -norm. In Table 1 the results for the first- and

second-order features are shown. For a description of the used performance measures Nearest-Neighbor/1-Tier/2-Tier/E-Mesure/Discounted-Cumulative-Gain see [9]. Experiments were made for different cutoff indices  $l_{\max}$ . We found that higher cutoff indices than 4 show only marginal improvements if at all.

**Table 1.** Results for finest granularity on the PSB. All results are given in percent. Higher rates mean better performance. (DW) stands for our new approach followed by  $l_{\max}$  the cutoff-band. (SH) for the old Spherical Harmonic approach. Best results in bold face.

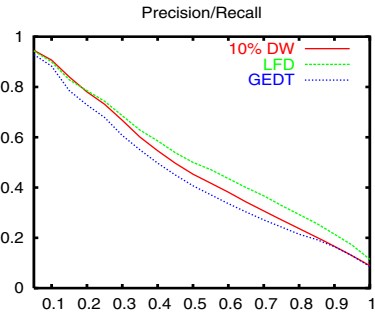
First-Order						Second-Order					
method	NN	1T	2T	EM	DCG	method	NN	1T	2T	EM	DCG
DW 1	48.3	24.6	34.4	19.8	52.3	DW 1	60.5	31.6	42.2	24.5	59.3
DW 2	53.4	28.2	38.2	21.9	55.4	DW 2	62.5	32.7	43.9	<b>25.5</b>	60.1
DW 3	54.0	28.3	38.0	22.0	55.5	DW 3	<b>63.5</b>	32.7	43.8	<b>25.5</b>	60.2
DW 4	<b>55.3</b>	<b>29.1</b>	<b>38.5</b>	<b>22.5</b>	<b>56.1</b>	DW 4	<b>63.5</b>	<b>32.9</b>	<b>44.0</b>	<b>25.5</b>	<b>60.5</b>
SH 4	50.1	26.1	36.1	21.0	53.6	SH 4	60.6	31.5	42.9	24.7	59.4

We see that the D-Wigner expansion is superior to SH expansion in all cases, which gives evidence that the proposed descriptors can, in fact, carry more information and discriminative power than the pure Spherical Harmonic approach. Of course, in this comparison, the number of features is larger in the D-Wigner case, because for each  $l$  we have  $2l + 1$  features instead of only one. The absolute number is also rather high, because the quantization setting for the D-Wigner case is 32, 2, 8, 8, (for  $d, \gamma, \alpha, \beta$ ) resulting in several thousand of features. So we additionally tried to reduce the number of features by feature selection using *Simba* [3]. In Figure 2 we show the performance results and precision/recall graph with 10% of the second-order features with  $l_{\max} = 4$ . In fact, feature selection improves the results very much, while reducing the number of features drastically. We also give results of two other shape descriptors, LFD and GEDT (for references see [9]). The PR-graph shows that our approach works with the same precision as the LFD approach, while recall is comparable to the GEDT method. It is astonishing that a group integration approach, which keeps second-order information only, i.e. relative properties about two points averaged over the whole shape, give us similar results like LFD which is basically a registration approach, or GEDT which keeps nearly the whole information about the shape.

## 6 Conclusion and Future Work

We presented how the irreducible representation of the 3D rotation group can be used for invariant shape representation. The D-Wigner expansion can be seen as the canonical generalization of the Spherical Harmonic expansion. We applied the D-Wigner expansion in the group integration framework and were able to show the superiority of them over SH-expansion in a shape retrieval task.

method	NN	1T	2T	EM	DCG
10% DW	65.7	34.8	46.0	26.7	62.1
LFD	65.7	38.0	48.7	28.0	64.3
GEDT	60.3	31.3	40.7	23.7	58.1



**Fig. 2.** Results for the second-order features with feature selection in comparison to LFD and GEDT. (For references see [9]).

The performance of the features is comparable to the currently best performing methods on the Princeton Shape Benchmark, but our approach is 'single-feature'-method, while LFD is a 'compositions' of features. For future work we want to apply our methods for volume data. Further we want to examine other methods to obtain invariance instead of just taking the magnitude of the rows of the 'matrix'-features.

## References

1. H. Burkhardt and S. Siggelkow. *Invariant features in pattern recognition - fundamentals and applications*. In *Nonlinear Model-Based Image/Video Processing and Analysis*. John Wiley and Sons, 2001.
2. D. Chen, X. Tian, Y. Shen, and M. Ouhyoung. On visual similarity based 3d model retrieval. In *Computer Graphics Forum*, volume 22/3, 2003.
3. R. Gilad-Bachrach, A. Navot, and N. Tishby. Margin based feature selection - theory and algorithms. In *Proc. of ICML04*, 2004.
4. M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz. Rotation invariant spherical harmonic representation of 3d shape descriptors. In *Symposium on Geometry Processing*, 2003.
5. W. Miller. Topics in harmonic analysis with applications to radar and sonar. *IMA Volumes in Mathematics and its Applications*, 1991.
6. R. Osada, T. Funkhouser, and B. Chazelle und D. Dobkin. Matching 3d models with shape distribution. In *Proceedings Shape Modeling International*, 2001.
7. M. Reisert and H. Burkhardt. Invariant features for 3d-data based on group integration using directional information and spherical harmonic expansion. In *the Proceedings of the ICPR06*, 2006.
8. M. Reisert and H. Burkhardt. Second order 3d shape features: An exhaustive study. *C&G, Special Issue on Shape Reasoning and Understanding*, 30(2), 2006.
9. P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser. The princeton shape benchmark. In *Shape Modeling International, Genova, Italy*, 2004.
10. D.V. Vranic, D. Saupe, and J. Richter. Tools for 3d-object retrieval: Karhunen-loeve transform and spherical harmonics. In *Proceedings of the IEEE Workshop Multimedia Signal Processing*, 2001.

# Shape Matching by Variational Computation of Geodesics on a Manifold\*

Frank R. Schmidt, Michael Clausen, and Daniel Cremers

Department of Computer Science  
University of Bonn  
Römerstr. 164, 53117 Bonn, Germany  
{schmidtf, clausen, dcremers}@cs.uni-bonn.de

**Abstract.** Klassen et al. [9] recently developed a theoretical formulation to model shape dissimilarities by means of geodesics on appropriate spaces. They used the local geometry of an infinite dimensional manifold to measure the distance  $\text{dist}(A, B)$  between two given shapes  $A$  and  $B$ . A key limitation of their approach is that the computation of distances developed in the above work is inherently unstable, the computed distances are in general not symmetric, and the computation times are typically very large. In this paper, we revisit the shooting method of Klassen et al. for their angle-oriented representation. We revisit explicit expressions for the underlying space and we propose a gradient descent algorithm to compute geodesics. In contrast to the shooting method, the proposed variational method is numerically stable, it is by definition symmetric, and it is up to 1000 times faster.

## 1 Introduction

The modeling of shapes and distances between shapes is one of the fundamental problems in Computer Vision with applications in the fields of image segmentation, tracking, object recognition, and video indexing. In recent years, a considerable amount of effort has been put into the understanding of closed planar curves modulo some transformations, which will be referred to as shapes. To measure the dissimilarity between two given shapes requires the definition and examination of metric spaces which model shapes (cf. [7,4,5,1]).

In 2003, Michor and Mumford [11] described a way to define a shape space using manifolds. The distance between two given shapes were defined as the minimal length of a path  $m$  on the manifold connecting these shapes. Such paths are known as *geodesics*. The model is presented in a very general fashion, i.e., in order to calculate geodesics on this manifold, a partial differential equation (PDE) has to be solved. Hence, it is not suitable for online-calculation to find the shortest geodesic between two given shapes.

In the same year, Klassen et al. also presented metric spaces using manifolds [9].<sup>1</sup> Their model is focused on closed planar curves parametrized by arclength.

---

\* This work was supported by the German Research Foundation, grant #CR-250/1-1.

<sup>1</sup> For an extension of the notion of geodesics to closed curves embedded in  $\mathbb{R}^3$  we refer to [8].



This simplification led to an ordinary differential equation (ODE) instead of a PDE for the geodesic-calculation. Moreover, the calculation of the shortest geodesic could in many cases be done within seconds using the so called *shooting method*. This method uses a searching beam from the initial shape. That beam will be changed until the target shape is found, where the beam is deformed according to the underlying metric just as a light beam is bent by gravity in the theory of general relativity.

In this paper, we will use the same manifold that was introduced in [9]. But we will abandon the shooting method and replace it by a *variational method* which is more stable, by definition symmetric and allows a faster algorithm than the algorithm introduced by Klassen et al. This variational method is a gradient descent method with respect to the energy functional  $E(m) := \int_0^1 \langle m'(t), m'(t) \rangle_{m(t)} dt$ .

This paper is organized as follows. In Section 2 we revisit the shape space and a toolbox of helpful functions that were presented in [9]. In Section 3 we review the shooting method and introduce an alternative variational method to calculate geodesics on the shape space. In Section 4 we compare both methods with special interest on correctness, accuracy and computation time. In Section 5 we provide a conclusion.

## 2 Modeling Shapes

Given any smooth closed planar contour  $\Gamma \subset \mathbb{C}$ , the group of translations, rotations and uniform scalings creates a family  $[\Gamma]$  of closed planar contours. The elements of this family have all one property in common - *their shape*. Therefore, we will consider the set of all such families and call this set the shape space. In this section, we will revisit a manifold that was proposed in [9] to handle this shape space. We are especially interested in morphings, i.e., smooth short transformations from one given shape to another. On the manifold, these morphings will be described by geodesics [6,3]. Hence, on the shape space a metric is induced which provides a measure of the dissimilarity of two given shapes.

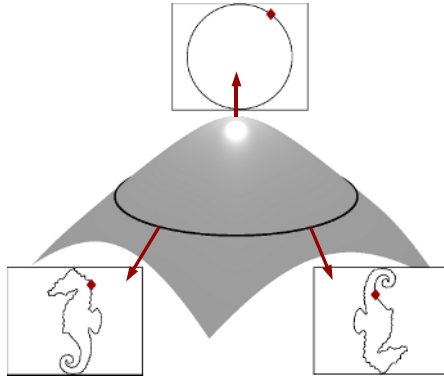
### 2.1 Manifold of Preshapes

To model shapes, we consider closed planar curves that are parametrized via the unit circle  $\mathbb{S}^1 = \{x \in \mathbb{C} \mid \|x\| = 1\}$ . A closed planar curve is therefore a  $C^\infty$ -mapping  $c : \mathbb{S}^1 \rightarrow \mathbb{C}$  with a non-vanishing derivative  $c'$ . Because the derivative of the mapping  $c$  ignores translation of the contour  $\Gamma$ , we will consider  $c'$  instead of  $c$ . To get rid of possible scalings, we fix the length of  $\Gamma$  by  $2\pi$ . This can be achieved by modeling any shape via curves  $c : \mathbb{S}^1 \rightarrow \mathbb{C}$  that are *parametrized by arclength*. Thus,  $c' : \mathbb{S}^1 \rightarrow \mathbb{S}^1$  can be modeled via a  $C^\infty$ -mapping  $\vartheta : [0; 2\pi] \rightarrow \mathbb{R}$  which realizes the following lifting-equations  $c'(e^{it}) = e^{i\vartheta(t)}$  and  $\vartheta(2\pi) = \vartheta(0) + 2\pi$ . This mapping  $\vartheta$  is unique up to addition of a constant  $2\pi\ell$ ,  $\ell \in \mathbb{Z}$ . Moreover, the addition of any  $r \in \mathbb{R}$  to  $\vartheta$  is equivalent to a rotation of  $c$  by the angle  $r$ . These observations lead to the manifold<sup>2</sup>  $\mathcal{C} := \Psi^{-1}((2\pi^2, 0, 0)^\top) \subset L^2 := L^2([0; 2\pi], \mathbb{R})$ ,

<sup>2</sup> In [9], two different manifolds were presented. We will restrict ourselves to the manifold that handles the angle-oriented mapping  $\vartheta$ .

$$\Psi(\vartheta) := \left( \int_0^{2\pi} \vartheta(\tau) \, d\tau, \int_0^{2\pi} \sin(\vartheta(\tau)) \, d\tau, \int_0^{2\pi} \cos(\vartheta(\tau)) \, d\tau \right)^\top.$$

As it was outlined in [9], this manifold  $\mathcal{C}$  does not describe the shape space. Moreover, one shape  $[\Gamma]$  can possess multiple representations in this so called preshape space  $\mathcal{C}$ . To be specific, for any  $\alpha \in \mathbb{R}$  the mappings  $c : t \mapsto c(e^{it})$  and  $c_\alpha : t \mapsto c(e^{i(t+\alpha)})$  describe the same closed planar contour  $\Gamma \subset \mathbb{C}$ . Let  $\vartheta$  and  $\vartheta_\alpha$  be the lifting representation for  $c$  resp.  $c_\alpha$ . Then, the set  $\{\vartheta_\alpha | \alpha \in [0; 2\pi[ \} =: \vartheta \cdot \mathbb{S}^1 \subset \mathcal{C}$  contains all different representations of  $\vartheta$  within  $\mathcal{C}$  that describe the same shape (cf. Figure 1). The notation  $\vartheta \cdot \mathbb{S}^1$  is motivated by the fact that



**Fig. 1.** Since any shape can be parametrized with differing starting points, it corresponds to a family of preshapes which form a closed curve on the manifold of preshapes. Symmetries of a given shape will be reflected by multiple coverings of this curve. In the case of a circle, this preshape curve will collapse to a single point.

$\alpha \mapsto \vartheta_\alpha$  is a group operation with at least  $2\pi\mathbb{Z}$  as stabilizer. The shape space  $\mathcal{S} := \mathcal{C}/\mathbb{S}^1$  consists of all orbits  $\vartheta \cdot \mathbb{S}^1 \subset \mathcal{C}$  [9]. Therefore, any metric  $\text{dist}_{\mathcal{C}}$  on  $\mathcal{C}$  induces the metric

$$\text{dist}_{\mathcal{S}}(\vartheta_1 \cdot \mathbb{S}^1, \vartheta_2 \cdot \mathbb{S}^1) := \min_{s_1 \in \mathbb{S}^1} \min_{s_2 \in \mathbb{S}^1} \text{dist}_{\mathcal{C}}(\vartheta_1 \cdot s_1, \vartheta_2 \cdot s_2) \tag{1}$$

on  $\mathcal{S}$ . In the next section, we will discuss the metric on any manifold  $M$  that is induced by geodesics. This geodesic metric will be used as  $\text{dist}_{\mathcal{C}}$  and thus, induces  $\text{dist}_{\mathcal{S}}$  via (1).

### 2.2 Geodesics on Manifolds

In this section, we will present the idea of geodesics and two different ways to calculate geodesics. Let  $\mathbb{E}$  be a Euclidean  $k$ -dimensional vector space (e.g.,  $\mathbb{R}^k \subset \mathbb{R}^n$ ).  $E$  possesses a scalar product denoted by  $\langle \cdot, \cdot \rangle$ . Using this product, the length of any smooth path  $m : [0; 1] \rightarrow \mathbb{E}$  is  $\text{len}(m) := \int_0^1 \langle m'(t), m'(t) \rangle^{\frac{1}{2}} \, dt$ . The

distance of two arbitrary points  $x, y \in \mathbb{E}$  can be defined as the minimal length of smooth paths that connect these two points, i.e.,  $m(0) = x$  and  $m(1) = y$ . To any path  $m$ , there exists a path  $\tilde{m}$  of same image and length that is parametrized by arclength. Moreover, every path of minimal length that is parametrized by arclength also minimizes the energy-functional  $E(m) := \int_0^1 \langle m'(t), m'(t) \rangle dt$ . In the case of  $\mathbb{E} = \mathbb{R}^k$ , the Euler-Lagrange equation becomes  $0 \equiv \frac{d}{dt}m' \equiv m''$ . Paths which realize this equation are called *geodesics*.

Now, consider any embedded  $k$ -dimensional manifold  $M \subset \mathbb{R}^n$ , e.g., a sphere or a cylinder. At any point  $x \in M$  there exists the  $k$ -dimensional tangent space  $T_x M$ . On this tangent space the scalar product of  $\mathbb{R}^n$  induces a scalar product denoted by  $\langle \cdot, \cdot \rangle_x$ . Given any smooth path  $m: [0; 1] \rightarrow M$ , the *length* of this path can be calculated by  $\text{len}(m) := \int_0^1 \langle m'(t), m'(t) \rangle_{m(t)}^{\frac{1}{2}} dt$ . Analogously to the Euclidean space, geodesics can be defined and a geodesic equation can be found. In the Euclidean case,  $m'$  and  $m''$  are  $k$ -dimensional vector fields along  $m$ . In the case of a manifold, only  $m'$  is a  $k$ -dimensional vector field, i.e.,  $m'(t) \in T_{m(t)}M$ . On the other hand,  $m''$  is an  $n$ -dimensional vector field that can be split into a tangential ( $k$ -dimensional) vector field  $m''^{\text{tan}}$  and a normal vector field  $m''^{\text{nor}}$ . With this notations the geodesic equation becomes  $0 \equiv m''^{\text{tan}}$ . Given a starting point  $x \in M$  and a starting direction  $v \in T_x M$ , the following differential equation

$$m(0) = x \qquad m'(0) = v \qquad m''^{\text{tan}}(t) \equiv 0$$

can be uniquely solved by a path  $m_{x,v}: \mathbb{R} \rightarrow M$ . This property leads to the definition of the so called exponential mapping  $\exp_x(v) := m_{x,v}(1)$ . Using this mapping, the distance of two arbitrary points  $x, y \in M$  is

$$\text{dist}_M(x, y) := \min_{\substack{m \text{ smooth path,} \\ m(0) = x, m(1) = y}} \text{len}(m) = \min_{\substack{v \in T_x M: \\ y = \exp_x(v)}} \langle v, v \rangle_x^{\frac{1}{2}}. \tag{2}$$

While the shooting method used in [9] makes use of the exponential mapping starting from an initial velocity  $v$  as indicated on the right side of (2), the variational method proposed in this paper directly relies on the definition of  $\text{dist}_M$  in the middle of (2).

### 2.3 Technical Issues

In this section, we will revisit a toolbox of functions that was presented in [9]. One important function that was used in [9], is the projection from an arbitrary function  $\vartheta \in L^2([0; 2\pi], \mathbb{R}) \supset \mathcal{C}$  onto the manifold. In [9, Section 3.2; Case 1] such a projection was elaborated. We will denote the projection by

$$P_\varepsilon : L^2([0; 2\pi], \mathbb{R}) \rightarrow \mathcal{C}_\varepsilon,$$

where

$$\mathcal{C}_\varepsilon := \Psi^{-1} \left( \left\{ r \left\| r - (2\pi^2, 0, 0)^\top \right\| < \varepsilon \right\} \right)$$

describes the manifold  $\mathcal{C}$  which is thickened by  $\varepsilon$ .  $\mathcal{C}_\varepsilon$  is an open subset of  $L^2([0; 2\pi], \mathbb{R})$ , and thus a manifold which contains  $\mathcal{C}$  as a submanifold.

We will also need the projection from the space  $L^2([0; 2\pi], \mathbb{R})$  on the tangent space  $T_\vartheta\mathcal{C}$  for any function  $\vartheta \in \mathcal{C}$ . This projection can be computed efficiently due to the small codimension of  $\mathcal{C}$  ( $= 3$ ) [9]. From now on, this projection will be denoted

$$P_\vartheta : L^2([0; 2\pi], \mathbb{R}) \rightarrow T_\vartheta\mathcal{C}.$$

To measure the distance of two given *shapes*  $\vartheta_1 \cdot \mathbb{S}^1$  and  $\vartheta_2 \cdot \mathbb{S}^1$ , the expression

$$\inf_{s_1 \in \mathbb{S}^1} \inf_{s_2 \in \mathbb{S}^1} \|\vartheta_1 \cdot s_1 - \vartheta_2 \cdot s_2\|_{L^2} = \inf_{s \in \mathbb{S}^1} \|\vartheta_1 - \vartheta_2 \cdot s\|_{L^2}$$

has to be calculated. The last equation holds since  $\mathbb{S}^1$  operates as an isometry on  $\mathcal{C}$ . Moreover, finding the minimizing  $s \in \mathbb{S}^1$  can be calculated via Discrete Fourier Transform [10]. Thus, this calculation needs only  $O(n \log(n))$  multiplications [2]. The function to calculate  $s \in \mathbb{S}^1$  given the preshapes  $\vartheta_1$  and  $\vartheta_2$  will be denoted  $\text{dft}_{\mathcal{C}} : \mathcal{C} \times \mathcal{C} \rightarrow \mathbb{S}^1$ .

### 3 Calculating Local Shape Morphings

In this section, two different algorithm to calculate geodesics between given shapes will be presented. Both algorithms will calculate the distance defined by (2). The first algorithm was presented by Klassen et al. [9] and uses a linearization of the exponential mapping. The second algorithm – proposed in this paper – will use the geodesic equation as gradient descent to minimize the functional  $E(m)$ .

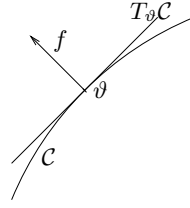
#### 3.1 Morphing Via the Exponential Mapping

As we have seen, the following functional can be calculated efficiently using  $\text{dft}_{\mathcal{C}}$ .

$$H_{\vartheta_1}^{\vartheta_2}(f) = \inf_{s \in \mathbb{S}^1} \|\exp_{\vartheta_1}(f) - \vartheta_2 \cdot s\|_{L^2}^2, \quad f \in T_{\vartheta_1}\mathcal{C}. \quad (3)$$

The linearization of  $\exp_{\vartheta_1}(f)$  is explained in detail in [9]. The distance between the orbits  $\vartheta_1 \cdot \mathbb{S}^1 \subset \mathcal{C}$  and  $\vartheta_2 \cdot \mathbb{S}^1 \subset \mathcal{C}$  is the minimal  $\|f\|$  of any  $f$  that realizes the minimal value of  $H_{\vartheta_1}^{\vartheta_2}(\cdot)$ .

The above method has some important drawbacks. First of all, the numerical stability of  $\exp_\vartheta(\cdot)$  depends very much on the curvature at the point  $\vartheta$ . Hence, one expects an asymmetric runtime behavior, because the curvature of  $\mathcal{C}$  is heterogenous. In addition, the last operation that is calculated in (3) is the shape alignment via  $\text{dft}_{\mathcal{C}}$ . Hence, this method can get stuck in a local minimum. In Section 4 we will provide an example of this problem. One additional drawback is the runtime of this method. In the next section, we propose an alternative variational approach to compute geodesics which resolves all these drawbacks.



**Fig. 2.** A deformation  $f$  from a given preshape  $\vartheta$  is orthogonal to the tangent space  $T_{\vartheta}\mathcal{C}$  at this given preshape, iff the projection of the deformed preshape  $\vartheta + f$  onto the preshape manifold  $\mathcal{C}$  is equal to  $\vartheta$

### 3.2 Morphing Via the Geodesic Equation

Instead of restricting ourselves to the tangent space of a preshape  $\vartheta_1$  and trusting in the numerical stability of  $\exp_{\vartheta_1}(\cdot)$ , let us consider the entire path from  $\vartheta_1$  to  $\vartheta_2$ . With the help of the geodesic equation  $m''^{\text{tan}} \equiv 0$  it is easy to verify, whether a given path is a geodesic or not. Moreover, the geodesic equation guarantees an equidistant path and thus, a variational approach will take care of an online gauge fix.

If  $m$  fails to be a geodesic,  $m'$  is a non-parallel vector field along  $m$  and  $m''^{\text{tan}}$  measures the curvature of  $m$  within the manifold  $\mathcal{C}$ . Let us observe this measure in a discretized version of  $m$  in detail. The path shall be discretized in  $n \in \mathbb{N}$  equidistant preshapes. Each preshape shall be discretized in  $N \in \mathbb{N}$  points. Thus, a discretized path is

$$m^{N,n} := \left( m(0)^N, \dots, m\left(\frac{i}{n-1}\right)^N, \dots, m(1)^N \right) \in \mathbb{R}^{N \times n} \quad , \text{ whereas}$$

$$\vartheta^N := \left( \vartheta(0), \dots, \vartheta\left(\frac{i}{N}\right), \dots, \vartheta\left(\frac{N-1}{N}\right) \right)^{\top} \in \mathbb{R}^N.$$

The discretized versions of  $P_{\varepsilon}$  and  $P_{\vartheta}$  will be known as  $P_{\varepsilon}^{\Delta}$  resp.  $P_{\vartheta}^{\Delta}$ . The vector field  $m'$  can be discretized by  $\frac{1}{n}m' \left( \frac{i+0.5}{n-1} \right) \approx m_{\cdot, i+1}^{N,n} - m_{\cdot, i}^{N,n}$  and the geodesic equation becomes

$$0 = P_{m_{\cdot, i}^{N,n}}^{\Delta} \left( \left( m_{\cdot, i+1}^{N,n} - m_{\cdot, i}^{N,n} \right) - \left( m_{\cdot, i}^{N,n} - m_{\cdot, i-1}^{N,n} \right) \right)$$

$$= -2P_{m_{\cdot, i}^{N,n}}^{\Delta} \left( m_{\cdot, i}^{N,n} - \frac{m_{\cdot, i-1}^{N,n} + m_{\cdot, i+1}^{N,n}}{2} \right).$$

Because of  $0 = P_{\vartheta}(f) \Leftrightarrow \vartheta \approx P_{\varepsilon}(\vartheta + f)$  (cf. Figure 2), we obtain the equation

$$m_{\cdot, i}^{N,n} = P_{\varepsilon}^{\Delta} \left( \frac{m_{\cdot, i-1}^{N,n} + m_{\cdot, i+1}^{N,n}}{2} \right). \quad (4)$$

Equation (4) can be interpreted as an iteration rule, that simulates the gradient descent and thus, moves a given path towards a geodesic. During this process the starting preshape  $m_{\cdot,1}^{N,n}$  and the target preshape  $m_{\cdot,n}^{N,n}$  aren't be altered. Thus, this process converges towards a geodesic from  $\vartheta_1$  to  $\vartheta_2$ . To calculate a geodesic from  $\vartheta_1$  to the orbit  $\vartheta_2 \cdot \mathbb{S}^1$ , we calculate a realignment in every iteration step. Our proposal for the iteration step is therefore<sup>3</sup>

```

Step 1: for (i=1; i<n-1; i++) {
     $m_{\cdot,i}^{N,n} := P_\varepsilon^\Delta \left( \frac{m_{\cdot,i-1}^{N,n} + m_{\cdot,i+1}^{N,n}}{2} \right)$ 
}
Step 2: for (i=1; i<n; i++) {
     $m_{\cdot,i}^{N,n} := m_{\cdot,i}^{N,n} * \text{dft\_C}(m_{\cdot,i}^{N,n}, m_{\cdot,i-1}^{N,n});$ 
}

```

By this extended iteration the starting preshape  $\vartheta_1$  remains fix but the target preshape  $\vartheta_2$  can be modified. This modification will only take place along the orbit  $\vartheta_2 \cdot \mathbb{S}^1$  and thus, the *shape* of  $\vartheta_2$  remains the target of the morphing.

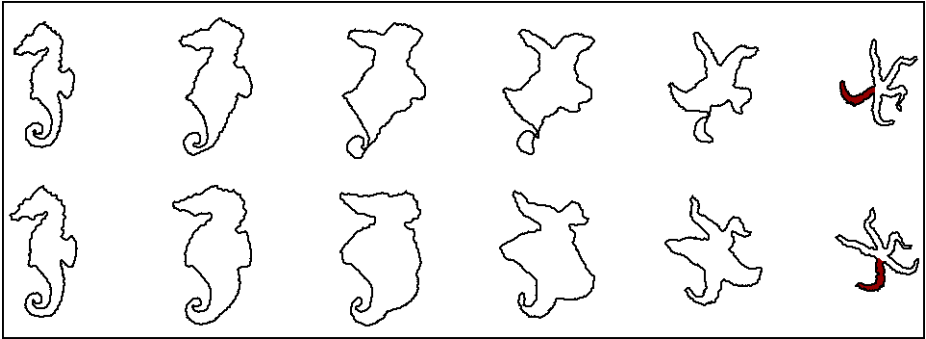
## 4 Benchmarking

To illustrate the difference between these two algorithms, we will discuss a specific morphing example. For this purpose, we have used the SQUID database of fish shapes [12]. In the first subsection, we will examine the morphing process between a seahorse and a starfish. This morphing will be done with a very high resolution ( $N = 500$  angles along each preshape;  $n = 300$  intermediate morphing steps). Specifically we show that the shooting method gets stuck in a local minimum, whereas the variational method calculates the global minimum with respect to shape realignments. In the second subsection we will analyze the speed of both algorithms for  $N = 500$  and  $n = 1, \dots, 100$ , showing in particular that our variational method computes minimas within seconds where the shooting method takes more than one hour.

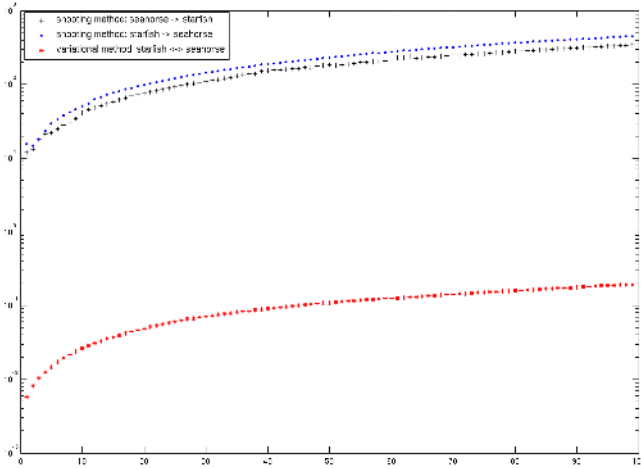
### 4.1 Dissimilarity of a Seahorse and a Starfish

Figure 3 shows the morphing of a seahorse towards a starfish. The first row shows the morphing according to the shooting method, whereas the second row shows the result of the variational method. Both are valid morphings of preshapes, but the calculated alignments are different in both algorithms. This leads to a self-intersection in the first case, whereas in the second case, the tail of the seahorse *unrolls* in an expected natural manner. This is due to the different alignments of the target shape. To emphasize the alignments, the same *region* of the target shape is colored. It's easy to see that the variational method moves the tip of the tail towards the tip of that region. Moreover, the first geodesic has the length 13.8, whereas the second geodesic has the length of 12.2. Therefore, the shooting method gets stuck in a local minimum.

<sup>3</sup> Note that each index starts at 0.



**Fig. 3.** The morphing of a seahorse towards a star fish is calculated via both algorithms. **First row:** The shooting method [9] gets stuck in a local minimum. **Second row:** Our variational method calculates the *global minimum* with respect to realignments. **Last column:** To illustrate the different alignments, the same region is colored in the target shape.



**Fig. 4.** Here the computation time to calculate geodesics is presented. The shapes are highly resolved and on the x-axis the discretization of the morphing is shown. The runtime via the variational method has two advantages. **Symmetry:** The geodesic calculation does not depend on the starting shape, whereas the runtime for the shooting method varies by ca. 25%. **Runtime:** The variational method is faster by a factor of 1000.

In this example, we used a discretization of  $N = 500$  for the preshapes. Therefore, there exist 500 different alignments for the target preshape. Calculating the geodesic distance between the preshapes with respect to all alignments, we could confirm that 12.2 is the global minimum with respect to preshape alignment. Thus, the calculation of realignments serves the purpose of finding the minimal distance between two given shapes.

## 4.2 Computation Time

Figure 4 shows the computation time of both methods. It varies from the computation time in [9] because we use highly resolved preshapes and the methods stop only if they can provide a very accurate result. On the horizontal axis the discretization resolution of the geodesic is noted. For the shooting method this is the discretization of the exponential mapping. For the variational method this is the number of shapes that discretize the path on the manifold. First of all, we see that the computation time is not symmetric for the shooting method. Moreover, the computation time varies by 20 to 30 percent. This is due to the fact that the shooting method depends highly on the curvature at the starting shape. The variational method is symmetric and thus, the runtime does not depend on the starting shape. In addition, the calculation time is less than 100 milliseconds in the highly resolved case. If we use the same resolution as in [9], the variational method takes only a few milliseconds.

## 5 Conclusion

We presented a new variational approach to calculate geodesics in the shape space introduced in [9]. This shape space consists of  $S^1$ -orbits within a manifold. We start with an arbitrary parameterization of two given shapes and a path between these two *points of the manifold*. This path is then shortened via our variational method by alternating a two-step iteration process. The first step uses a gradient descent method and the second step realigns efficiently all preshapes along the observed path.

The proposed variational approach has several advantages over the shooting method used in [9]: Firstly, it is more stable since in contrast to the exponential

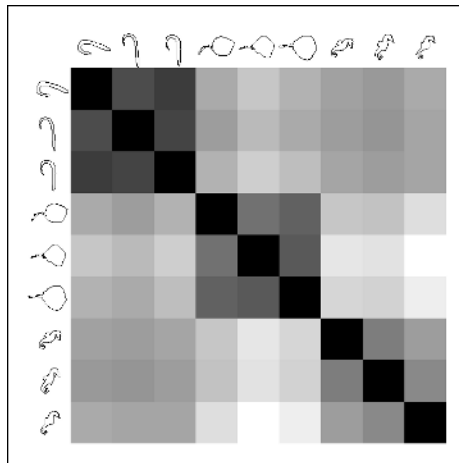


Fig. 5. The confusion matrix for a set of nine shapes



map, the variational method does not accumulate projection errors. Moreover, our gradient descent method provides an online gauge fix. Secondly, the formulation of the variational method does not rely on the starting shape, and thus the formulation of the metric is numerically symmetric. Thirdly, the calculation time is considerably smaller. In our examples, the calculation time typically improves by a factor of 1000. In addition, in our experiments our algorithm provides the *globally* optimal alignment between the given shapes. The practical implication of these drastic improvements in speed is that for a database of 100 shapes of high resolution, the confusion matrix consists of 4950 different entries and can be calculated in 8 minutes instead of 5 days<sup>4</sup>. Thus, the efficient use of this metric allows to cluster a considerable amount of shapes (cf. Figure 5). Future work will be focused on generalizing the concepts of geodesics on shape spaces to higher-dimensional shapes (e.g., surfaces).

## References

1. M. Bergtholdt and C. Schnörr. Shape priors and online appearance learning for variational segmentation and object recognition in static scenes. In *Pattern Recognition, Proc. of DAGM*. LNCS, Springer, 2005.
2. P. Bürgisser, M. Clausen, and M. A. Shokrollahi. *Algebraic Complexity Theory*. Grundlehren der Mathematischen Wissenschaften, Vol. 315. Springer-Verlag, 1997.
3. S. Chern, W. Chen, and K. S. Lam. *Lectures on Differential Geometry*. World Scientific, 1999.
4. D. Cremers, T. Kohlberger, and C. Schnörr. Shape statistics in kernel space for variational image segmentation. *Pattern Recognition*, 36(9):1929–1943, 2003.
5. D. Cremers and S. Soatto. A pseudo-distance for shape priors in level set segmentation. In N. Paragios, editor, *IEEE 2nd Int. Workshop on Variational, Geometric and Level Set Methods*, pages 169–176, Nice, 2003.
6. M. P. do Carmo. *Differential Geometry of Curves and Surfaces*. Prentice-Hall, 1976. 503 pages.
7. I. L. Dryden and K. V. Mardia. *Statistical Shape Analysis*. Wiley, Chichester, 1998.
8. E. Klassen and A. Srivastava. Geodesics between 3d closed curves using path-straightening. In *European Conference on Computer Vision (ECCV)*, volume 3951 of *LNCS*, pages 95–106, Graz, Austria, 2006. Springer.
9. E. Klassen, A. Srivastava, W. Mio, and S. H. Joshi. Analysis of planar shapes using geodesic paths on shape spaces. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(3):372–383, 2003.
10. J. S. Marques and A. J. Abrantes. Shape alignment – optimal initial point and pose estimation. *Pattern Recognition Letters*, 18(1):49–53, 1997.
11. P. Michor and D. Mumford. Riemannian geometries on spaces of plane curves. *J. of the European Math. Society*, 2003.
12. F. Mokhtarian, S. Abbasi, and J. Kittler. Efficient and robust retrieval by shape content through curvature scale space, 1996.

---

<sup>4</sup> Even at low resolution the shooting method needs more than one hour.

# A Modification of the Level Set Speed Function to Bridge Gaps in Data

Karsten Rink and Klaus Tönnies

Department of Simulation and Graphics  
University of Magdeburg, Germany  
{karsten, klaus}@isg.cs.uni-magdeburg.de

**Abstract.** Level set methods have become very popular means for image segmentation in recent years. But due to the data-driven nature of this methods it is difficult to segment objects that appear unconnected within the data. We propose a modification of the level set speed function to add a “bridging force” that allows the level set to leap over gaps in the data and segment an object despite artifacts or partial occlusions. We propose two methods to define such a force, one model-based and one image-based. Both versions have been applied to a series of test images, as well as medical data and photographic images to show their adequacy for image segmentation.

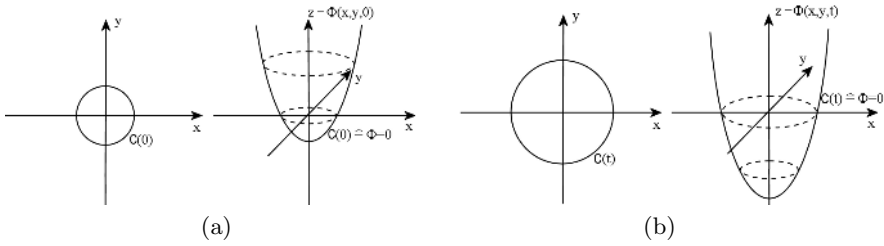
## 1 Introduction

Level set methods have become very popular in recent years. They have a wide range of applications in many different domains of science. Examples range from computer graphics[16], motion tracking[11], simulations of flame propagation[1] and compressible gas dynamics[9] to shortest path[3] and seismic travel time calculations[14].

Image processing is an important field for the application of level sets. Especially in image segmentation their implicit definition offers an alternative to the well-known explicit deformable models, like mass-spring-models[4] or finite element methods[8]. If the number of objects or the shape of the object that should be segmented is not known in advance, level set methods allow the definition of a model based on the properties of the desired object. This is also useful if the shape of the object has many degrees of freedom or has large variations between different data sets. Examples for their application in medical image analysis are the segmentation of the vascular tree or the bronchial tubes.

The drawback of level set methods is that artifacts distorting the data may pose a bigger problem to these methods than they do to explicit models. This is due to the data-driven nature of the front propagation process using level sets.

In this paper we propose a new force term for level set methods that allows to bridge gaps between parts of objects originating from missing information and thus to include another more model-based aspect to the definition of the speed function.



**Fig. 1.** Illustration of the front propagation process: The left image shows an initial curve  $C$  and the level set function  $\Phi$  at time  $t=0$ . The right image shows both functions at a later time.

The paper is structured as follows: In section 2 we give a short introduction to level set methods and the definition of the speed function needed for the propagation process. Our modification of the speed function is presented in section 3. We show some experimental results in section 4 and conclude this paper in section 5.

## 2 Level Sets and Speed Functions

### 2.1 An Introduction to the Level Set Method

Level sets were introduced by Osher and Sethian [10] for the solution of surface motion problems. They are used to describe the evolution of a curve  $C$  over time according to a given speed function  $F$ .

Let  $C \in \mathbb{R}^n$  be a parameterised closed curve and  $C(\mathbf{x})$  be the family of curves generated by the movement of the initial curve  $C_{t=0}(\mathbf{x})$  along its normal direction. The speed of this movement is a function based on various elements, e.g. the local curvature of  $C$ . To allow topological changes of the evolving front,  $C$  is embedded as a zero level set into a higher dimensional function  $\Phi \in \mathbb{R}^{n+1}$ .

$$C_{t=0}(\mathbf{x}) = \{\mathbf{x} | \Phi(\mathbf{x}, t = 0) = 0\}. \tag{1}$$

This leads to the level set equation

$$\Phi_t + F|\nabla\Phi| = 0, \tag{2}$$

where  $|\nabla\Phi|$  denotes the normalised gradients of the level set function and  $F$  is the speed function determining how fast the front moves.

The advantage of this representation is that  $\Phi$  always remains a function even if  $C$  splits, merges or forms sharp corners. Also, this representation is independent of the number of dimensions of  $C$ . As  $\Phi$  changes over time its zero level set  $\Phi(\mathbf{x}, t) = 0$  always yields the propagating front, i.e.  $C(\mathbf{x})$  at time  $t$ .

An example of this process is illustrated in Figure 1.

An extensive description of the level set method with all its aspects and mathematical background can be found in [13].

## 2.2 Speed Functions of Level Set Applications

As mentioned in the last section, the behaviour of the front propagation process depends on the definition of the speed function  $F$ . Since the focus of this paper is the application of level set methods for image segmentation, we will discuss aspects of the speed function adequate for this application.

In contrast to explicit models used for segmentation in image processing, level sets do not have to be initialised near the the desired object boundary (although this saves computational time and usually makes the segmentation process more robust). Instead the initial contour can be placed almost anywhere, but should be located either completely inside or outside the object to be segmented.

The easiest way a speed function can be defined is simply a constant inward or outward motion  $F_A$  (also called “advection”). To assure numerical stability at shocks (e.g. corners) and to prevent the front from crossing over itself, it is necessary to use upwind schemes [10] for the calculation of the advection term. Also, the time steps  $\Delta t$  have to be small enough to avoid numerical instabilities[13].

The inclusion of a curvature term is also useful for many applications (e.g. it is the basis for the use of Level Sets for the simulation of compressible gas dynamics [9]), and it allows the use of front propagation methods in image analysis where noisy data and incomplete object boundaries are a common problem.

The local curvature  $\kappa$  is defined

$$\kappa = \nabla \frac{\nabla \Phi}{|\nabla \Phi|}. \quad (3)$$

For  $C \in \mathbb{R}^2$  this results in

$$\kappa = -\frac{\Phi_{xx}\Phi_y^2 - 2\Phi_x\Phi_y\Phi_{xy} + \Phi_{yy}\Phi_x^2}{(\Phi_x^2 + \Phi_y^2)^{\frac{3}{2}}}. \quad (4)$$

Depending on the characteristics of the underlying data the curvature term should be weighted to adjust its influence on the propagation process. Therefore, we will refer to the regularisation term as

$$F_\kappa = \varepsilon \kappa, \quad (5)$$

with  $\varepsilon \geq 0$ . Modifications of the curvature term for the calculation of Min/Max-flow are described in [6] where Level Sets are used for image enhancement.

Both,  $F_A$  and  $F_\kappa$  are what in explicit models (e.g. [5]) would be called “internal forces”, that is, influences on the evolution of the curve that are purely model-based. A second group of forces is based on the data, therefore called “external forces”. We will give here some examples that are specific to image processing applications as they incorporate image features for the modification of the speed function. The calculation of these terms will be given for 2D data sets. However, their generalisation to higher dimensions is straightforward.

Malladi et. al.[7] proposed a speed term to stop the propagating front at image gradients. One way to define such a term is

$$F_{\nabla I}^{(1)}(x, y) = \frac{1}{1 + |\nabla G_\sigma * I(x, y)|}, \quad (6)$$

where  $(x,y)$  is a pixel of the given image and  $G_\sigma * I(x,y)$  denotes an image convolved with a Gaussian low pass filter with a standard deviation of  $\sigma$ . The new term is then incorporated into the speed function as a scalar. Therefore,

$$F = F_{\nabla I}(F_A + F_\kappa). \quad (7)$$

Another way to define this gradient-based speed term is

$$F_{\nabla I}^{(2)}(x,y) = e^{-|\nabla G_\sigma * I(x,y)|}. \quad (8)$$

This term results in smaller values near gradients and higher values in homogenous regions than given by  $F_{\nabla I}^{(1)}$ .

A modification of this gradient-based term was proposed by Caselles et. al.[2], who changed the speed function to

$$F = F_{\nabla I}(F_A + F_\kappa) + \nabla g \cdot \nabla \Phi, \quad (9)$$

with  $g(x,y) = -|\nabla(G_\sigma * I(x,y))|$ . The added term attracts the front towards edges even if it has already crossed over them. Obviously this makes the segmentation process more robust.

Other image features may be implemented in a similar fashion. For instance, an image-based speed term for a front that should just expand within a given grey value  $G_{seed}$  would be defined

$$F_G = \begin{cases} 1, & \text{if } I(x,y) = G_{seed}, \\ 0, & \text{otherwise.} \end{cases} \quad (10)$$

All image-based terms need to be combined and weighted in a meaningful fashion and may then be used the same way  $F_{\nabla I}$  was used in equation (7).

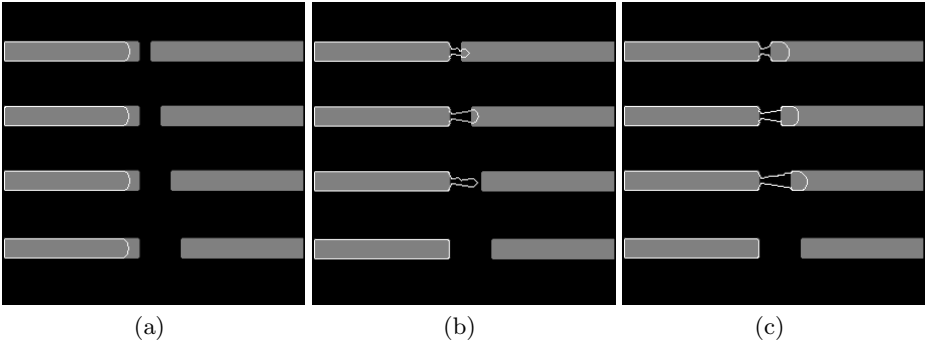
There is a third group of speed terms in literature that is neither purely model-based nor data-based. We will refer to those modifications of the speed function as “geometric terms”. The given examples work in 3D, their definition in higher dimensions is more complicated than with the speed terms given above.

Van Bemmelen et. al[15] define a “vesselness filter” for the segmentation of blood vessels. This function is based on the eigenvalues of the Hessian matrix at each pixel in the data set. The resulting speed term gives high values inside of cylindrical objects and low values otherwise.

A similar approach by Young et. al[8] uses a cylinder that is fitted into a volume at different angles. Here, too, grey values are used to calculate the fitting accuracy. This modification was also used for the segmentation of the vascular tree.

It is obvious that forces purely based on image features are often easy to define but have their limits with noisy data. More model-based aspects (like the mentioned “vesselness filter”) would benefit the segmentation process greatly but are usually limited to special applications. Also, the independence of the number of dimensions of the data the level set is used in is lost.

Therefore, we try to define a speed term that is somewhat weaker in its contribution to the speed function but consistent with the definition of level sets and not limited to a single application.



**Fig. 2.** The propagating front evolving under the new force term at three points in time. This “bridging force” is not used until the left sides of the bars are segmented. Then it lets the front leap to the right side of the bars where it continues to propagate normally. Note, that the front does not cross the gap in the bottom bar because the distance between both halves is too great.

### 3 Bridging Gaps in Data

The goal of this section is to define a speed term that allows the segmentation of objects whose representation in the data is disconnected. In medical imaging gaps in objects may occur due to partial volume effects when thin objects are seemingly disconnected due to a large pixel spacing in the data set. Other examples may be disconnected lines in drawings or, more generally, objects divided due to occlusions by other objects.

To solve this problem we define a new speed term that allows the propagating front to “look ahead”. That is, we need to incorporate a term that for each pixel on the front decides if it should be moved even if the underlying image features in the data (grey values, gradients, etc.) suggest that it should be stopped.

The implementation is very straightforward: we calculate the surface normal for each pixel on the front. This is trivial due to the definition of level sets and is computationally inexpensive. Knowing the direction of the normal we can choose distances at which to analyse if the underlying data favours further propagation of the level set at these locations. If the data indicates that the object continues in direction of the normal, an additional force  $F_B$  is added to the speed function  $F$ , changing it to

$$F = F_I(F_A + F_\kappa) + F_B(r, i). \quad (11)$$

Here,  $F_I$  denotes the combination of all image-based speed terms as described in the previous section. The two parameters  $r$  and  $i$  of the newly defined term  $F_B$  denote the interval of the surface normal of a pixel that is examined and the sampling of the pixels in that given interval, respectively. The second parameter is only introduced to save computational cost since it is usually not necessary to analyse every pixel along the normal. Figure 2 illustrates this process. For

convenience we will refer the the new speed term as “bridging force” in the remainder of this paper.

To avoid that the front leaks into the background at locations where a gap is bridged the other speed terms can be switched off. This would still connect both parts of the object but forms only very thin bridges as only very few pixels will propagate over a given gap. A better solution is an adequate definition of the other speed terms and to use e.g. curvature to avoid leaking. This strategy was used with the experiments in section 4.

This definition of the bridging force is partly image-based, as it uses image features, as well as model-based features. It assumes that regions within a certain range having the same properties belong to the same object.

Depending on the properties of the underlying data it may be difficult to define image features characteristic for a single object. The most simple feature, the grey value, may occur also in regions not belonging to the desired object.

A slight change in definition makes the bridging force purely model-based and more reliable: If the user is required to choose seed points in both parts of the disconnected object, the front approaches a gap in the representation from both sides. In this case the process of deciding if a gap should be bridged becomes much simpler because the algorithm can assert if a region is already segmented, that is, if the level set function  $\Phi$  has negative values on the examined locations.

Leaking does not pose a problem if curvature is used to keep the front smooth.

Finally, note that both definitions of the “bridging force” are independent of the dimensionality of the data just as the level set method itself and may thus be used for various applications without changing their calculation.

## 4 Experimental Results

We now want to present some segmentation results using the new speed term.

A series of test images was prepared to determine the abilities of the segmentation using the modified speed function.

We used two classes of test images containing a total number of 180 gaps with a size between 5 and 30 pixels: class 1 are grey value images with foreground objects in various intensities, class 2 are images with white objects on a black background. Both classes of images were tested with objects of different thickness. The original images were used as well as images with added gaussian noise in various magnitudes.

Both versions of the newly defined bridge force were applied to the 180 samples. The speed function of the level set also contained the gradient- and grey value based term as well as the curvature term defined in section 2. The parametrisation of the level set was kept constant for both versions of the new speed term. Segmentation results could be more exact with an adjusted parameter set for each image but results were also satisfactory with the global set (see table 1 for an overview of the results).

**Table 1.** Results from the application of the modified speed function to the test images. The numbers show how many of the existing gaps in the data were crossed successfully. “Image-based” denotes the image-based version of the bridging force, “Model-based” the model-based version. Class 1 are grey value images and Class 2 are black & white images. “Class 1 > 10” and “Class 2 > 10” denote the results, if the test images containing thin structures of less than 10 pixels are left out.

Data set	All images	Class 1	Class 1 > 10	Class 2	Class 2 > 10
Overall	92%	92.5%	97.5%	91%	100%
Image-based	87%	89%	95%	83.5%	100%
Model-based	97%	96.5%	100%	98.5%	100%

Overall, 92% of the gaps within the test images were bridged correctly. We experienced problems with thin structures of less than 10 pixels width. Due to curvature it is possible that at no front pixel the gradient is directed at the target object. This effect is getting more and more unlikely the thicker the object is at the gap. Leaving out structures with 10 or less pixels thickness, only four gaps were not bridged. Examples of test images are shown in figure 3.

In images with more than 30 % added noise the front sometimes leaked out at objects and gaps if the difference between grey values of object and background was too small (see figure 3(c), for instance). Higher weights for the image-based speed terms or the curvature may prevent this, but they also prevent the bridging force from crossing over gaps in the data and extremely slow down the whole propagation process.

We also applied a level set with the modified speed function to CT data for the segmentation of blood vessels. In [12] we already used level sets for the segmentation of the vascular tree in these data sets and experienced problems with unconnected parts of vessels due to partial volume effects. First tests with the new bridging force seem promising as the modified speed function allowed the level set to connect such vessels, as depicted in figure 5.

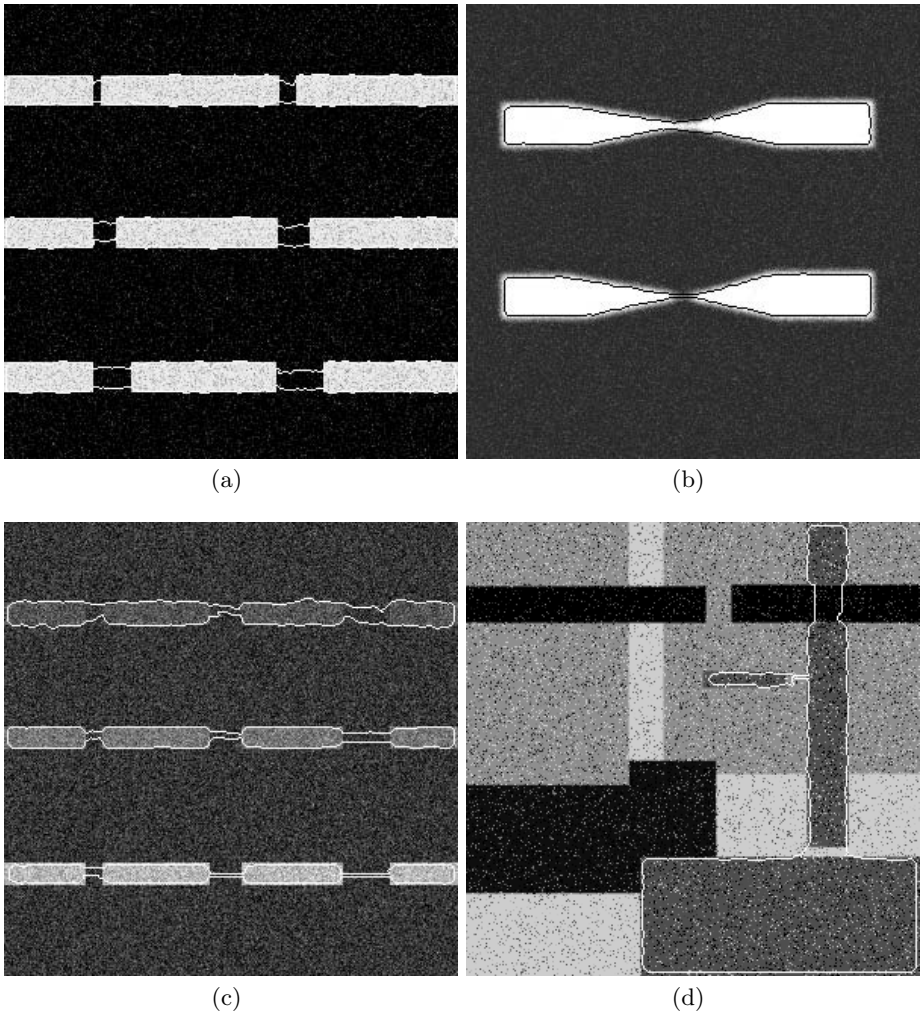
## 5 Conclusions

We presented two ways to define a new speed term for a level set speed function. An image-based speed term that allows the propagating front to bridge over gaps in the presentation of an object in the data and a model-based speed term to explicitly connect fronts within a certain range of each other.

Both versions of the modified speed function were successfully applied to various test images, as well as medical data and photographic images. The system proved to be robust to noise, with the model-based speed term connecting 100% of the gaps in test images containing objects of more than 10 pixels width.

Nevertheless there are various improvements to be made. Obviously, the problem of connecting thin structures has to be solved in the future. It may help to analyse a cone in front of each pixel, instead of just the normal direction. This

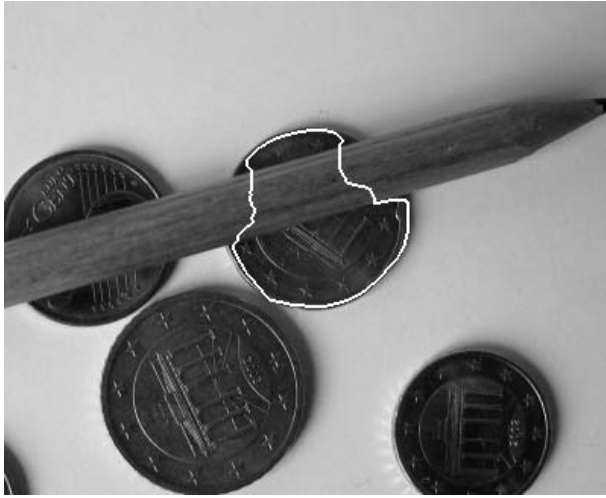




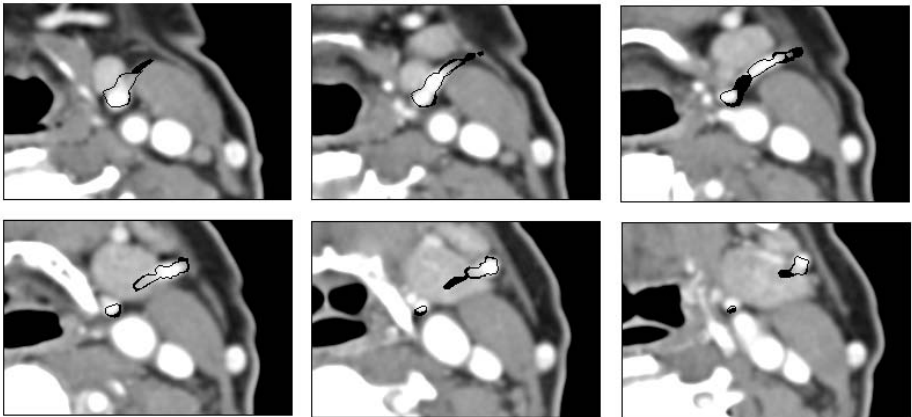
**Fig. 3.** Examples of segmentation results on test images. Images (a) and (b) are of class 1, images (c) and (d) are of class 2. To all images 33% gaussian noise was added.

will slow down the propagation process significantly, though, as it will become more difficult to analyse the data ahead of the front.

Another problem are u-shaped structures that should not attract themselves via a bridging force. For the model-based speed term this could be solved with minimal computational cost by assigning an identifier to each front. In this case, the new speed term would not be applied if fronts have the same identifier. The problem is more difficult with the image-based speed term and we will hopefully solve this problem in the future.



**Fig. 4.** Segmentation of a partially occluded object using the modified speed function. Note, that the “bridge” between both parts of the coin does neither try to approximate the occluded part of the coin nor leaks out into the occluding object. With the use of a minimum curvature speed term, as described in [6], even the occluded parts of the object should be approximated correctly.



**Fig. 5.** Application of the modified speed function to medical images for the segmentation of blood vessels. The images show six successive slices of a CT data set with a slice spacing of 3 mm. A vessel branches off at a steep angle and due to partial volume effects the lower and upper part of the vessel seem disconnected. This makes it impossible for a level set without the additional bridging force to connect both parts of the vessel. Using our additional speed term, the level set segments the vessel correctly.

## References

1. Abu-Gharbieh, R., Gustavsson, T., Kaminski, C., Hamarneh, G.: Flame Front Matching and Tracking in PLIF Images using Geodesic Paths and Level Sets. *IEEE Int. Conf. on Comp. Vis., Workshop on Variational and Level Set Methods in Comp. Vis.* (2001) 112–118.
2. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic Active Contours. *Int. J. Comp. Vis.* **22**(1) (1991) 61–79
3. Deschamps, T.: Curve and Shape Extraction with Minimal Path and Level-Set techniques. Applications to 3D Medical Imaging. Ph.D. thesis. L'Université de Paris-Dauphine. (2001)
4. Hamarneh, G., McInerney, T., Terzopoulos, D.: Deformable Organisms for Automatic Medical Image Analysis. *MICCAI 2001.* 66–76
5. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active Contour Models. *Int. J. Comp. Vis.* **1**(4) (1988) 321–331
6. Malladi, R., Sethian, J. A.: Image Processing: Flows under Min/Max Curvature and Mean Curvature. *Graph. Mod. and Image Proc.* **58**(2) (1996) 127–141
7. Malladi, R., Sethian, J. A., Vemuri, B.: Shape Modelling with Front Propagation: A Level Set Approach. *IEEE Trans. on PAMI.* **17**(2) (1995) 158–175
8. McInerney, T., Terzopoulos, D.: A Dynamic Finite Element Surface Model for Segmentation and Tracking in Multidimensional Medical Images with Application to Cardiac 4D Image Analysis. *Comput. Med. Imaging Graph.*, **19**(1) (1995) 69–83
9. Mulder, W., Osher, S. J., Sethian, J. A.: Computing Interface Motion in Compressible Gas Dynamics. *J. Comp. Phys.* **100** (1992) 209–228
10. Osher S., Sethian, J. A.: Fronts Propagating with Curvature Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulation. *J. Comp. Phys.* **79** (1988) 12–49
11. Paragios, N., Deriche, R.: Geodesic Active Contours and Level Sets for the Detection and Tracking of Moving Objects. *IEEE Trans. on PAMI.* **22**(3) (2000) 266–280
12. Rink, K., Törsel, A. M., Tönnies, K.: Segmentation of the Vascular Tree in CT Data using Implicit Active Contours. *Bildverarbeitung für die Medizin 2006.* 136–140
13. Sethian, J. A.: *Level Set Methods and Fast Marching Methods.* Cambridge University Press. (1999)
14. Sethian, J. A., Popovici, A. M.: 3-D travelttime computation using the fast marching method. *Geophysics.* **64**(2) (1999) 516–523
15. Van Bommel, C. M., et. al.: A Level-Set-Based Artery-Vein Separation in Blood-Pool Agent CR-MR Angiograms. *IEEE Trans. Med. Imag.* **22**(10) (2003) 1224–1234
16. Whitaker, R. T.: Isosurfaces and Level-Set Surface Models. Technical Report UUCS-02-010. School of Computing, Univ of Utah. (2002)
17. Young, S., Pekar, V., Weese, J.: Vessel Segmentation for Visualization of MRA with Blood Pool Contrast Agent. *MICCAI 2001.* 491–498

# Generation and Initialization of Stable 3D Mass-Spring Models for the Segmentation of the Thyroid Cartilage

Jana Dornheim<sup>1</sup>, Lars Dornheim<sup>1</sup>, Bernhard Preim<sup>1</sup>, Ilka Hertel<sup>2</sup>,  
and Gero Strauss<sup>2</sup>

<sup>1</sup> Otto-von-Guericke-Universität, Postfach 4120, D-39106 Magdeburg, Germany

<sup>2</sup> Hals-Nasen-Ohren-Universitätsklinik, Universitätsklinikum Leipzig, Liebigstr. 18a,  
D-04103 Leipzig, Germany

**Abstract.** The preoperative planning of primary tumor resections in the larynx region shall be supported by a 3D visualization of the patient-specific anatomy and pathological situation. This requires a segmentation of the larynx cartilage structures from computed tomography (CT) datasets.

In our work, we use 3D Stable Mass-Spring Models (SMSMs) for this segmentation task. Thereto, we create a specific 3D deformable shape model for the segmentation of the thyroid cartilage. A new concept for elastic initialization of this model is presented, allowing the deformable model to adapt specifically to patient-specific shape variations and pathological deformations.

We show that using our generation and initialization method, prototypical 3D deformable shape models can be adapted to very different patients without any prior training and prior knowledge about new patients' data.

## 1 Introduction

In the case of tumor affections in the larynx and lower hypopharynx, the patient's life expectancy and further life quality depend strongly on the required surgical treatment. The parts of the larynx which need to be resected, determine the patient's ability to breathe, swallow and speak. For the decision on a surgical strategy, the extent of the tumor must be evaluated with respect to infiltration of the following structures:

- the vocal chords and muscles (usually judged by laryngoscopy),
- the glottis, subglottic and supraglottic space, and
- the larynx cartilages, in particular the epiglottis, thyroid cartilage, cricoid cartilage, and the two arytenoid cartilages (often judged by CT [1]).

For the assessment of the air and cartilage structures, a 3D visualization of the patient-specific anatomy and pathological situation is desirable to reduce uncertainties in the chosen surgical procedure. This requires a precise segmentation of

the larynx and its substructures from neck CT datasets. These images are very rich of different small structures of high signal intensity (Figure 4 in the evaluation section gives an impression) making any segmentation task very difficult.

The inhomogenous nature of the cartilage itself makes its segmentation a challenging task, for which neither simple edge-based techniques, such as LiveWire, nor gray-value-based segmentation techniques are appropriate. Since profound anatomical knowledge is needed to bridge areas of weak signal in the cartilage wall, we target at a 3D model-based segmentation of the thyroid cartilage.

## 2 State of the Art

For the segmentation of the thyroid cartilage, no specific approaches exist to our knowledge. General methods for the segmentation using 3D deformable models are known [2]: *3D Active Contours* or *Balloons* [3] incorporate rough shape knowledge by means of a viscosity condition. The use of an inflation force prevents them from shape collapse and drives them towards the target structure's contour. This global representation of shape does however not allow to model complex shape information. *Implicit 3D deformable models* [4] do not bear the problem of instability and need no inflation force. However, they are restricted to describing regular geometric shapes, that can be described by a simple equation. *Active Shape Models (ASM)* and *Active Appearance Models (AAM)* [5] provide support for segmenting more complex shapes by means of a statistical analysis of training data. They require large amounts of training data and a very good correspondance of 3D points, which makes model creation laborious and segmentation results potentially imprecise. For the segmentation of pathological shape variations (e.g. caused by a tumor), ASMs and AAMs are principally inappropriate, because these shape variations are very individual and cannot be trained.

*Stable Mass-Spring Models (SMSMs, [6])* are prototypical 3D shape models, that need no training, but an initial model describing the expected shape. These models are especially appropriate for tracking and searching as well as for segmentation, if the individual structure is known in general. They are very robust to noise and gaps in the data, as [7] show for the segmentation of the left ventricle in 3D SPECT.

In our application, such specific deformations need to be modeled. Therefore it is not enough to create an SMSM like in [8] that prototypically models the target structure. A further adaption to the patient-specific pathological shape variation is necessary, which will be introduced in this work. This way, a segmentation model is created, that is directly tailored to an individual patient and does not represent unnecessary shape knowledge about other patient's like statistical models would.

## 3 Method

In our work, we construct an SMSM of the thyroid cartilage semi-automatically from a manual sample segmentation. We thereto adapt and refine the model

creation (section 4) introduced in [8], which proposes a model topology consisting of two parts created independantly from each-other:

1. an (outer) surface submodel containing masses with *gradient sensors* and the contour faces representing the modeled object contour, and
2. an (inner) volumetric submodel containing *intensity sensors*.

Both submodels are connected afterwards to an overall model.

To adapt this prototypical model to each individual patient’s data, an elastic initialization technique is introduced (section 5.1) to translate, rotate, scale and deform the constructed general model nonlinearly and model-consistently to fit the segmentation target structure by means of key masses. Because of these starting conditions, the segmentation of the thyroid cartilage becomes possible in this difficult data.

## 4 Model Generation for the Thyroid Cartilage

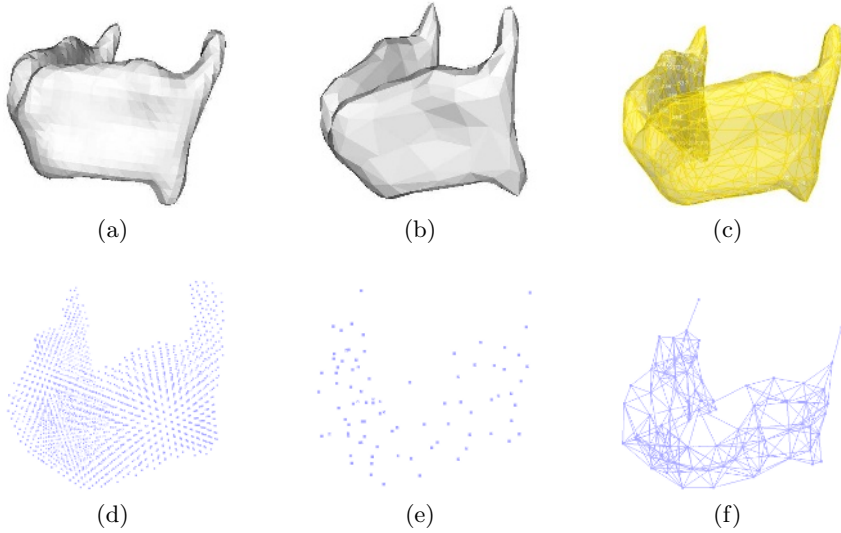
The semi-automatic model generation for the thyroid cartilage is based on a manual segmentation created from a dataset with a visually average-shaped larynx. The sample segmentation is available as a binary volume dataset. For our examination, two different models were created and evaluated:

1. One *volumetric* model, according to the model creation of [8], consisting of a surface submodel and a volumetric submodel connected by 1:1 connections.
2. For comparison, a pure *surface model* is created, consisting of the surface submodel only. This model was employed to evaluate whether segmentation based on edge detection alone is more appropriate than using gray value information.

### 4.1 Generation of the Surface Model

The sample segmentation was resampled down to an isotropic voxel size for efficient model generation. From this resampled segmentation, an isosurface was generated using the marching cubes method (Fig. 1(a)), smoothed and simplified with Quadric Error Metrics [9] to a certain degree (Fig. 1(b)), that is adjustable for different abstraction levels. The resulting number of triangles (in our case 50–200) provides an appropriate modelling of the cartilage shape while still allowing for realtime model simulation (Fig. 1(c)).

The resulting surface was used to create the surface model: For each vertex of the mesh, a mass point was created. For each edge in the mesh, a spring connecting the incident vertices, resp. mass points, was created. All masses and spring constants throughout the model were set to 1.0. A *direction-weighted gradient sensor* [10] was attached to each mass point of the surface model, ensuring that contour masses are only attracted by image gradients of the same direction as the incident surface normal. This prevents the model from being distracted by neighbouring but irrelevant gradients, which is a common problem in neck CT datasets.



**Fig. 1.** Stages of model creation (surface submodel: (a)-(c), volumetric submodel: (d) - (f)) for the thyroid cartilage: **(a)** Isosurface, **(b)** Surface simplification, **(c)** Surface submodel, **(d)** Mass point initialization, **(e)** Mass point reduction, **(f)** Volumetric submodel

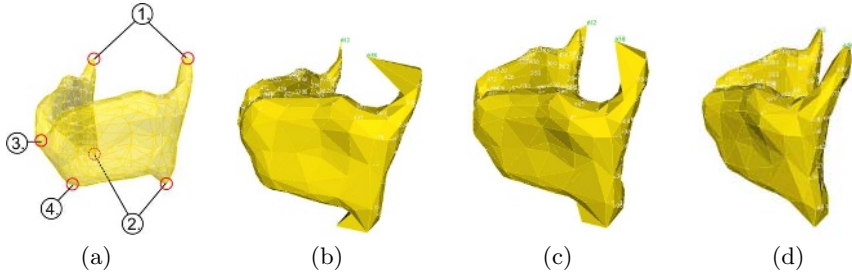
## 4.2 Generation of the Volumetric Model

For the generation of the volumetric model, inner masses with attached *intensity sensors*, dragging these masses towards neighbouring voxels of a certain gray value interval, have to be created and combined with the surface submodel. Thereto, an initial set of mass points is created by placing one mass point at each voxel of the resampled manual segmentation (Fig. 1(d)). Then, the initial point set is reduced iteratively by the following relaxation.

**Reduction of Inner Masses.** For each mass point, all mass points inside a neighbourhood of radius  $r$  are moved to their common center of mass and merged. By iterating this relaxation, the initial point set is reduced considerably, and fills the manual segmentation evenly (Fig. 1(e)).

The convergence of the relaxation towards a reasonable point set depends on the choice of  $r$ . According to our experiments, a radius of half the desired minimum distance of two mass points leads to a convergence representing the original shape well. A dense placement of the volume masses has the advantage that the inner properties of the segmentation target structures are measured at more positions, which is equivalent to a higher sampling rate. We therefore always chose  $r$  to be within  $[\text{voxelsize}; \sqrt{2} \cdot \text{voxelsize}]$ .

**Cross-Linking of the Inner Masses.** Each of the resulting volume mass points is linked with each neighbouring mass point within a user-defined radius



**Fig. 2.** Elastic model initialization: (a) Key masses, (b) Key masses placed, (c) Model during adaptation to key positions, (d) Model after adaptation to key positions

$p$ . In our model of the thyroid cartilage, a radius of  $p = 10\text{mm}$  (for a voxel size of 2.148 mm) led to good results in all cases (Fig. 1(f)).

### 4.3 Connecting Inner and Outer Submodel

Both the volume and the surface submodel are interconnected by springs and merged to one volumetric model. A 1:1 interconnection (each point of the surface submodel is connected to the closest mass point of the volume submodel) has proved to be appropriate for good model stability.

## 5 Segmentation Process

### 5.1 Elastic, Model-Consistent Initialization

A good initial adjustment of the deformable model to the individual patient’s larynx shape is needed, so that the adaptation of the model to the dataset will not be distracted by ‘wrong’ gray value information of adjacent structures.

For this initialization, the model’s position, rotation, scaling and expected shape have to be adjusted for our application. Classical initialization methods correcting only the model’s position, rotation and/or scaling are not sufficient according to our tests (see section 6.2 for details).

We therefore introduce a new initialization method, which exploits the model specification already available: In the created deformable model, a mass point at each of the most prominent landmarks is marked as a *key mass* at the end of the model creation process. The user can specify the positions of these key masses by clicking into the dataset. The 6 key masses for the model of the thyroid cartilage are positioned at the *cornu superius left* and *right* (1.), *cornu inferius left* and *right* (2.), as well as the *upper (Adam’s apple)* (3.) and *lower end of the larynx front side* (4.) (Fig. 2(a)).

These key masses are then fixed and the model simulation is started with only the spring and torsion forces active, but all sensor input turned off<sup>1</sup>. The

<sup>1</sup> Simulation parameters: spring force weighting  $w_f = 5.0$ , torsion force weighting  $w_t = 10.0$ , damping factor  $d = 0.001$ , simulation time step  $\Delta t = 0.05$ .



internal forces, normally representing the model’s shape knowledge during a regular segmentation, adapt the model’s shape to the key positions (Fig. 2(b)-(d)), while keeping the model as consistent as possible to the shape knowledge it represents. With this method, the model adjusts itself flexibly and nonlinearly to the specified key positions. Guided by the key masses, it adjusts position, size, orientation and shape during this process to the individual patient’s anatomy.

After complete adaptation, the rest lengths and rest directions of all springs are set to their current (deformed) length and direction values. This way, the new expected shape is anchored in the model.

## 5.2 Model Adaptation

After initialization, the precise model adaptation is started<sup>2</sup>. All key masses are left fixed, so that the model adaptation occurs within their frame of reference. This way, the model is kept at the correct position. Besides, the lengths of the cornu superius and inferius vary widely among different patients. By keeping the top of the cornu fixed, we can ensure that the whole cornu is found. The simulation is stopped, when the model speed falls below a certain threshold, which always happens because of the damping.

# 6 Evaluation

## 6.1 Data Material and Ground Truth

12 CT datasets of the neck were acquired for preoperative planning, containing the larynx. The slice thickness of the datasets ranged from 1.5 mm to 6.0 mm. The datasets varied significantly w.r.t. signal-to-noise ratio, contrast and motion artifacts. In 3 datasets, the larynx was displaced or partially destroyed due to tumor affection. On all 12 datasets, a manual segmentation of the thyroid cartilage was created by an experienced user and controlled by a radiologist. These verified manual segmentations were used as a ground truth for the evaluation.

## 6.2 Model Initialization

To evaluate the single effect of our elastic initialization method from section 5.1, we compare it to the classical initialization methods of positioning and positioning with independent scaling for each axis, where always the optimal initialization is computed with regard to the ground truth. Rotation correction did not make sense here, since the datasets have because of the same imaging process all the same principal direction.

---

<sup>2</sup> Simulation parameters: sensor force weighting  $w_s = 0.05$  for gradient sensors,  $w_s = 0.001$  for intensity sensors, spring force weighting  $w_f = 1.0$ , torsion force weighting  $w_t = 2.0$ , damping factor  $d = 0.001$ , simulation time step  $\Delta t = 0.05$ .

**Table 1.** Average initialization results of the standard initialization methods (position and position / scale) and our elastic, model-consistent initialization method compared to the ground truth, measured for 11 CT datasets of the neck using a model created on the 12th dataset (leave-one-out-test)

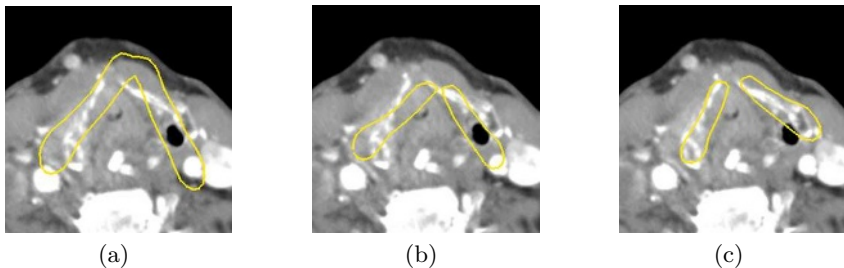
Evaluation Measure	Position	Position / Scale	Elastic
Hausdorff Distance	30.54 mm	29.16 mm	20.06 mm
Average Distance	4.73 mm	4.09 mm	2.90 mm

In the optimal case, a “perfect” model initialization would be equal to the ground truth. We therefore calculated the shape (border) distances (Hausdorff and average distance) of both classical initialization methods and our elastic, model-consistent initialization method to the ground truth for evaluation. Table 1 shows, that the new elastic, model-based initialization technique places the model roughly 30 % - 40 % closer to the ideal segmentation result than the classical initialization methods. This is an important improvement for every segmentation method using a local search technique, such as the SMSM approach used for our application.

Furthermore, these numbers show, that the elastically initialized model’s shape approximates already the individual shape of the segmentation target structure for a single patient. Otherwise, the shape distances from table 1 would not be so much lower than the ones from the classic initialization methods, which optimally match the ground truth using models without deformation allowed, but only scaling (Fig. 3 illustrates this fact).

### 6.3 Segmentation Experiments

From a dataset with an average-shaped larynx, a volumetric model (consisting of a surface and a volumetric submodel) and a pure surface model were generated as described above (section 3). The key masses were marked manually and used throughout all experiments. The two models were then applied to the remaining 11 datasets (leave-one-out-test) in the following manner:



**Fig. 3.** Enhancement by elastic model initialization: (a) Manual initialization using only translation, (b) Manual initialization using translation and scaling, (c) Elastic initialization with 6 landmarks

**Table 2.** Average segmentation results of the 2 models and 1 experienced user compared with the ground truth, measured for 11 CT datasets of the neck

Evaluation Measure	Volumetric model	Surface model
Hausdorff Distance	11.11 mm	9.84 mm
Average Distance	1.20 mm	1.06 mm

1. The user marks the key positions in the dataset (6 markers).
2. The model is automatically positioned and scaled according to the bounding box of the key positions.
3. The model is adapted to the key positions (1 click for stopping this phase).
4. The newly adapted shape is automatically learned by the model.
5. The model adaptation to the dataset is performed with the key masses still fixed. (1 click for model stopping).

The segmentation results for both models, as well as the manual segmentation results of an experienced user were compared with the given ground truth by different evaluation measures (Tab. 2).

## 6.4 Results

In all 11 datasets, the thyroid cartilage could be robustly segmented with an average border distance of 1.064 mm to the ground truth. No significant loss of segmentation quality could be found in the cases of pathological larynx shapes (Fig. 4(a)). Weak-signal holes in the cartilage were successfully bridged by the model’s intrinsic shape knowledge (Fig. 4(b)). The model adaptation time needed for elastic initialization was 0.5–1.5 minutes for all datasets, the model adaptation to the datasets took 2–4 minutes per dataset (all measures performed on a standard PC: Pentium M, 1,7 GHz, 512 MB RAM).

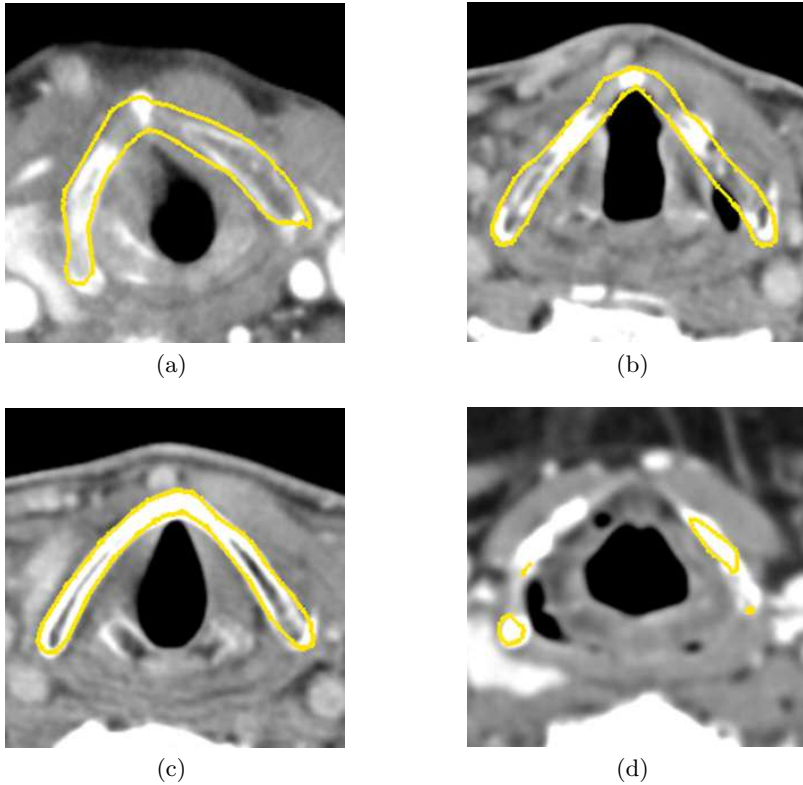
Our results show, that the volumetric model is not superior to the pure surface model. In fact, with intensity sensor weighting  $w_s > 0.001$ , the intensity sensors tend to be attracted by false gray values in neighbouring structures. This leads to strong model instability and significantly worse segmentation results. We therefore recommend using a pure surface model for the segmentation of the thyroid cartilage, which will be less affected by gray value inhomogeneities.

The model’s adaptation to the datasets is significantly better in the lower part of the thyroid cartilage (i.e. below the adam’s apple) than near the upper border, which may cause up to 50 % of the observed undersegmentation. This can be attributed to several circumstances:

- The lower part of the thyroid cartilage tends to be signal-intensive, while the model often lacks signal support in the upper part. This may lead to strong undersegmentation of this area (Fig. 4(d)). This causes the relative high values the Hausdorff distance. A simple user interaction might prevent this behaviour.

- In some cases where the hyoid bone is very close to the thyroid cartilage, the upper sensors of the model are attracted by its strong signal. In order to prevent leaking, the os hyoideum may be subtracted from the dataset first. Another possibility is to integrate the hyoid bone into the model to ensure that the masses for the thyroid cartilage are kept at appropriate distance.

Except for the hyoid bone, no other neighbouring structures distracted the model. This must be attributed to the new method for initialization followed by shape learning. Without these techniques, the thyroid cartilage could not be separated from the thyroid gland, blood vessels and the trachea robustly.



**Fig. 4.** Results of the model adaptation to the datasets

## 7 Discussion

A deformable 3D model (Stable Mass-Spring Model) has been constructed and adapted for the segmentation of the inhomogeneous and complex-shaped thyroid cartilage. We introduced model-consistent (position, orientation, rotation and

shape) adaptability of the model to individual patient shape variations by means of an elastic initialization. In contrast to statistical shape models however, our method is not limited to a pre-learned range of shape variations. Instead, it is always initialized to represent the shape information it needs by means of a few key masses. This makes it especially suited for segmenting pathological structures.

Compared with a manual or LiveWire segmentation, the model offers a drastic reduction of interaction effort. Already now, the model can be used at least as a presegmentation of the cartilage, which needs only be corrected at 2–3 positions by the user. In contrast to other 3D models, such as implicit models and ASMs, interaction is intuitively supported by the explicit shape representation of our model.

## References

1. Myers, E.N., ed. In: *Operative Otolaryngology: Head and Neck Surgery*. Volume 1. W. B. Saunders Company (1997) 403–443
2. McInerney, T., Terzopoulos, D.: Deformable models in medical image analysis: A survey. *Medical Image Analysis* **1** (1996) 91–108
3. Cohen, I., Cohen, L.D., Ayache, N.: Using deformable surfaces to segment 3d images and infer differential structures. *CVGIP: Image Understanding* **56** (1992) 242–263
4. Bardinet, E., Cohen, L.D., Ayache, N.: A parametric deformable model to fit unstructured 3D data. *CVIU* **71** (1998) 39–54
5. Cootes, T., Edwards, G., Taylor, C.: Comparing active shape models with active appearance models. In: *BMVC*. (1999) 173–182
6. Dornheim, L., Tönnies, K.D., Dornheim, J.: Stable dynamic 3D shape models. In: *ICIP*. (2005) III–1276–1279
7. Dornheim, L., Tönnies, K.D., Dixon, K.: Automatic segmentation of the left ventricle in 3D SPECT data by registration with a dynamic anatomic model. In: *MICCAI*. (2005) 335–342
8. Dornheim, L., Dornheim, J., Tönnies, K.D.: Automatic generation of dynamic 3d models for medical segmentation tasks. In: *SPIE: Medical Imaging*. (2006)
9. Garland, M., Heckbert, P.S.: Surface simplification using quadric error metrics. In: *SIGGRAPH*. (1997) 209–216
10. Dornheim, L., Dornheim, J., Seim, H., Tönnies, K.D.: Aktive Sensoren: Kontextbasierte Filterung von Merkmalen zur modellbasierten Segmentierung. In: *Bildverarbeitung für die Medizin*. (2006)

# Preserving Topological Information in the Windowed Hough Transform for Rectangle Extraction

Dan Cireșan and Dana Damian

“Politehnica” University of Timișoara, Romania  
Faculty of Automation and Computers  
{cdanc, dd3593}@cs.utt.ro

**Abstract.** We present a new method for extracting rectangular shapes from images. It uses a windowed Hough transform and adds a new coordinate to store the precise pixel distribution of a line by means of a topological relation. By an early and rigorous check of each edge candidate, performed in the new expanded Hough space, the edge space is significantly reduced, thus simplifying further processing. Moreover, the edge checking algorithm provides flexibility in choosing the diameter of the circular search window. The method is robust, revealing a good recognition quality when applied to both synthetic and real images.

## 1 Introduction

Many elementary man-made structures have, when projected onto a plane, a rectangular form, making it imperative to develop accurate techniques that manipulate these types of shapes. These techniques cover a wide spectrum of environmental, medical and industrial applications: building recognition from satellite images [12], vehicle recognition [18]; determining the atomic resolution structure of the enormous number of proteins encoded in genomes [17,2], analyzing cryo-electron microscopy and simple particle images [19]; and recognizing patterns in robotic vision. For this reason, a wide range of algorithms have been developed to manipulate the large amount of acquired data. The major techniques in rectangle detection rely on matching criteria and line analysis.

Rectangular building extraction from stereo aerial images is approached in [13] by hierarchical perceptual grouping and matching, with subsequent verification of real shadows and building walls. Another semantic outlook over the feature extraction task can be found in [11], where perceptual grouping is used, as well as graph structures. The latter is formulated as a function of the type and values of connections between lines, followed by a step in which building hypothesis are generated by means of mathematical and geometrical relations. The above-mentioned approaches make use mainly of high-level feature analysis, but low-level analysis is also a sustainable method in the detection of rectilinear organizations as a first step toward the high-level one, or as an interleaved operation. In this area, the Hough transform [9] has been exploited in a variety

of ways. Pioneering work has been done by Ballard [1], which proposes a generalization of the Hough transform. The algorithm uses directional information of an edge image to construct a geometrical transformation from the image space to Hough space and eventually to detect arbitrary shapes, specifically lines, circles and parabolas. Several more variations of the transform are known, like Randomized HT [16], Probabilistic HT [8], Hierarchical HT[14]. Combining the variational methods of the Generalized HT and the Randomized HT, [7] addresses the problem of object detection given a template that resides with the sought shape, and exploits the advantages of both techniques.

In [6] a different extension is introduced to the Generalized HT, relying on a composition of weak affine transformations. An additional contribution is the incorporation of a local shape variability to improve the detection.

In [3], regular polygon detection is approached by the use of a five-dimensional space that is collapsed to a three-dimensional space considering an a posteriori probability. Also, the continuous log-likelihood of the probability density function of regular polygons is defined. Including additional a priori information, a real-time road sign detector is constructed. Jung and Schramm [10] describe a holistic approach that directly detects the rectangles from the Hough accumulator. The rectangle in the image space is sought with a small window shaped according to the rectangle dimension that slides above the image, and thus, using the Hough transform, a rectangle centered in the middle of the window can be detected. When the window size is only slightly greater than the rectangle diagonal, the method offers positive results.

The present work proposes a new method for rectangle detection by extending the Hough space with a new dimension to account for the way edge points spread according to their position on a detected line. Our method offers a higher accuracy of rectangle recognition and exhibits robustness on a variety of images.

## 2 Preservation of Topological Information

The classical Hough Transform is an efficient technique for line detection [5]. Undergoing a simple geometrical transformation,  $\rho = x \cos \theta - y \sin \theta$ , pixels in the image can be classified according to the line they belong to, in the cell of the Hough accumulation matrix. For complex images, individualizing the objects by means of a sliding window, as shown in [10], is vital for accurate detection. The window acts like a delimitation of the focused object from the background objects and reduces the uncontrolled influence over the values of the accumulator space cells correspondent to the currently analyzed area. However, if the distortion attributable to noise surpasses a certain threshold assertion, failing cases of detection occur. The main reason is that the accumulator configuration does not preserve the information about edge points position within the image, thus hindering the classification of objects. By preserving the points topology in the Hough space, we improve extraction with fewer exceptions and limitations.

### 2.1 Hough Transform Columns

The proposed algorithm adds a third dimension to the Hough plane, ensuring a bijective correspondence between a detected line and the new dimension, by means of their Hough parametric definition. We will name this attached dimension a column, due to its suggestive geometrical associations. Each column will hold the exact distribution of the edge points along the line, facilitating segment detection and the discrimination between valuable lines and noise. Let us consider Fig. 1a. It depicts the search region within the image space – a corona with boundaries  $D_{min}$  and  $D_{max}$ , which stand for the minimum edge of a sought rectangle and its diagonal, respectively. Point  $P(x, y)$  lies on the line  $d$  to which a Hough transform [9], assigns a  $(\Theta, \rho)$  coordinate in a unique manner. In order

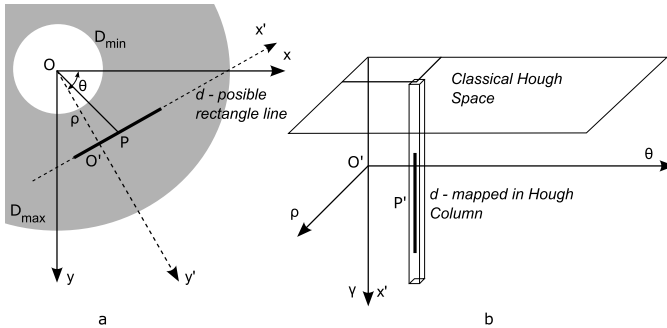


Fig. 1. Point P mapped from the image space (a) in a  $\gamma$ -Column (b)

to map every point  $P$  into a Hough space column as illustrated in Fig. 1b and preserve the same line context organization, a natural choice is to use as a reference point the foot  $O'$  of the perpendicular from the vertex  $O$  to the  $(\Theta, \rho)$  line. This yields a change of coordinate system such that  $P(x', y') = P(\gamma, 0)$ . The mapping between the two metric spaces is attained by means of an isometry, namely a roto-translation of parameters  $\Theta$  and  $\rho$ , respectively, where the topological relation is  $\gamma = x \sin(\Theta) + y \cos(\Theta)$ . This way,  $O'$  will encode the middle of the column,  $\gamma_{O'} = 0$ . When mapping the edge points in the Hough parameter space as a voting consequence, they will not only accumulate, but also linearly distribute along a  $(\Theta, \rho)$   $\gamma$ -column, representing the line that a point lies on. Hence, the procedure will literally import the lines from the original space into columns, according to their  $(\Theta, \rho)$  identification tuple.

Cloning the information from the original space to the Hough space provides a compact framework for analysis operations entirely in the Hough space. This eliminates the repeated switches between spaces when computing the correspondence between peaks in the accumulator cells and lines in the original image.



## 2.2 Sifting the Candidate Line Segments

After the columns of the Hough space are exhaustively populated, the next step is to perform a sifting of line segments which respect a predefined threshold of contiguity. Our implementation keeps track of the gaps in a similar manner to Song and Cai [15] who performed segment line extraction, taking account of segment discontinuities in order to precisely reconstruct the lines in the original space. Prior to this step, we ensure that no line that consists of a number of points less than the minimum allowed, a fraction of  $D_{min}$ , passes to the next analysis. It follows from the geometry constraints that the point  $O'$  would eventually be the center of one rectangle side. Since the generic origin  $O'$  coincides with the center of symmetry of any chord stretched inside the corona, which is also the maximum-length support of any possible segment, we can exploit the symmetry property of a rectangle side with respect to this origin. The algorithm below sets out in the center  $O'$  and scans the chord in both directions (step 1, step 4). Here, a single case of crossing is presented, the other one is analogous.

It accumulates the total number of valuable pixels on its way (step 5) and tolerates gaps lower than the threshold  $gap$  (step 3). It truncates both sides of the chord remainder at the level at which the gap - either  $gap_{up}$  or  $gap_{down}$  - that has reached the maximum acceptance rate began.

1. for  $l = 1$  to  $D_{max}/2$
2. if  $H[\Theta, \rho, D_{max}/2 - l] = 0$  and  $gap_{up} = gap$
3.  $gap_{up} \leftarrow gap_{up} + 1$ ; break
4. if  $H[\Theta, \rho, D_{max}/2 - l] \neq 0$
5.  $points \leftarrow points + 1$ ;  $gap_{up} = 0$
6. else  $gap_{up} \leftarrow gap_{up} + 1$

The case when a gap occurs in the center of symmetry is treated similarly.

At the end, a comparison is carried out between the number of filled pixels currently quantized and the actual length of the acquired edge denoted as  $edgelen\theta$ . We have experimentally chosen  $threshold_p = 0.6 * D_{min}$ , to check whether the current edge is at least the minimum edge length,  $D_{min}$  (step 1). Another fractional voting,  $threshold_e = 0.4 * edgelen\theta$  (step 3), validates the compliance of a number of pixels adequate for a rectangle edge.

1. if  $points > threshold_p$
2.  $edgelen\theta \leftarrow 2 * [l - \max(gap_{down}, gap_{up})] + 1$
3. if  $points \geq threshold_e$  return  $edgelen\theta$  else return 0

The outcome is a line segment, with negligible interruptions, which fulfills the symmetry quality, and is a good candidate for a rectangle side. The method brings conclusive advantages like an early reduction of residual noise and a significant decrease of the requested computational and storage resources as a consequence of a diminished rectangle search space.

### 2.3 Assembling the Rectangle

In the previous step, a screening filter is used to classify an amount of bounded lines into candidates for rectangle edges. Jung and Schramm [10] collected evidence for rectangle feature extraction from a Hough space, seeking specific relations that match a set of geometrical constraints of a rectangle. For our approach, we have modified their set of rules as discussed below.

For a given rectangle whose center coincides with the Cartesian origin, let the four vertices be  $V_1, V_2, V_3, V_4$ , where  $V_1V_4 = V_2V_3$  and  $V_1V_2 = V_3V_4$ . The Hough transform will map the aforementioned edges into peaks  $H_3(\Theta_3, \rho_3)$ ,  $H_4(\Theta_4, \rho_4)$ ,  $H_1(\Theta_1, \rho_1)$ ,  $H_2(\Theta_2, \rho_2)$ , respectively as in Figure 2.

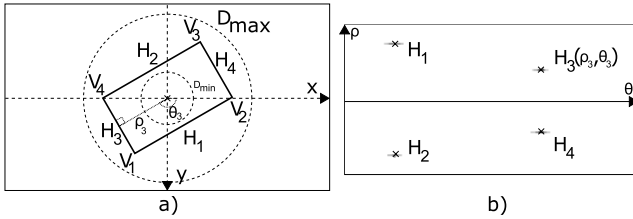


Fig. 2. Windowed Hough Space

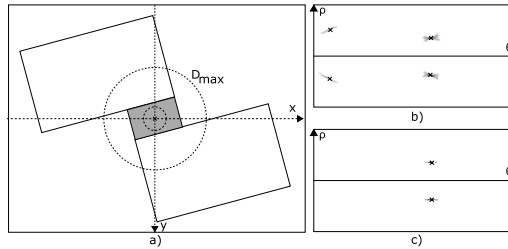
Peaks that design a rectangle shall obey the following rules:

1.  $|\Theta_1 - \Theta_2| < \varepsilon_\Theta$  and  $|\Theta_3 - \Theta_4| < \varepsilon_\Theta$ , designating the parallelism between two opposite rectangle sides, i.e. the pairs  $(H_1, H_2)$  and  $(H_3, H_4)$  respectively.
2.  $|\rho_1 + \rho_2| < \varepsilon_\rho$  and  $|\rho_3 + \rho_4| < \varepsilon_\rho$ , meaning two opposite rectangle sides should lie equidistant with respect to the considered center of the search corona; if these two first rules are satisfied, each of the two pairs will be called a peak pair, and the validation proceeds further.
3.  $valp((\Theta_1, \rho_1), (\Theta_2, \rho_2)) = \min(val(\Theta_1, \rho_1), val(\Theta_2, \rho_2))$ , where we denote by  $valp((\Theta_1, \rho_1), (\Theta_2, \rho_2))$  the global magnitude of the peak pair (analogous for the other peak pair). Thus, at this step a decision is made concerning the final amplitude of opposite sides. The computed value is assigned to both arguments of the function. The minimum is to be taken as the length of both sides, therefore the equality of opposite edges of a rectangle is achieved.
4.  $val(\Theta_1, \rho_1) \geq 2 * average(\rho_3, \rho_4) * threshold_\rho$ , where  $val(\Theta_1, \rho_1)$  is the yielded magnitude of peak  $H_1$  ; the case repeats analogously for the other three peaks left. This means that a rectangle edge length is permitted to be greater than or equal to a fraction of the distance between the two parallel edges enclosed in the other peak pair. This constraint brings a considerable improvement over [10] who imposed a negligible difference between the length of opposite edges, that is, for an overloaded image, eligible rectangles can be refractory to the rules of selection.
5.  $||\Theta_1 - \Theta_3| - 90| < \varepsilon_\perp$  is the orthogonality test for adjacent rectangle edges.

### 3 Enhancements to Rectangle Extraction

One major benefit of the third dimension is the removal of almost all false positive and false negative cases in rectangle detection. These cases frequently appear in images where rectangles are adjacent and have similar orientations.

**False positive.** In [10] the authors identify cases where the detection is inaccurate. In our implementation, these situations are far less. An example of a wrong detection is the one depicted in Figure 3a, where the false positive rectangle is drawn in grey. Among the geometrical restrictions established in [10], one defines two parallel edges of a rectangle if each of the corresponding peaks quantizes the same number of pixels, that is, the edges have the same length. However, if the points distribution is neglected, a cell gathers points that can play the role of a rectangle edge as effect of accumulation only. In this manner, the accumulation matrix (Figure 3b) is identical to that of the wrongly detected rectangle, with respect to the number, values and position of the peaks. Our algorithm avoids the confusion because it intercepts a gap larger than the permitted one in the center of the grey rectangle edge and disregards it (see the small edges of the filled rectangle in Figure 3c). Another case of false positive detection occurs when a true rectangle "borrows" edge(s) from a nearby rectangle, thus becoming wider than in reality. The new algorithm avoids generating nonexistent edges also in this case, and the wider rectangle does not appear.



**Fig. 3.** a) Test image. b) Standard Windowed Hough Transform detected a false positive rectangle. c) The new method do not detect false peaks.

**False negative.** If the line support for a rectangle edge contains points from other objects or noise, the peak values of two parallel edges will no longer be the same and thus the rectangle is not detected. This is the case of a false negative. In our implementation, the decision about the congruence of parallel edges is made by truncating the longer segment to the borders of the shorter one and taking into account segment discontinuity. The truncation is obtained by considering the alignment of the corresponding line supports with respect to the locus of their centers. This is the way we can afford elasticity in choosing the value of  $D_{max}$ , otherwise  $D_{max}$  would have been the jurisdiction that limits the detection of rectangles to those inscribed in the circle of radius  $D_{max}$  only. Hence, the new approach is robust, rejecting extraneous features and environmental noise.

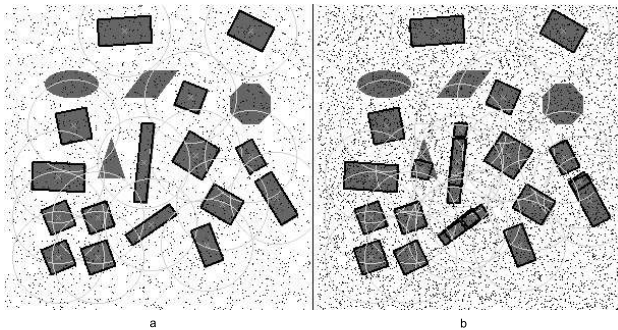
## 4 Experiments

This section aims to expose the algorithm to different kinds of inputs, that is, synthetical and real images. Our algorithm sets to work after a preprocessing step of contour extraction [4], so a good contour extraction is mandatory. In these experiments, we performed an exhaustive search of the rectangles, applying the algorithm to every pixel of the image. This can be optimized to aiming areas of interest only. For each rectangle from the image, the algorithm detects several candidates with similar centers, orientations and sizes. A clustering technique ensures the uniqueness of the detected rectangles.

The memory requirements due to the third dimension of the Hough matrix are of the order of few megabytes, negligible compared to usual memory sizes. For example, a window diameter of 100 pixels requires 2 MB memory.

### 4.1 Synthetic Images

We have employed synthetical images to test the correctness of theoretical aspects of the new method. Here, attention was focused on its behavior regarding noise resistance. To check the recognition quality in the presence of noise, we produced "salt and pepper" artifacts (Figure 4) over an edge-extracted image. At density  $\rho = 0.7$ , where  $\rho = 0.7$  means that the number of added points represent 70% of the number of points on the objects edges, the algorithm does not suffer from noise. Weak detections were recorded when the density limit was raised to  $\rho = 2.0$ . In Figure 4b, there are a few rectangles which are fragmented into smaller rectangular pieces. The phenomenon occurs with greater probability over rectangles with very short sides. This class of rectangles is very vulnerable to noise, as intense noise can generate false edges. Also, the noise resistance can be observed on natural images, in Figure 5 and Figure 6.

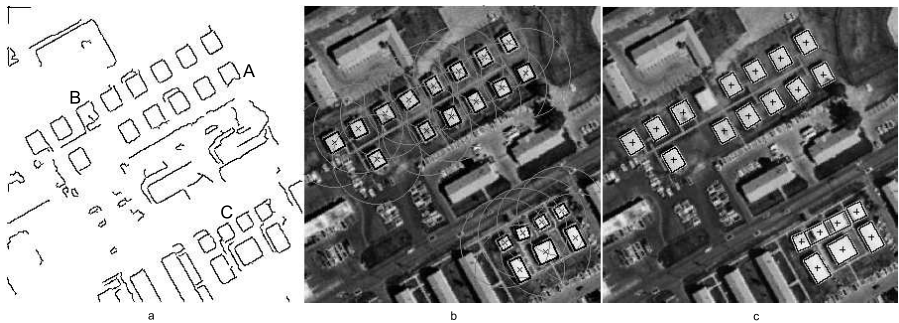


**Fig. 4.** a) Noised image with  $\rho = 0.7$ . b) Noised image with  $\rho = 2.0$ .

## 4.2 Real Images

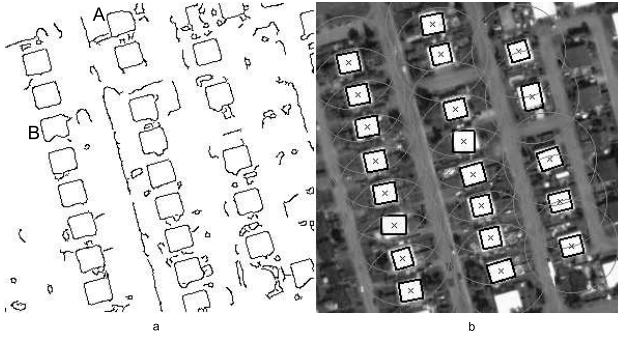
We have also tested the quality of recognition on various natural images. For the aerial image from [10], the detection was very good, recording no false positives nor false negatives (Figure 5). On an image with buildings of various dimensions, we have repeated the experiment with several values of  $D_{max}$  and obtained each time the same positive results. This proves the independence of the algorithm against  $D_{max}$ , in contrast with a well tuned value needed by [10]. The grey circles in the image represent the  $D_{max}$  borders of the search windows. The quality of detection is mainly dependent on thresholds assigned to the size of the gap permitted for a rectangle edge to have, and to the quantization step  $\Delta\rho$  chosen for  $\rho$ . The  $\theta$  quantization step was adapted to the variation of  $D_{max}$  as being the view angle of a pixel at a distance of  $D_{max}/2$  pixels:  $\Delta\theta = \text{round}(\arcsin \frac{2}{D_{max}})$ .

In Figure 5a, rectangles A and B were not omitted although one is occluded (rectangle A) and the other is considerably deformed (rectangle B). This is a consequence of the choice of  $\Delta\rho = 2$  and the value of  $gap = 2$ . In most of the cases,  $\Delta\rho$  is responsible for eliminating occlusions. An occluded edge can have pixels distributed on adjacent levels, interpreted as a big gap. A permissive choice of  $\Delta\rho$  is able to realign the segment, contracting this gap. The value of the  $gap$  parameter brings a significant contribution in avoiding the classification of rectangles that would elongate across those in the neighborhood (C in Figure 5c, [10]). The undetected rectangles are due to lack of information from the contour-based image. The algorithm was subjected to robustness tests on sets

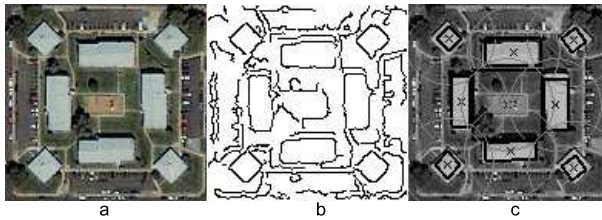


**Fig. 5.** a) Edge map. b) Rectangles detected by us. c) Rectangles detected in [10].

of satellite imagery. The rectangle features are identifiable, thus proving the feasibility of the extraction as in Figures 6 and 7. Although  $D_{max}$  exceeds the circumference of any construction, the recognition yielded good results. As seen in image 6, occlusions of rectangles A and B were vulnerable to parameters  $\Delta\rho = 3$  and  $gap = 3$ . Figure 7 proves that the algorithm behaves adequately regardless of rectangle size or orientation. It also passed the test of environmental noise.



**Fig. 6.** a) Edge map. b) Detected rectangles on satellite image.



**Fig. 7.** a) Original image. b) Edge map. c) Detected rectangles on satellite image.

## 5 Conclusions and Future Development

We address the problem of detecting rectangular structures from images through a process of contour-orientated segmentation. Our work aims to improve the usage of the Hough transform for this purpose. As Hough space loses important information regarding the position of points on lines, length of line segments and endpoints, we have added a third dimension to the original Hough accumulator, where we store the distribution of points for a current line. Thus, we can check every line segment for its consistency (lack of discontinuities) and for the symmetry property with respect to the center of its chord-support. The main advantage concerns the quality of detection: our technique ensures a significant reduction of false positive and false negative cases. Moreover, our method is no longer constrained by the choice of the search window diameter, therefore, it can detect rectangles with a wider range of dimensions. These improvements come at no extra computational cost, the time needed to compute point distribution is compensated by the significant decrease of the number of candidate lines for rectangle edges.

Future work will be centered on three aspects. We will try to improve the speed of the algorithm; we will attempt to automate edge detection by using heuristics, and to automatically adjust the maximum value of the parameter *gap* proportional with rectangle size; and we will extend the method to regular polygons.

## References

1. Dana H. Ballard. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122, 1981.
2. J. Banks, R. Rothnagel, B. Pailthorpe, and B. Hankamer. Automatic particle picking algorithms for high resolution single particle analysis. In *Australian Pattern Recognition Society Workshop on Digital Image Computing*, pages 127–132, 2005.
3. Nick Barnes, Gareth Loy, David Shaw, and Antonio Robles-Kelly. Regular polygon detection. In *10th IEEE International Conference on Computer Vision (ICCV)*, pages 778–785. IEEE Computer Society, 2005.
4. F. John Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.
5. Richard O. Duda and Peter E. Hart. Use of the Hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1):11–15, 1972.
6. Olivier Ecabert and Jean-Philippe Thiran. Adaptive Hough transform for the detection of natural shapes under weak affine transformations. *Pattern Recognition Letters*, 25(12):1411–1419, 2004.
7. Ping-Fu Fung, Wing-Sze Lee, and Irwin King. Randomized generalized Hough transform for 2-D grayscale object detection. In *International Conference on Pattern Recognition (ICPR)*, volume 3, pages 511–515, 1996.
8. C. Galambos, Josef Kittler, and Jiri Matas. Progressive probabilistic Hough transform for line detection. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1554–1560. IEEE Computer Society, 1999.
9. P.V.C. Hough. Method and means for recognising complex patterns. U.S. Patent No. 3069654, 1962.
10. Cláudio Rosito Jung and Rodrigo Schramm. Rectangle detection based on a windowed Hough transform. In *XVII Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI)*, pages 113–120. IEEE Computer Society, 2004.
11. Taejung Kim and Jan-Peter Muller. Development of a graph-based approach for building detection. *Image and Vision Computing*, 17(1):3–14, 1999.
12. Z.J. Liu, J. Wang, and W.P. Liu. Building extraction from high resolution imagery based on multi-scale object oriented classification and probabilistic Hough transform. In *Proceedings of the IGARSS 2005 Symposium*, 2005.
13. Sanjay Noronha and Ramakant Nevatia. Detection and modeling of buildings from multiple aerial images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(5):501–518, 2001.
14. Phil L. Palmer, Josef Kittler, and Maria Petrou. Using focus of attention with the Hough transform for accurate line parameter estimation. *Pattern Recognition*, 27(9):1127–1134, 1994.
15. Jiqiang Song, Min Cai, Michael R. Lyu, and Shijie Cai. A new approach for line recognition in large-size images using Hough transform. In *International Conference on Pattern Recognition*, vol. 3, pages 33–36. IEEE Computer Society, 2002.
16. Lei Xu, Erkki Oja, and Pekka Kultanen. A new curve detection method: randomized Hough transform (RHT). *Pattern Recognition Lett.*, 11(5):331–338, 1990.
17. Z. Yu and C. Bajaj. Detecting circular and rectangular particles based on geometric feature detection in electron micrographs. *J. Structural Biology*, 145:168–80, 2004.
18. Tao Zhao and Ramakant Nevatia. Car detection in low resolution aerial image. *8th Int'l. Conf. on Computer Vision*, pp. 710–717. IEEE Computer Society, 2001.
19. Yuanxin Zhu, Bridget Carragher, Fabrice Mouche, and Clinton S. Potter. Automatic particle detection through efficient Hough transforms. *IEEE Transactions on Medical Imaging*, 22(9):1053–1062, 2003.

# Fast Scalar and Vectorial Grayscale Based Invariant Features for 3D Cell Nuclei Localization and Classification

Janina Schulz<sup>1</sup>, Thorsten Schmidt<sup>1</sup>, Olaf Ronneberger<sup>1</sup>, Hans Burkhardt<sup>1</sup>,  
Taras Pasternak<sup>2</sup>, Alexander Dovzhenko<sup>2</sup>, and Klaus Palme<sup>2</sup>

<sup>1</sup> Albert-Ludwigs-Universität Freiburg, Institut für Informatik, Lehrstuhl für  
Mustererkennung und Bildverarbeitung

<sup>2</sup> Albert-Ludwigs-Universität Freiburg, Institut für Biologie II, Botanik  
jschulz@informatik.uni-freiburg.de

**Abstract.** Since biology and medicine apply increasingly fast volumetric imaging techniques and aim at extracting quantitative data from these images, the need for efficient image analysis techniques like detection and classification of 3D structures is obvious. A common approach is to extract local features, e.g. group integration has been used to gain invariance against rotation and translation. We extend these group integration features by including vectorial information and spherical harmonics descriptors. From our vectorial invariants we derive a very robust detector for spherical structures in low-quality images and show that it can be computed very fast. We apply these new invariants to 3D confocal laser-scanning microscope images of the Arabidopsis root tip and extract position and type of the cell nuclei. Then it is possible to build a biologically relevant, architectural model of the root tip.

## 1 Introduction

Groupwise Haar integration [1] of scalar 2D and 3D images has been successfully used to classify pollen grains [2] and to segment and classify cells in tissue samples [3]. These Haar integration features are solely based on scalar values like gray value and gradient magnitude, but ignore the direction of the gradient, which is an extremely robust feature, even under varying transformations and lighting conditions. This robustness is shown by e.g. [4], who use the direction of the gradient as their main features and gain impressive results on 2D images. We extend the groupwise Haar integration framework by including vectorial gradient information and by using spherical harmonics descriptors. We furthermore show how the generalized Hough transform for spheres can be considered as a special case of our vectorial Haar integration features.

The aim of this paper is to extract the location of the cell nuclei in the Arabidopsis root tip from 3D microscope images and furthermore decide if a cell nucleus is in interphase or in a phase of mitosis. Interphase and mitosis are phases of the cell cycle, in interphase the cell nucleus is in a non-dividing state. During



mitosis the actual division into two daughter cells takes place. We concentrate on interphase (comprising more than 95% of the cells) and metaphase, which is the most characteristic phase of the different phases of mitosis. Our final aim is to build a theoretical model of the root growth, therefore it is essential to gain information about the distribution of cell divisions inside the root tip.

We are not aware of groups using image analysis on 3D microscope images of the *Arabidopsis* root tip. Still, [5] identify and track cell nuclei in 2D images of *C. elegans* embryos. But in contrast to our 3D images, the cell nuclei in [5] are well separated and can be extracted using local signal maxima. [6] use simple features and a classification tree to classify tumor cells from normal cells in 2D images.

## 2 Description of the Data

Since we intend to gain as much information as possible about the location of the cell nuclei and their phase of mitosis in *Arabidopsis thaliana*, we stain the root tips with a fluorescent dye that binds to DNA (deoxyribonucleic acid), which is mainly located inside the cell nuclei. We use DAPI (4',6-diamidino-2-phenylindole), a common fluorescent staining. The roots are taken from plants at the age of three to five days, embedded in glycerol and captured as a 3D stack with a Zeiss LSM 510 META microscope with a water objective (C-Apochromat 63x/1.2 W corr) and an excitation wavelength of 364nm. The image quality depends on the age of the roots and the preparation steps (staining and washing), but the achieved image quality is reproducible. Fig. 1 shows an example slice of one of the 3D stacks used in the experiments. Most of the cell nuclei are cells in interphase and have a roughly spherical appearance with an unstained nucleolus inside each nucleus. In metaphase, the stained part of the cell usually has the shape of a flat disk. We use images with a voxelsize of  $0.6\mu\text{m}$  (for the detection of the cell nuclei in metaphase) and of  $0.25\mu\text{m}$  for all other computations.

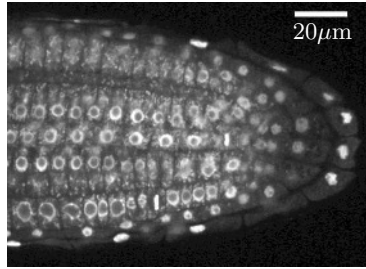
We first compute a series of invariant features for each voxel in the image (as presented in sec. 3), and then classify these features by use of a support-vector machine into the three classes *center of a cell nucleus in interphase*, *center of a cell nucleus in metaphase* and *no center of a cell*. Finally we visualize our results in a preliminary model of the *Arabidopsis* root tip.

## 3 Invariant Features by Groupwise Haar-Integration

Groupwise Haar integration [1] gains invariance of an image  $X$  under certain group operations by integration of a kernel function  $f$  over all these group operations:

$$I_f(X) = \int_G f(gX) dg \quad (1)$$

$G$  is the transformation group, under which the features  $I_f(X)$  should be invariant,  $g$  is one element of  $G$ . Function  $f$  computes a scalar value by a nonlinear,



**Fig. 1.** A typical slice of a 3D microscope image of the Arabidopsis root tip. Several characteristic aspects of the data are obvious: most cell nuclei are in interphase and are characterized by a roughly spherical contour but differing heavily in intensity. Each inner cell nucleus contains a nucleolus which forms a dark sphere, a double contour arises. In the middle and in the lower part of the figure, we see two cells in metaphase (vertical bars).

but otherwise arbitrary combination of all gray values in  $X$ . As the integral is independent of the particular position and orientation of image  $X$ , the integral is invariant under  $G$ . These features can be computed either for a whole image or a subimage  $X$  (*blockwise*) or *voxelwise* for each voxel  $\mathbf{x}_0 \in X$ . In this case, we shift the origin to voxel  $\mathbf{x}_0$  and integrate over all rotations. This results in features for each voxel such that in a later classification step, all voxels are classified separately. In the upcoming sections, the features are computed voxelwise unless otherwise indicated.

### 3.1 2-Point Grayscale Invariants

A very simple, but frequently ([3],[2]) used type of kernel function are functions  $f$  with

$$f(X) = f_1(X(\mathbf{0})) \cdot f_2(X(\mathbf{r})) \quad (2)$$

where  $f_1, f_2$  are arbitrary functions on the image  $X$ , and  $X(\mathbf{0}) = X(0,0,0)$ ,  $X(\mathbf{r}) = X(0,0,r)$ ,  $r \in \mathbb{R}$ . The characteristic criterion of  $f$  is that its evaluation depends only on two points in the image. [2] shows that a fast evaluation of  $I_f(X)$  is possible using fast convolution. As functions  $f_1$  and  $f_2$  we have chosen the identity, the square root, and the exponentiation to the powers of two and three, using both, image  $X$  and gradient magnitude image  $|\nabla X|$  as an input. Radius  $r$  has been in the range of  $1\mu\text{m}$  to  $5\mu\text{m}$ . As a preprocessing step, different gaussian filters with a standard deviation between 0.05 and 4 have been applied. This can be interpreted as using a smoothed 2-point kernel function that does not depend on two points but on two gaussian regions and leads to features robust against noise.

### 3.2 Voxelwise Vector Based Grayscale Invariants

Only scalar gray value information at different positions in the image has been used with the previously described invariants. But the grayscale invariant frame-

work can be theoretically extended to include directional information [7]. We decided to use the gradient as the most important directional information and we associate every point  $\mathbf{x}$  with its gray value gradient  $(\nabla X)(\mathbf{x})$ . The general formula of a kernel function that depends on the gradient image  $\nabla X$  and is invariant under the rotation  $R$  around point  $\mathbf{x}_0$  is

$$I_f(X, \mathbf{x}_0) = \int_G f(g_R g_{\mathbf{x}_0} \nabla X) dg_R \quad (3)$$

Here  $g_{\mathbf{x}_0}$  denotes the translation of point  $\mathbf{x}_0$  to the origin and  $g_R$  operations of the rotation group  $G$ . To guarantee that  $I_f(X, \mathbf{x}_0)$  is computationally affordable, we restrain  $f$  to only depend on few values. We choose the simplest type of kernel functions, a 1-point kernel, for  $f$ :

$$f(\nabla X) = f_1(\nabla X(\mathbf{r})) \text{ with } f_1(\mathbf{u}) = \frac{\mathbf{u}}{|\mathbf{u}|} \cdot \mathbf{w} \quad (4)$$

$\mathbf{w}$  denotes a fixed unit vector,  $\cdot$  is the scalar product. Function  $f_1$  computes the scalar product of its argument with a fixed given vector (both vectors being unit vectors). The invariant  $I_f(X, \mathbf{x}_0)$  becomes

$$I_f(X, \mathbf{r}, \mathbf{x}_0) = \int_G f(g_R g_{\mathbf{x}_0} \nabla X) dg_R \quad (5)$$

$$= \int_G f_1((g_R g_{\mathbf{x}_0} \nabla X)(\mathbf{r})) dg_R \quad (6)$$

$$= \int_G \frac{(g_R g_{\mathbf{x}_0} \nabla X)(\mathbf{r})}{|(g_R g_{\mathbf{x}_0} \nabla X)(\mathbf{r})|} \cdot \frac{\mathbf{r}}{|\mathbf{r}|} dg_R \quad (7)$$

We now consider the special case of Euclidean coordinates and thus integrate over all rotation matrices  $R$ . The inverse matrix  $R^{-1}$  undoes the rotation of the gradients under rotation of the image  $X$ . This is a major difference to the computation of grayscale invariants on images with only scalar values. Here  $O_3$  is the group of all rotation matrices.

$$I(X, \mathbf{r}, \mathbf{x}_0) = \int_{O_3} R^{-1} \frac{(\nabla X)(R \mathbf{r} - \mathbf{x}_0)}{|(\nabla X)(R \mathbf{r} - \mathbf{x}_0)|} \cdot \frac{\mathbf{r}}{|\mathbf{r}|} dR \quad (8)$$

This invariant is a strong measurement for how spherical given structures around point  $\mathbf{x}_0$  are as it accumulates gradients that show in radial direction towards  $\mathbf{x}_0$ . We use this as a basic detector for the roughly spherical cell nuclei in interphase. As only nonlinear kernel functions are able to distinguish between complex equivalence classes, we include another highly nonlinear component to our invariant, that improves results significantly. We choose a peak-like gaussian function  $G_\sigma$  as a nonlinear weight of the scalar product, applied before integration.

Our invariant only uses unit vectors and thus dismisses all information about how strong the gradients are. As a result, the feature is independent of the strength of the edges and of the gray value. This is mostly desired, as the contours differ markedly in strength. Thus we explicitly do not weight the summands

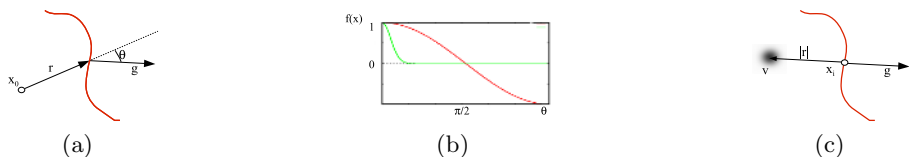
with the gradient magnitude, yet we decided to include the absolute gray value  $X(R\mathbf{r} - \mathbf{x}_0)$  as a factor. So, we avoid detecting dim and high-frequency noise and at the same time emphasize cell nuclei with a comparably bright, but very soft contour. Finally, we compute the invariant on both the original image and the inverted image to utilize both, the contour of the cell nucleus and the inner contour of the nucleolus.

The integration over all rotation matrices is impractical for large images, because it has to be evaluated for all points  $\mathbf{x}_0 \in X$  and all possible radii  $|\mathbf{r}|$ . [7] avoids this problem by computing the integral only for a small, mostly random subset of all points, but this is only reasonable if the data is already segmented. Thus we developed a very fast method to approximate eq. (8) combined with a gaussian peak and compute  $I(X, \mathbf{r}, \mathbf{x}_0)$  for all points  $\mathbf{x}_0 \in X$  and for a set of  $m$  different radii  $|\mathbf{r}_i|$  in  $O(|X| \cdot m)$ .

**Fast Computation of Vector Based Grayscale Invariants.** In eq. (8) combined with a gaussian peak only very few of the summands contribute significantly to the integral. We use this sparseness to reduce computation time tremendously by changing the evaluation order. We do not compute the integral over all rotation matrices for all  $(\mathbf{r}, \mathbf{x}_0)$  sequentially, but we consider for each point  $\mathbf{x}_i \in X$  all integrals  $I(X, \mathbf{r}, \mathbf{x}_0)$  to which  $\mathbf{x}_i$  contributes significantly. The respective contribution of each point  $\mathbf{x}_i$  can be accumulated easily for each integral  $I(X, \mathbf{r}, \mathbf{x}_0)$  at once by introducing a voting scheme based on an iteration over all gradients. Therefore, for every voxel  $\mathbf{x}_i$  with associated gradient  $(\nabla X)(\mathbf{x}_i)$  and for every possible radius  $r$ , we vote for the coordinates  $(x, y, z)$  of the point  $\mathbf{v}$  that lies in direction of the gradient at distance  $r$  from  $\mathbf{x}_i$ , as this point is the main contributor to the integral. This results in a four-dimensional parameter space  $V(x, y, z, r)$  that reflects how strong a perfect sphere with radius  $r$  is expressed around position  $\mathbf{v} = (x, y, z)$ . Afterwards  $V(x, y, z, r)$  is smoothed with a four-dimensional gaussian filter to become robust against disturbances of the spherical structure. That way we take the gaussian distribution applied to eq. (8) into account. It is not strictly equivalent but includes a smoothing in direction of the radius, which is not given in eq. (8) but desired. As a result, local maxima reflect centers of spheres. They are found by sequentially extracting global maxima and setting the neighborhood defined by radius  $r$  in  $V(x, y, z, r)$  to zero. Using a divide & conquer approach it is possible to extract  $k$  maxima in  $O(N + r_{\max} \cdot \log N/d)$  instead of the naive  $O(kN)$ , what makes the effort for extraction of maxima negligible compared to the invariant computation. During accumulation of the votes in  $V(x, y, z, r)$  it is possible to skip very low gradients and thus reduce computation time even more without worsening the results.

Regarding this computation method it becomes obvious that the 1-point vector based grayscale invariants form basically a generalized Hough transform (GHT) [8] for spheres. The generalized Hough transform usually considers the angle between the gradient at a point  $\mathbf{x}_i$  and the vector from  $\mathbf{x}_i$  to a point of reference (center) and maintains a memory-intensive lookup table. This is what eq. (8) implicitly does, but it is outperformed with respect to both, time and memory, by the use of the gaussian  $G_\sigma$  and the fast computation method.

Thus we have shown, that the generalized Hough transform for spheres can be considered as a special case of vectorial grayscale invariants, namely of those with the simple 1-point kernel of eq. (4). The ability to discriminate between complex equivalence classes increases with the complexity of the kernel function, especially 2- and 3-point kernel functions are more powerful than 1-point kernel functions. Thus the vectorial grayscale invariants form a very powerful framework embedding the robustness of the GHT.



**Fig. 2.** Fig. (a) visualizes eq. (8). Starting from a base point  $\mathbf{x}_0$  (i.e. a potential center of a cell nucleus) the scalar product between vector  $\mathbf{r}$  and gradient  $\mathbf{g}$  is computed. Fig. (b) shows how the scalar product would behave against the angle  $\theta$  between  $\mathbf{r}$  and  $\mathbf{g}$  (red), whereas the weighting with a gaussian function (green) assures that only small  $\theta$  near 0 contribute to the integral. In fig. (c) the fast computation method (sec. 3.2) is illustrated. At every point  $\mathbf{x}_i$  a smoothed vote is given for the point  $\mathbf{v}$  that lies in opposite direction of the gradient  $\mathbf{g}$  at distance  $|\mathbf{r}|$ . Comparing fig. (a) and (c) it becomes obvious how the fast computation method inverts the evaluation steps.

### 3.3 Spherical Harmonics Descriptors

An additional set of invariants is computed by using spherical harmonics descriptors [9]. We expand the gray values on spheres around certain points  $\mathbf{x}_0$  in spherical harmonics and determine the bandwise distribution of the signal energy. These spherical harmonics descriptors can easily be embedded in the Haar integration framework.

Every function  $f$  in spherical coordinates  $(\theta, \phi, \rho)$  that does not depend on  $\rho$  can be expanded in a series of spherical harmonics  $Y_l^m(\theta, \phi)$ :

$$f(\theta, \phi) = \sum_{l=0}^{\infty} \sum_{m=-l}^l C_l^m \cdot Y_l^m(\theta, \phi) \quad (9)$$

The coefficients  $C_l^m$  are computed by a projection of function  $f$  onto the basis:

$$C_l^m = \int f(s) Y_l^{m*}(s) ds \quad (10)$$

To gain rotation invariant features we expand at every point  $\mathbf{x}_0$  a sphere with radius  $r$  in spherical harmonics and analyse the bandwise fraction of the total

signal energy for each band  $l$ . The operation  $g_r$  denotes a scaling of the sphere to radius  $r$ :

$$I(l, r, \mathbf{x}_0) = \frac{\left( \sum_{m=-l}^l \int X(\mathbf{x}) \cdot (g_{\mathbf{x}_0} g_r Y_l^{m*})(\mathbf{x}) \, d\mathbf{x} \right)^2}{\int X^2(\mathbf{x}) \cdot (g_{\mathbf{x}_0} g_r Y_0^{0*})(\mathbf{x}) \, d\mathbf{x}} \tag{11}$$

This can be considerably simplified as the spherical harmonics form an orthogonal basis:

$$I(l, r, \mathbf{x}_0) = \frac{\sum_{m=-l}^l \left( \int X(\mathbf{x}) \cdot (g_{\mathbf{x}_0} g_r Y_l^{m*})(\mathbf{x}) \, d\mathbf{x} \right)^2}{D + \int X^2(\mathbf{x}) \cdot (g_{\mathbf{x}_0} g_r Y_0^{0*})(\mathbf{x}) \, d\mathbf{x}} \tag{12}$$

The denominator reflects the total signal energy of the sphere to which we add a denoising term  $D$  to become robust against small energy peaks (i.e. noise). The invariant of band  $l = 2$  is particularly well suited to characterize the flat disk shape of cells in metaphase. We evaluated only band 2 on spheres with radii up to  $3.5\mu\text{m}$  and a series of gaussian preprocessing filters with standard deviations of 0.03, 0.15 and 0.27. For  $D$  it has proven sensible to use a value almost in the region of the total energy.

**General Spherical Invariants.** Another set of invariants that helped to describe the cell nuclei are what we named general spherical invariants. They can also be expressed as a Haar integration kernel and they are a generalization of 2-point grayscale invariants, pseudo 3-point invariants, see [3], and spherical harmonics descriptors. They can be computed according to

$$I_{f_1, f_2, f_3}(X, \mathbf{x}_0, r, l) = f_1(X(\mathbf{x}_0)) \cdot f_2 \left( \sum_{m=-l}^l \left( \int f_3(X) \cdot (g_{\mathbf{x}_0} g_r Y_l^m)(\mathbf{x}) \, d\mathbf{x} \right)^2 \right)$$

Here,  $f_1, f_2, f_3$  are transformations on image  $X$ , e.g. pointwise exponentiation to different exponents. This possibility to include a variety of nonlinear transformations is one advantage of these invariants. We evaluated band 0, 1, 2, 3 and 4 with radii up to  $5.5\mu\text{m}$  and functions  $f_1(X) = X$  and  $f_{2,3}(X) = \sqrt{X}$  after applying a gaussian smoothing filter (with  $\sigma \in \{0.06, 0.18, 0.3\}$ ).

### 3.4 Radius and Gray Value Cooccurrence Matrices

Our classification results can be further improved by including an explicit measurement how gray values are distributed in a local region around point  $\mathbf{x}_0$ . Therefore we build a two-dimensional matrix for every voxel  $\mathbf{x}_0$  with entries of the absolute number of voxels with gray value  $g_i$  at distance  $r_i$  to point  $\mathbf{x}_0$ . For the gray values we use five bins and eight for the radii up to  $9\mu\text{m}$ . The radius and gray value cooccurrence matrix is computed with input images  $X$  and  $|\nabla X|$ . Furthermore we use the minimal, maximal and average distance of all bright points, i.e. points with at least 80% of the maximal gray value in a local region around point  $\mathbf{x}_0$  in the gaussian smoothed image  $G_\sigma(X)$ , and their

standard deviation as features. Another small subset of our features compute the square root of the sum of all points in distance  $r$  from point  $\mathbf{x}_0$  in images  $X$  and  $|\nabla X|$ . Radius  $r$  is chosen between  $0.6\mu\text{m}$  and  $12.0\mu\text{m}$ , and as a preprocessing step, gaussian filters of standard deviation  $\sigma = 0.03$  and  $0.18$  are used.

### 3.5 Evaluation of the Features

Our aim is to classify each voxel as being a central point of either a cell in interphase or a cell in metaphase or none of it. To reach this with a minimal effort of computing time we divide the process into two steps:

1. Detection of the cell nuclei in interphase
  - (a) Evaluation of the vector based grayscale invariants (sec. 3.2). They are a very good estimate for the position of cell nuclei in interphase as they detect spherical structures.
  - (b) To verify these hypotheses for cell nuclei in interphase, we compute further blockwise invariants in a local spherical subimage around the maxima detected in step 1 (according to sections 3.1, 3.4).
  - (c) The invariants are used as features by a support-vector machine (SVM) to classify the subimages into two classes *cell nucleus* and *not a cell nucleus*.
2. Detection of the cell nuclei in metaphase
  - (a) The invariants from sec. 3.1 and 3.3, the original image and gradient magnitude images smoothed with gaussian filters are used as voxelwise features. A support-vector machine classifies each voxel into one of the classes *centers of cells in a mitosis phase* and *other voxels*

We use a two-class support-vector machine with a gaussian kernel with parameters  $\gamma = 0.001$  and  $\text{cost} = 10$ . These parameters were selected by a grid search done on a large range of  $\gamma$  and the cost factor.

## 4 Experiments and Discussion

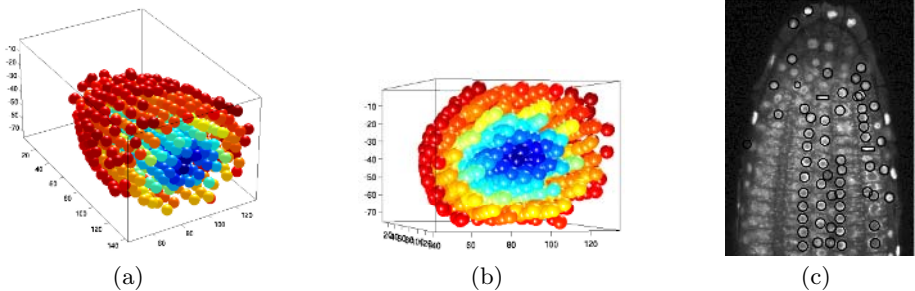
The invariants are selected and optimized for the localization and classification of cell nuclei in 3D confocal laser-scanned microscope images of the Arabidopsis root tip. For evaluation we chose five 3D image stacks from five different plants, computed the invariant features, trained the support-vector machine with two of the stacks and used the three remaining stacks as strictly separated test sets.

Our quantitative results in table 1 and fig. 3 show that a sensible model can be built with the information extracted by our invariants. It has been possible to classify over 80% of the cell nuclei correctly. It is not relevant if we miss some of the cell nuclei in interphase, but it is important to retrieve enough information about the location of these cell nuclei that it is possible to define the architecture of the Arabidopsis root tip, this means to identify the different cell files (fig. 3). We have easily reached this aim. For the biological research concerning root growth it is important that as few cell nuclei in metaphase as possible are missed.

This allows us to draw conclusions about the statistical distribution of dividing cells given a sufficient amount of data sets. The fact, that only one cell nucleus in metaphase has been missed in only one test set is a very strong result. It allows us to rely on the total recall of the cell nuclei in metaphase. If full precision is needed, a human interactor only has to double check for false positives in the very small-sized set of localized cell nuclei. These quantitative results can easily be visualized in a 3D model of the root tip (fig. 3 (a), (b)). Each sphere corresponds to one detected cell nucleus. The coloration of the spheres is according to their distance from the hull of the root tip. This allows to distinguish between the different cell files and to identify the cellular architecture of the root tip. Figure 3 (c) shows a slice of the original 3D microscope image. We have marked the cell nuclei detected by our invariants.

**Table 1.** Quantitative results. We show the confusion tables of the voxelwise classification of the voxels in three test sets of whole Arabidopsis root tips from different plants. Another two data sets from different plants have been used as training sets. The results show that it has been possible to extract most of the cells in interphase (I.). They are needed to gain information about the architecture of the Arabidopsis root tip. Furthermore it has been possible to detect all cell nuclei in metaphase (M.), except one in one test set. They are the crucial information for a biological analysis of the root growth. In the case of missed centers in interphase, we distinguish between cell centers missed by our vector invariants (sec. 3.2) and cells missclassified by the SVM, the sums in the confusion tables represent that. The class of voxels that are not the center of a cell in interphase or metaphase is abbreviated to *no c.* for *no center*.

N1	classified as			N2	classified as			N3	classified as		
	no c.	I.	M.		no c.	I.	M.		no c.	I.	M.
no c.	$8 \cdot 10^7$	37	1	no c.	$8 \cdot 10^7$	32	1	no c.	$8 \cdot 10^7$	34	4
I.	184+26	<b>934</b>	0	I.	232+39	<b>1009</b>	0	I.	255+37	<b>962</b>	0
M.	1	0	<b>12</b>	M.	0	0	<b>5</b>	M.	0	0	<b>10</b>



**Fig. 3.** Visualization of the results. Each sphere represents a cell nucleus we detected. The cellular architecture is clearly visible in fig. (a) and (b) as the cells form long strands towards the tip. In the example slice in fig. (c) the detected cells in interphase are marked with a circle, the cells in metaphase with a box.



## 5 Conclusion and Further Work

This paper introduces a composition of partly new invariant features that are based on grayscale invariants. The established scalar grayscale invariants have been significantly extended to include vectorial information. In particular we have shown how a robust detector for spherical structures can be derived from vectorial invariants and how it can be computed very fast.

We apply our set of invariants to laser-scanned 3D images of Arabidopsis root tips where the cell nuclei have been stained. We correctly classify about 80% of the cell nuclei in interphase and have succeeded in building an architectural model of the root tip. No tedious manual counting and/or segmentation of the cells in 3D stacks is required any more to analyze the cellular arrangement.

Furthermore we have very reliably localized the cells in metaphase (near 100% recall), which is crucial for further research in the field of Arabidopsis root growth. To measure growth at a cellular level, we need a strong, quantitative indicator, where cell division takes place.

An automated large-scale evaluation of 3D Arabidopsis microscope images based on the work done is planned for the near future. Further work will include microscope images of plants marked with green fluorescent proteins (GFP), these gene expressions are able to color exactly one or two of the cell files. This simplifies the classification of the cell nuclei into these cell files enormously and thus a more stable analysis of the file-based distribution of the cell nuclei is possible.

## References

1. Schulz-Mirbach, H.: Invariant features for gray scale images. DAGM-Symposium, Bielefeld, Germany (1995)
2. Ronneberger, O., et al.: General-purpose object recognition in 3d volume data sets using gray-scale invariants. ICPR, Quebec, Canada (2002)
3. Fehr, J., et al.: Self-learning segmentation and classification of cell-nuclei in 3d volumetric data using voxel-wise gray-scale invariants. DAGM-Symposium, Vienna, Austria (2005)
4. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *Int. J. of Computer Vision*, 60, 2 (2004)
5. Bao, Z., et al.: Automated cell lineage tracing in caenorhabditis elegans. *PNAS* (2006)
6. Wirjadi, O., et al.: Automated feature selection for the classification of meningioma cell nuclei. *Bildverarbeitung für die Medizin* (2006)
7. Reisert, M., et al.: General purpose invariant 3d features based on group integration using directional information and spherical harmonic expansion. ICPR, Hong Kong (2006)
8. Ballard, D.: Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2) (1981)
9. Kazhdan, M., et al.: Rotation invariant spherical harmonic representation of 3d shape descriptors. *Symp. on Geom. Process.* (2003)

# Integrating Recognition and Reconstruction for Cognitive Traffic Scene Analysis from a Moving Vehicle

Bastian Leibe<sup>1</sup>, Nico Cornelis<sup>2</sup>, Kurt Cornelis<sup>2</sup>, and Luc Van Gool<sup>1,2</sup>

<sup>1</sup> ETH Zurich, Switzerland

{leibe, vangool}@vision.ee.ethz.ch

<sup>2</sup> KU Leuven, Belgium

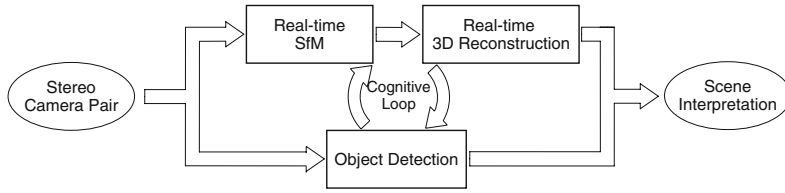
{firstname.lastname}@esat.kuleuven.be

**Abstract.** This paper presents a practical system for vision-based traffic scene analysis from a moving vehicle based on a cognitive feedback loop which integrates real-time geometry estimation with appearance-based object detection. We demonstrate how those two components can benefit from each other's continuous input and how the transferred knowledge can be used to improve scene analysis. Thus, scene interpretation is not left as a matter of logical reasoning, but is instead addressed by the repeated interaction and consistency checks between different levels and modes of visual processing. As our results show, the proposed tight integration significantly increases recognition performance, as well as overall system robustness. In addition, it enables the construction of novel capabilities such as the accurate 3D estimation of object locations and orientations and their temporal integration in a world coordinate frame. The system is evaluated on a challenging real-world car detection task in an urban scenario.

## 1 Introduction

Our target application is the analysis of traffic scenes, especially the detection of parked and moving cars in crowded urban areas. Such an analysis has straightforward applications in automatic driver assistance systems for identifying potentially dangerous traffic situations and as a basis for higher-level assistance functions. For example, the accurate localization of parked cars may be used to direct a focus of attention to image locations at which an inadvertent child might suddenly enter the street. As most of the child's body will be occluded by other vehicles, detection is particularly difficult in those situations, and contextual priming may buy precious reaction time.

However, detection from a moving vehicle is notoriously difficult because of the combined effects of egomotion, blur, unknown scene content, significant partial occlusion, and rapidly changing lighting conditions between shadowed and brightly lit areas. In addition, geometric scene context, which has been routinely used for surveillance and tracking applications from static cameras (e.g. [7,12]), is far harder to obtain in a moving vehicle, where continuous recalibration is needed due to the changing environment and vehicle pitch during acceleration and deceleration. While considerable progress has been made in relatively clean highway situations (e.g. [2,1]), the reliable detection of vehicles and pedestrians in crowded urban areas is still an important challenge [5].



**Fig. 1.** Overview of our system integrating recognition and geometry estimation

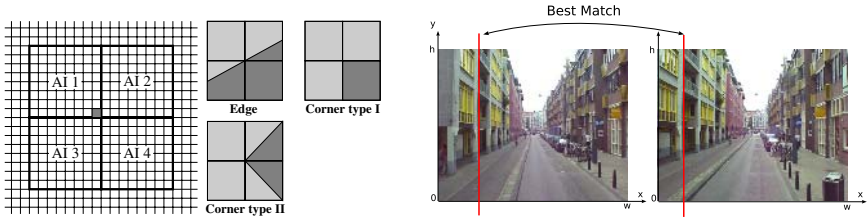
In this paper, we focus purely on vision as the most informative sensor. However, we integrate different cues and processing modalities: structure-from-motion (SfM), stereo reconstruction, and object detection. Our system is based on the idea of cognitive loops. While each of the component modules in isolation is limited, their interaction and exchange of information can compensate for the individual weaknesses and contribute to a reliable system response. Thus, the SfM and reconstruction modules collect knowledge about the scene geometry and the camera’s relative pose in it. However, relying on the assumption that a dominant part of the scene change is caused by egomotion, the estimation breaks down in crowded traffic situations. By detecting other moving objects and factoring out their influence on the scene change, the recognition module helps to obtain more reliable estimates. The recognition system, on the other hand, can profit immensely from knowledge about the scene geometry by applying ground plane constraints that the SfM and reconstruction modules can deliver.

The paper is structured as follows. The following section gives an overview of the proposed system. Sections 2 and 3 then describe the two main components and their interaction in detail. Section 4 finally presents experimental results.

**System Overview.** Figure 1 shows a visualization of our system setup. Our input data are two video streams recorded by a calibrated stereo rig mounted on top of the test vehicle, which are annotated with GPS/INS measurements. From this data, a Structure-from-Motion (SfM) algorithm first computes a camera pose for each image. Subsequently, these poses are used to generate a compact reconstruction of the surrounding road surface and facades using a fast dense-stereo algorithm [3]. Both of those stages are highly optimized and run at about 25 fps. In parallel, an object detection module is applied to both camera images in order to detect cars in the scene. The three modules are integrated in a tight cognitive loop. For each image, the object detection module receives scene geometry information, extracted from the previous frame, from the other two modules and feeds back information about detected objects to them, which is then used for processing the next frame. Thus, the modules exchange information that helps compensate for their individual failure modes and improves overall system performance. The next sections explain the different modules in detail.

## 2 Real-Time Geometry Estimation

In the first pathway, our system computes and permanently updates an estimate of the surrounding scene geometry. As space does not permit an in-depth discussion of well-known algorithms for Structure-from-Motion pipelines [6] and dense stereo [14], we



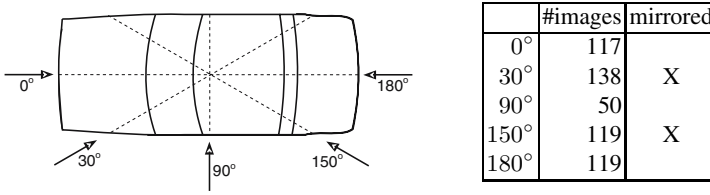
**Fig. 2.** (left) The fast feature measure used for SfM. (right) Rectified stereo pair.

limit the description to those changes that were made to allow for high-speed processing. Details of the described algorithms can be found in [3].

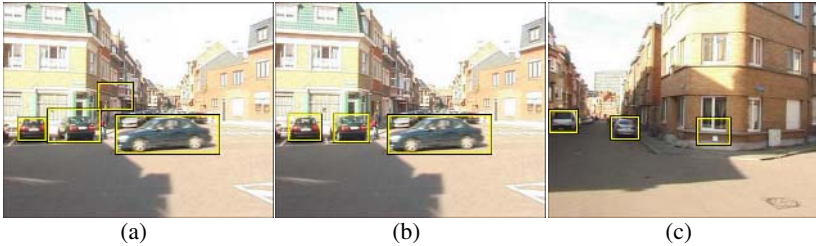
**Real-Time Structure-from-Motion Computation.** A real-time feature matcher extracts image feature points by finding local maxima of a very simple feature measure, based on the average intensity (AI) of four sub-regions (Fig. 2(left)):  $d = \text{abs}((\text{AI1} + \text{AI4}) - (\text{AI2} + \text{AI3}))$ . The extracted features are matched between consecutive images based on a fast sum of absolute intensity differences and then fed into a classic SfM pipeline, which reconstructs feature tracks and refines 3D point locations by performing triangulation. Sufficient baseline between images is guaranteed by only accepting a new image when the GPS or odometry signals sufficient movement. For efficiency reasons, only the green channel of one of the cameras is processed during SfM. A bundle adjustment routine is running in parallel with the main SfM algorithm to refine camera poses and 3D feature locations for previous frames and thus reduce drift. Additional GPS and odometry information can be used to guide feature matching during fast turns, to compensate for remaining drift, and to transfer the cameras into a global world coordinate system. The drift-compensated and globally aligned cameras are then rectified so that their up-vector is parallel to the world gravity vector. This ensures that 3D lines parallel to the gravity vector are displayed as vertical lines in each stereo pair.

**Real-Time 3D Reconstruction.** Next, a real-time geometry module reconstructs building facades using the (realistic) assumption that those can be modeled by ruled surfaces (i.e. surfaces made up of non-intersecting line segments) which are parallel to the gravity vector. For each rectified stereo pair, disparity values are computed for every vertical line using a single dynamic programming pass which is based on the ordering constraint and a robust line-based similarity measure (c.f. Fig.2(right)). Besides the tremendous gain in speed compared with algorithms which run dynamic programming on each horizontal scan line, the reconstruction becomes more accurate, as information over each vertical scan line can be integrated. The reconstructed volumes from all stereo pairs are then integrated over time into a topological map by a voting based carving algorithm. Finally, the road itself is reconstructed by fitting lines through the known contact points of the test vehicle's wheels with the road. This way of road reconstruction is not only faster than using dense stereo algorithms, but also more accurate since roads are often not textured enough for dense stereo.

**Derivation of Geometric Constraints.** For each image, the geometry module computes an estimate of the current ground plane by fitting a plane through the reconstructed road surface around the wheel contact points and extrapolating it along the current view-



**Fig. 3.** (left) Visualization of the viewpoints the single-view detectors were trained on. (right) Number of training images used for each view.



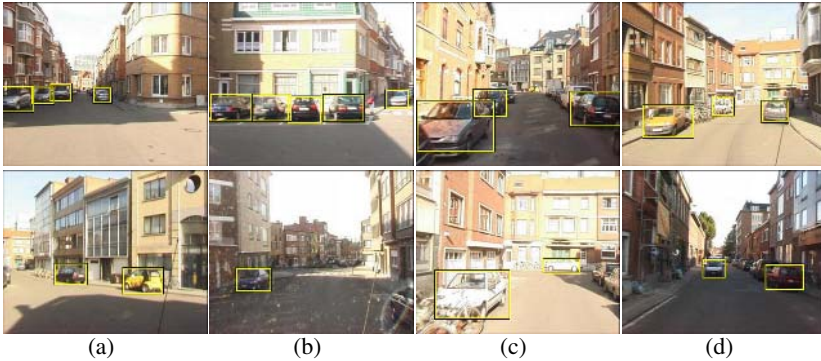
**Fig. 4.** Effect of scene geometry constraints: (a) object hypotheses before and (b) after ground plane constraints are enforced; (c) False positive that is filtered out by facade constraints

ing direction. By intersecting this plane with already reconstructed building facades, we can restrict the possible space in which objects may occur. This information is then passed to the recognition module to guide and improve object detection performance.

### 3 Object Detection

The recognition system is based on the ISM approach [8]. A bank of 5 single-view ISM detectors is run in parallel to capture different aspects of cars (see Fig. 3 for a visualization of their distribution over viewpoints). For efficiency reasons, we make use of symmetries and run mirrored versions of the same detectors for the other semi-profile views. All detectors share the same set of initial features: *Shape Context* descriptors [11], computed at *Harris-Laplace*, *Hessian-Laplace*, and *DoG* interest regions [11,10]. During training, extracted features are clustered into appearance codebooks, and each detector learns a dedicated spatial distribution for the codebook entries that occur in its target aspect. During recognition, features are again matched to the codebooks, and activated codebook entries cast probabilistic votes for possible object locations and scales according to their learned spatial distributions. The votes are collected in 3-dimensional Hough voting spaces, one for each detector, and maxima are found using MSME [8].

**Integration of Ground Surface Constraints.** Geometric scene constraints, such as the knowledge about the ground surface on which objects can move, can help detection in several important respects. First, they can restrict the search space for object hypotheses to a corridor in the  $(x, y, scale)$  volume, thus allowing significant speedups and filtering out false positives. Second, they make it possible to evaluate object hypotheses under a



**Fig. 5.** (top) Car detections on typical images from the city scenario. (bottom) Examples for the difficulties in this scenario: (a) motion blur, (b) lens flaring, (c) bright lighting (d) strong shadows.

size prior and “pull” them towards more likely locations. Last but not least, they allow to place object hypotheses at 3D locations, so that they can be corroborated by temporal integration. In the following, we use all three of those ideas to improve detection quality.

Given the camera calibration from SfM and a ground plane estimate from the 3D reconstruction module, we can estimate the 3D location for each object hypothesis by projecting a ray through the base point of its bounding box and intersecting it with the ground plane. If the ray passes above the horizon, we can trivially reject the hypothesis. In the other case, we can estimate its real-world size by projecting a second ray through the bounding box top point and intersecting it with a vertical plane through its 3D base. Using this information, we can formally express the likelihood for the real-world object  $H$  given image  $I$  by the following marginalization over the image-plane hypotheses  $h$ :

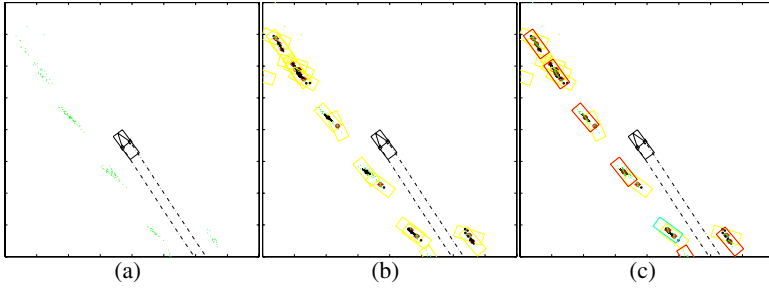
$$p(H|I) = \sum_h p(H|h, I)p(h|I) \sim \sum_h p(h|H)p(H)p(h|I) \quad (1)$$

where  $p(H)$  expresses a prior for object sizes and distances, and  $p(h|H)$  reflects the accuracy of our 3D estimation. In our case, we enforce a uniform distance prior up to a maximum depth of 70m and model the size prior by a Gaussian. The hypothesis scores are thus modulated by the degree to which they comply with scene geometry, before they are passed to the next stage (Fig. 4(a,b)).

**Multi-view Integration.** In order to fuse the single-view hypotheses into a consistent system response, we next apply the following multi-view integration stage. We first compute a top-down segmentation for each hypothesis  $h$  according to the method described in [8]. This yields two per-pixel probability maps  $p(\text{figure}|h)$  and  $p(\text{ground}|h)$  per hypothesis. With their help, we can express the hypothesis likelihood  $p(h|I)$  in terms of the pixels it occupies:

$$p(h|I) = \sum_{\mathbf{p} \in I} p(h|\mathbf{p}) = \sum_{\mathbf{p} \in \text{Seg}(h)} p(\mathbf{p} = \text{figure}|h)p(h). \quad (2)$$

where  $\text{Seg}(h)$  denotes the segmentation area of  $h$ , i.e. the pixels for which  $p(\mathbf{p} = \text{figure}|h) > p(\mathbf{p} = \text{ground}|h)$ . We then search for the optimal combination of



**Fig. 6.** Visualization of the temporal integration stage: (a) estimated 3D object locations (in green); (b) real-world object hypotheses obtained by mean-shift clustering (in yellow); (c) final hypotheses selected by the QBOP (in red)

hypotheses that best explains the image content under the constraint that each pixel can be assigned to at most one hypothesis. This is achieved by solving the following Quadratic Boolean Optimization Problem (QBOP):

$$\max_m m^T Q m = m^T \begin{bmatrix} q_{11} & \cdots & q_{1M} \\ \vdots & \ddots & \vdots \\ q_{M1} & \cdots & q_{MM} \end{bmatrix} m \quad (3)$$

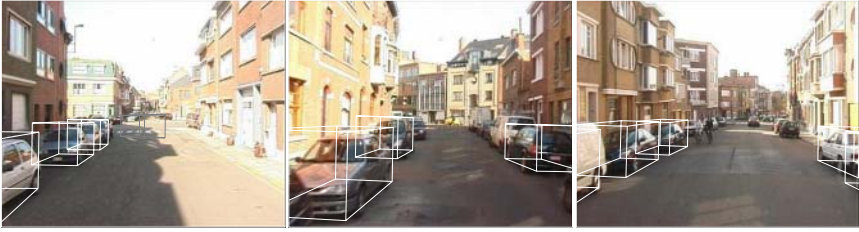
where  $m = (m_1, m_2, \dots, m_M)$  is a vector of indicator variables, such that  $m_i = 1$  if hypothesis  $h_i$  is accepted and 0 otherwise.  $Q$  is an interaction matrix whose diagonal elements  $q_{ii}$  express the merits of each individual hypothesis, while the off-diagonal elements  $q_{ij}$  express the cost of their overlap. In theory, we could directly use the hypothesis likelihood to define the merit. However, since we are dealing with incomplete information from sparsely sampled interest regions, we have to add a regularization term incorporating the number of pixels  $N$  in the figure-ground segmentation, as well as a normalization factor  $A_{\sigma,v}(h)$ , expressing the *expected area* of a hypothesis at its detected scale and aspect. The merit terms thus becomes

$$q_{ii} = -\kappa_1 + \frac{p(h_i|H_i)p(H_i)}{A_{\sigma,v}(h_i)} \left( (1-\kappa_2)N + \kappa_2 \sum_{\mathbf{p} \in \text{Seg}(h)} p(\mathbf{p} = \text{fig} \cdot |h_i) \right). \quad (4)$$

For the interaction terms, we measure the hypothesis overlap in the image and subtract the contribution of the overlapping area from the hypothesis  $h^* \in \{h_i, h_j\}$  that is farther away from the camera.

$$q_{ij} = -\frac{1}{2} \frac{p(h^*|H^*)p(H^*)}{A_{\sigma,v}(h^*)} \sum_{\mathbf{p} \in \text{Seg}(h_i) \cap \text{Seg}(h_j)} ((1-\kappa_2) + \kappa_2 p(\mathbf{p} = \text{fig} \cdot |h^*)) \quad (5)$$

This formulation allows to select the best global interpretation for each image from the output of the different single-view detectors. Since typically only a subset of hypotheses produces overlaps, it is generally sufficient to compute a fast greedy approximation to the optimal solution. Examples for the resulting detections are shown in Figure 5.



**Fig. 7.** Online 3D car location estimates (using only information from previous frames)

**Integration of Facade Constraints.** Using the information from 3D reconstruction, we add another step to check if hypothesized 3D object locations lie behind reconstructed facades (c.f. Fig. 4(c)). As this information will typically only be available after a certain time delay (i.e. when our system has collected sufficient information about the facade), this filter is applied as part of the following temporal integration stage.

**Temporal Integration.** The above stages are applied to both camera images simultaneously. The result is a set of 3D object hypotheses for each frame, registered in a world coordinate system. Each hypothesis comes with its 3D location, a 3D orientation vector inferred from the selected viewpoint, and an associated confidence score. Since each individual measurement may be subject to error, we improve the accuracy of the estimation process by integrating the detections over time.

Figure 6 shows a visualization of the integration procedure. We first cluster consistent hypotheses by starting a mean-shift search with adaptive covariance matrix from each new data point  $H$  and keeping all distinct convergence points  $\mathcal{H}$  (Fig. 6(b)). We then select the set of hypothesis clusters that best explains our observations by again solving a QBOP, only this time in the 3D world space:

$$\tilde{q}_{ii} = -\tilde{\kappa}_1 + \sum_{H \in \mathcal{H}_i} e^{-(t-t_i)/\tau} ((1 - \tilde{\kappa}_2) + \tilde{\kappa}_2 p(H|\mathcal{H}_i)p(H|I)). \quad (6)$$

$$\tilde{q}_{ij} = -\frac{1}{2} \sum_{H \in \mathcal{H}_i \cap \mathcal{H}_j} e^{-(t-t^*)/\tau} ((1 - \tilde{\kappa}_2) + \tilde{\kappa}_2 p(H|\mathcal{H}^*)p(H|I) + \tilde{\kappa}_3 O(\mathcal{H}_i, \mathcal{H}_j)) \quad (7)$$

where  $p(H|\mathcal{H}_i)$  is obtained by evaluating the location of  $H$  under the covariance of  $\mathcal{H}_i$ ;  $\mathcal{H}^*$  denotes the weaker of the two hypothesis clusters; and  $O(\mathcal{H}_i, \mathcal{H}_j)$  measures the overlap between their real-world bounding boxes, assuming average car dimensions. This last term is the main conceptual difference to the previous formulation in eqs. (4) and (5). It introduces a strong penalty term for hypothesis pairs that overlap physically. In order to compensate for false positives and moving objects, each measurement is additionally subjected to a small temporal decay with time constant  $\tau$ . The results of this procedure are displayed in Fig. 6(c).

**Estimating Car Orientations.** Finally, we refine our orientation estimates for the verified car hypotheses using the following two observations. First, the main estimation errors are made both along a car’s main axis and along our viewing direction. Since the latter moves when passing a parked car, the cluster’s main axis is slightly tilted towards our egomotion vector (c.f. Fig.6(a)). Second, the semi-profile detectors, despite being



trained only for  $30^\circ$  views, respond to a relatively large range of viewpoints. As a result, the orientation estimates from those detectors are usually tilted slightly away from our direction of movement. In practice, the two effects compensate for each other, so that a reasonably accurate estimate of a car's main axis can be obtained by averaging the two directions. Some typical examples of the resulting 3D estimates are shown in Fig. 7.

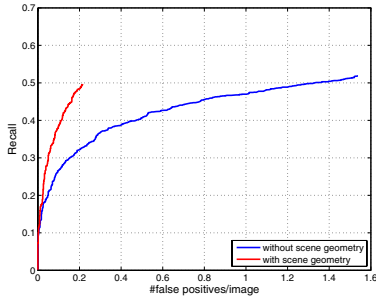
**Feedback into SfM and Reconstruction Modules.** The results of the previous stages have demonstrated that object detection can benefit considerably from knowledge about the scene geometry, delivered by the SfM and 3D reconstruction modules. However, those modules can also benefit from the results of object detection.

As discussed above, the SfM module relies on the assumption that a dominant part of the scene change is caused by egomotion. As a result, moving and/or shiny cars degrade the accuracy of the estimated camera positions. Although RANSAC outlier rejection [4] can to a certain degree compensate for this, there are many natural car motions that can be misinterpreted as static because of ambiguities in their image projection. E.g. following a car in the same lane at more or less the same speed on a straight stretch makes it clearly indistinguishable from a static object at infinity. Also, a car approaching on the other lane with a speed correlated to ours is indistinguishable from a static car parked somewhere in the middle of both lanes. Similarly, the fast 3D reconstruction relies on the assumption that the scene can be represented by ruled surfaces. Obviously, this is no longer the case when cars are parked in front of the facades. As a result, the cars introduce erroneous measurements into the dense stereo calculations which may influence the accuracy of the resulting scene geometry estimate (and thus of the ground plane estimate that will be provided to the detection module for the next frame).

The object detection module therefore completes the cognitive loop by feeding back information about its detections into the SfM and Reconstruction modules. By informing the SfM algorithm where cars can be expected, features will not be instantiated or tracked in those areas, thereby avoiding erroneous measurements which would result from tracking non-stationary points on moving and shiny cars. Similarly, object detection helps the reconstruction module by segmenting out all detected cars, so that the dense stereo reconstruction can focus on image areas that fulfill the ruled surface assumption. This continuous feedback of information is a crucial point for guaranteeing system reliability in complex real-world scenarios.

## 4 Experimental Results

In order to evaluate our method, we applied it to a test sequence, recorded by a camera vehicle over a distance of approximately 500m. The stereo input streams were captured at the relatively low resolution of  $380 \times 288$  pixels due to restrictions of the recording setup. Altogether, the data set comprises 1175 image pairs, which are processed at their original resolution by the SfM and reconstruction modules and bilinearly interpolated to twice that size for object detection (similar to [10]). The 5 single-view detectors were trained on images taken from the LabelMe database [13], for which viewpoint annotations and rough polygon outlines were already available (c.f. Fig.3). In all experiments, we set  $\kappa_2 = 0.95$ ,  $\tilde{\kappa}_2 = 0.5$ , and plot performance curves over the value of  $\kappa_1$ .



**Fig. 8.** (left) Comparison of the detection performance with and without scene geometry constraints. (right) 3D car location estimates using also information from future frames.

For a quantitative estimate of the performance improvement brought about by the inclusion of geometry constraints, we annotated the first 600 frames of the video sequence by marking all cars that were within a distance of 50m and visible by at least 40-50%. It is important to note that this includes many difficult cases with partial visibility, so it is unrealistic to expect perfect detection in every frame. We then evaluated the detection performance with and without ground plane constraints using the evaluation criterion from [9]. The results of this experiment are shown in Figure 8. As can be seen from the plot, detection reaches a level of about 50% recall in both cases. While the original recognition system yields 1.3 false positives per image at this level of recall, the inclusion of ground plane constraints significantly reduces the false positive rate to one every five images at 50% recall, or even one every ten images at 40% recall.

Counted over its full length, the sequence contains 77 (sufficiently visible) static and 4 moving cars, all but 6 of which are correctly detected in at least one frame. The online estimation of their 3D locations and orientation usually converges at a distance between 15 and 30m and leads to a correct estimate for 68 of the static cars; for 5 more, the obtained estimate would also have been correct, but does not reach a sufficiently high confidence level to be accepted. The estimates can further be improved by backpropagating also information from future frames (Fig. 8(right)). The SfM and reconstruction modules also profit from the feedback from object detection in terms of increased robustness. However, the exact benefit is hard to quantify, since no ground truth was available for the 3D measurements.

## 5 Discussion and Conclusion

In this paper, we have presented a system for cognitive traffic scene analysis that closely integrates structure-from-motion, 3D reconstruction, and object detection into a cognitive loop. At first view, it might seem unintuitive to incur the overhead of executing all three of those components in parallel, just to improve recognition performance. However, rather the opposite is the case: each individual task becomes considerably easier by its integration in the cognitive loop and the continuous feedback from the other

modules. As we have shown in this paper, the close interaction between the different modules increases both the recognition and 3D estimation performance, as well as the robustness of the entire system. In addition, our highly efficient implementation of the SfM and reconstruction modules allows them to run at video frame rate, so that their inclusion entails no additional delay. Although our current implementation of the object detector is not optimized for real-time processing yet, its individual stages are sufficiently simple, so that a time-efficient implementation is well possible.

In future work, we will aim to improve the representation of moving cars by adding a dedicated motion model. Secondly, we plan to extend recognition to other traffic participants, such as pedestrians and bicyclists, which was hitherto hindered by the poor resolution of our input video streams. Inferring a selective focus of attention from the detected car locations will help overcome this problem. Last but not least, we will optimize the implementation of our object detector for inclusion into a real-time application.

**Acknowledgments.** This work is funded, in part, by the EU project DIRAC (IST-2005-27787). We also wish to acknowledge the support of the K.U.Leuven Research Fund's GOA project MARVEL, Wim Moreau for the construction of the stereo rig, and TeleAtlas for providing additional video data to test on.

## References

1. L. Andreone, P.C. Antonello, M. Bertozzi, A. Broggi, A. Fascioli, and D. Ranzato. Vehicle detection and localization in infra-red images. In *Intel. Vehicles Symp.'02*, 2002.
2. M. Betke, E. Haritaoglu, and L.S. Davis. Real-time multiple vehicle tracking from a moving vehicle. *MVA*, 12(2):69–83, 2000.
3. N. Cornelis, K. Cornelis, and L. Van Gool. Fast compact city modeling for navigation pre-visualization. In *CVPR'06*, 2006.
4. M. Fischler and R. Bolles. Random sampling consensus: A paradigm for model fitting with application to image analysis and automated cartography. *Comm. ACM*, 24:381–395, 1981.
5. D. Gavrilu and V. Philomin. Real-time object detection for smart vehicles. In *ICCV'99*, pages 87–93, 1999.
6. R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2000.
7. D. Koller, K. Daniilidis, and H.-H. Nagel. Model-based object tracking in monocular image sequences of road traffic scenes. *IJCV*, 10(3):257–281, 1993.
8. B. Leibe and B. Schiele. Scale invariant object categorization using a scale-adaptive mean-shift search. In *DAGM'04*, Springer LNCS, Vol. 3175, pages 145–153, 2004.
9. B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. In *CVPR'05*, 2005.
10. D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2), 2004.
11. K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *PAMI*, 27(10), 2005.
12. A. Mittal and L.S. Davis. M2 tracker: A multi-view approach to segmenting and tracking people in a cluttered scene. *IJCV*, 51(3):183–203, 2003.
13. B.C. Russell, A. Torralba, K.P. Murphy, and W.T. Freeman. Labelme: a database and web-based tool for image annotation. Technical Report AIM-2005-025, MIT AI Lab, 2005.
14. D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47:7–42, 2002.

# Sparse Patch-Histograms for Object Classification in Cluttered Images

Thomas Deselaers, Andre Hegerath, Daniel Keysers, and Hermann Ney

Human Language Technology and Pattern Recognition Group  
RWTH Aachen University, Aachen, Germany  
deselaers@cs.rwth-aachen.de

**Abstract.** We present a novel model for object recognition and detection that follows the widely adopted assumption that objects in images can be represented as a set of loosely coupled parts. In contrast to former models, the presented method can cope with an arbitrary number of object parts. Here, the object parts are modelled by image patches that are extracted at each position and then efficiently stored in a histogram. In addition to the patch appearance, the positions of the extracted patches are considered and provide a significant increase in the recognition performance. Additionally, a new and efficient histogram comparison method taking into account inter-bin similarities is proposed. The presented method is evaluated for the task of radiograph recognition where it achieves the best result published so far. Furthermore it yields very competitive results for the commonly used Caltech object detection tasks.

## 1 Introduction

In the last years, part-based models in general, and patch-based models in particular, have gained an enormous amount of interest in the computer vision community [1, 2, 3]. These approaches offer some immediate advantages such as robustness against occlusion and translation invariance because the parts can be modeled more or less independently and thus an object that is partly occluded can be classified correctly as long as the visible parts can be recognized.

Nearly all approaches presented extract features only from a subset of positions in the images: most approaches use interest point detectors [1, 2, 3], random points [4], or points from a regular grid [5]. Obviously, by choosing a subset of feature extraction points, image information is lost which may result in decreased recognition performance. This may be passable in the case of general object recognition and detection, but can be unsuitable in the case of medical image analysis where no details may be missed. In contrast to all these approaches, the method presented here can efficiently deal with arbitrarily many features and thus we choose to extract several features at each position of the image. Only recently, some approaches that extract local features from all positions in the image were proposed [6, 7].

Similar to other approaches [5, 8], the presented approach uses patches, i.e. subimages, extracted from the images. Feature vectors representing the patches are derived from a PCA dimensionality reduction. These feature vectors are then stored in a special histogram structure that allows us to store high-dimensional feature vectors, which are then classified using various classification methods.

Another type of information that is often discarded when part-based models are applied is the spatial relationship between the parts. Many approaches completely discard these data [1, 5], and other approaches that explicitly model spatial relationships [8] have to be greatly simplified in order to become computationally feasible [2]. In the model presented here, the positions of the patches can be integrated directly without significant increase in computation time or storage requirements.

Furthermore, many approaches require time-consuming preprocessing steps such as vector quantization, to create a code-book of possible object parts [5, 8, 9]. Our approach skips this step and instead uses a generalized form of a code-book that is identical for all kinds of data. That is, the code-book is not learned from training data but is fixed before we know what data we will deal with. Obviously, this code-book needs a large amount of possible ‘code-words’ but due to an efficient representation this becomes computationally feasible.

The remainder of this paper is structured as follows: In the next section, we introduce the feature extraction technique and the sparse histogram representation of the images. In Section 3 we shortly introduce the three classification methods that are used to recognize the images represented by the sparse histograms. Section 4 describes the databases used to evaluate the methods and Section 5 presents and compares the experimental results with the best results published so far. Finally, the paper is shortly summarized and concluded in Section 6.

## 2 Sparse Histograms of Image Patches

Histograms are a well-known method to represent the distribution of data and are applied in the field of computer vision in various ways. One problem with histograms is that they become difficult to handle if the dimensionality of the input data is large, because the number of bins in a histogram grows exponentially with the number of dimensions of the data. For example, given 8 dimensional input data and only 4 subdivisions per dimension results in  $4^8 = 65,536$  bins.

To overcome this problem, we propose to use a sparse representation of the histograms, i.e. we store only those bins whose content is not empty. Sparse histograms have been used for other applications before [10]. This representation allows us to create histograms for data of arbitrary dimensionality. The only practical limitation to the size of the histogram is that for very large sizes, most of the bins that actually contain an element will contain only one single element, and this makes the comparison of histograms very unreliable.

## 2.1 Features

It has been shown that patches extracted from the images are a suitable means of representing local structures in images [5, 8, 9]. Thus, we choose to extract patches of different sizes at every position in each image. More precisely, we extract square patches with the edge lengths 7, 11, 21, and 31 pixels, which are then scaled to a common size of 15 pixels to be able to process them jointly later. These multiple patch sizes allow to account for objects of various sizes and lead to a certain invariance with respect to scale changes. A very similar approach to account for different scales was used in [11].

All patches are extracted from all training images and then a PCA transformation is jointly estimated. Using this PCA transformation all patches are reduced in dimensionality.

## 2.2 Creation of Histograms

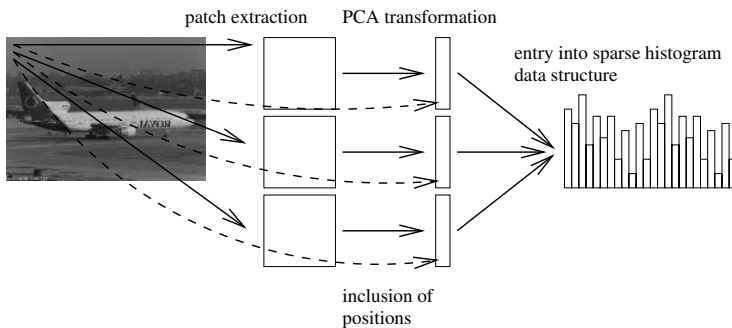
The distribution of the feature vectors described in the previous section is then approximated using a histogram. To reduce the necessary storage, the histograms are created without explicitly storing any feature vector. Thus, the creation of the histograms is a three step procedure: in the first step, the PCA transformation is determined as described above. In the second step, the mean and the variance of the transformed patches are calculated to determine a reasonable grid for the histograms. In the last step, the histograms themselves are created. For each of these steps, all training images are considered.

1. In the first step, all possible patches in various sizes from all training images are extracted and their mean and the covariance matrix are estimated to determine the PCA transformation matrix.
2. Given this PCA transformation matrix and the means, the mean  $\mu_d$  and the variance  $\sigma_d^2$  for each component  $d$  of the transformed vectors is calculated to determine the bin boundaries for the histograms. The bins for component  $d$  are uniformly distributed between  $\mu_d - \alpha\sigma_d$  and  $\mu_d + \alpha\sigma_d$ .
3. Then, we consider all dimensionality reduced patches from the training images and create one histogram per training image. This step is depicted in Figure 1. The processing is from left to right: first the patches are extracted, then PCA transformed, then the position of the patch is concatenated to the PCA transformed feature vector, and finally the vectors are inserted into the sparse histogram data structure.

As mentioned above, the patches are not explicitly stored in any of these steps as this would lead to immense memory requirements.

Informal experiments have shown that 6 to 8 dimensions for the PCA reduced vectors lead to the best results, and that  $\alpha = 1.5$  is a good value to determine bin boundaries. Values exceeding the given range are clipped.

*Spatial Information.* One serious issue with many part-based models is the incorporation of spatial information. To incorporate spatial information in our



**Fig. 1.** Creation of sparse histograms. Solid arrows denote appearance information of the patches, dotted arrows denote spatial information of the patches.

approach, we simply concatenate the extraction position to the PCA reduced feature vectors and thus simply add two further components to the histograms. These additional components can easily be handled by the histograms. As the range of values for each component is calculated individually and independently of the other components, no special processing of these additional components is required. One issue with the inclusion of the absolute patch extraction positions is that translation invariance, normally one of the major advantages of part-based models, is partly lost. Still, currently it is unclear how to incorporate relative position information into the model presented here. It will be shown later that for the tasks considered here, either the translation invariance is not required, or translations are sufficiently represented in the training data.

### 3 Classification of Sparse Patch Histograms

Given the sparse histograms that represent the images, any classifier that is able to handle the sparse representation can be used. We have tested three different classifiers: the nearest neighbor classifier in which we use two different distance functions, a classifier based on log-linear models trained using the maximum entropy criterion, and support vector machines.

#### 3.1 Nearest Neighbor Classification

Nearest neighbor classification is often used as a baseline for classification. Immediate advantages are that no expensive training process is necessary, implementation can be done easily, and different distance functions can be used to compare the data used. In accordance with [12] we use Jeffrey Divergence to compare histograms. To classify the histogram  $h$  representing the image  $X$  the following decision rule  $r(x)$  is used:

$$h \mapsto r(h) = \arg \min_k \left\{ \min_{n=1 \dots N_k} d(h, h_n) \right\}, \quad (1)$$

where  $h_n$  is the histogram representing the  $n$ th training image from class  $k$ . The Jeffrey Divergence  $d(h, h')$  between two histograms  $h$  and  $h'$  is defined as

$$d(h, h') = \sum_{c=1}^C h_c \log \frac{2h_c}{h_c + h'_c} + h'_c \log \frac{2h'_c}{h'_c + h_c}. \quad (2)$$

Here,  $h_c$  and  $h'_c$  are the  $c$ th bins of the histograms  $h$  and  $h'$ , respectively.

One problem with the Jeffrey Divergence is that similarities between neighboring bins are completely neglected. Other distance measures that take into account inter-bin-similarities, for example the earth mover's distance [12], are too computationally expensive to be used for histograms with several thousand bins. We propose to use a much simpler way of taking into account neighboring bins that is inspired by an image matching algorithm [13]. This method is called *Histogram Distortion Model* (HDM) and it can be implemented for any bin-by-bin histogram comparison measure straightforwardly, as long as neighborhoods are defined for the underlying histograms. Given a bin at position  $c = (c_1, \dots, c_D)$ , we use the bin from position  $\gamma$  out of the neighborhood  $U(c)$  of  $c$  that minimizes the resulting distance. Here, we use it as an extension to the Jeffrey Divergence, i.e., we replace the distance function  $d(h, h')$  by  $d_{\text{HDM}}(h, h')$  with

$$d_{\text{HDM}}(h, h') = \sum_{c=1}^C \min_{\gamma \in U(c)} h_c \log \frac{2h_c}{h_c + h'_\gamma} + h'_\gamma \log \frac{2h'_\gamma}{h'_\gamma + h_c}. \quad (3)$$

A related but computationally more expensive way to account for neighboring bins in the comparison of histograms would be to smooth the histograms. Here, the smoothing would lead to non-sparse histograms and thus it would lead to greatly increased computational requirements.

### 3.2 Maximum Entropy Classification

Maximum entropy classification and log-linear models are a well-known way to model probability distributions in natural language processing and in image recognition [14].

The maximum entropy approach directly optimizes the class posterior probability  $p(k|X)$ . Thus, it is a discriminatively trained model. Here, we want to model the posterior probability  $p(k|h)$  where  $h$  is the sparse histogram representing image  $X$ . Thus, the model for  $p(k|h)$  is

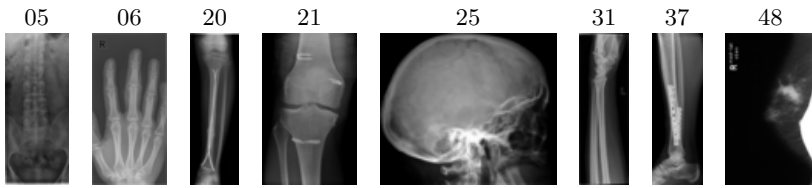
$$p(k|h) = \frac{1}{Z(h)} \exp \left( \alpha_k + \sum_{c=1}^C \lambda_{kc} h_c \right) \quad (4)$$

where  $h_c$  is the  $c$ th bin of the histogram  $h$  and  $Z(h)$  a normalization factor.

Efficient algorithms to determine the parameters  $\{\alpha_k, \lambda_{kc}\}$  exist. We use a modified version of generalized iterative scaling [15] to decrease the necessary computational effort. For classification, Bayes' decision rule is used:

$$h \mapsto r(h) = \arg \max_k \{p(k|h)\}. \quad (5)$$





**Fig. 2.** Example images of the IRMA 10000 database together with their class

### 3.3 Support Vector Machines

Support-vector-machines (SVM) are often used as a classification method that provides reasonable performance across various tasks. In the experiments we tried linear, polynomial and radial basis function kernels and optimized all parameters in cross-validation experiments on the training data.

## 4 Databases and Experimental Results

This section briefly presents the two databases used to evaluate our method, the IRMA 10000 database of medical radiographs and three of the Caltech object databases.

### 4.1 IRMA 10000

The IRMA 10000 database<sup>1</sup> was used in the automatic annotation task of the 2005 ImageCLEF evaluation [17]. It consists of 10,000 fully classified radiographs taken randomly from medical routine at a large hospital. The images are split into 9,000 training and 1,000 test images and are subdivided into 57 classes. Example images for some of the classes are given in Figure 2. In the ImageCLEF 2005 automatic annotation task a total of 40 runs were submitted by 12 groups. In Table 1 we give the best results from the evaluation and compare our results to these. To keep the computing requirements low, we scaled all images such that the longest edge was 128 pixels while preserving the aspect ratio.

### 4.2 Caltech Databases

To compare the performance of our method to object recognition algorithms from other groups, we use some of the Caltech databases that were introduced by Fergus et al. [8]. The task is to determine whether an object is present in an image or not. For this purpose, several sets of images containing certain objects (airplanes, faces, and motorbikes) and a set of background images not containing any of these objects<sup>2</sup> are given. The images are of various sizes and

<sup>1</sup> <http://irma-project.org>

<sup>2</sup> <http://www.robots.ox.ac.uk/~vgg/data>



**Fig. 3.** Example images from the Caltech data sets airplanes, faces, and motorbikes, and a background image

for the experiments they were converted to gray images. The airplanes and the motorbikes task consist of 800 training and 800 test images each, the faces task consists of 436 training and 434 test images. For each of these tasks, half of the images contain the object of interest and the other half does not. An example image of each set is shown in Figure 3. Many different groups have published results for these data. In Table 2 we summarize the best results we are aware of for each of the tasks to compare our results to. Here, we scaled the images to a common height of 128 pixels to keep the computing requirements low and to avoid the known issue that it is possible to classify some of the images just by image size [5].

## 5 Experimental Results

In this section, we present the results we obtained using sparse histograms of image patches for the IRMA and the Caltech tasks.

Table 1 gives an overview of the best results obtained for the IRMA tasks from the ImageCLEF 2005 evaluation [17] along with the results we obtained using sparse patch histograms with and without position information. For all experiments, the patches were reduced to 6 components using PCA. For the experiments with position, two components representing position were concatenated to the data vector thus resulting in 8 dimensional data. For all experiments, each component was subdivided into four steps, thus resulting in 4,096 and 65,536 bin-histograms for the experiments without and with spatial information respectively. These parameters were determined in informal cross-validation experiments to perform best on the average: For dimensionality reduction we measured the performance for dimensionalities between 4 and 10. Furthermore, we tried 2 to 6 subdivisions per component.

The results we obtained for this task are better than all results that are published for these data so far. With and without positions, the error rate is greatly improved using the histogram distortion model in comparison to using only the Jeffrey Divergence. This shows that the histogram distortion model is, at least partly, able to compensate for the sparseness of the histograms. As mentioned above, an alternative to the histogram distortion model would be to smooth the histograms, but informal experiments have shown that, apart from the problems of storage, the improvement is lower than using the deformation

**Table 1.** Results for the IRMA data. The comparison results are taken from the ImageCLEF 2005 automatic annotation task [17].

method	rankgroup	error rate [%]
image distortion model	1RWTH Aachen	12.6
image distortion model & texture feature	2IRMA Group	13.3
patch-based object classifier (maximum entropy)	3RWTH Aachen	13.9
patch-based object classifier (boosting)	4Uni Liège	14.1
image distortion model & texture feature	5IRMA Group	14.6
patch-based object classifier (decision trees)	6Uni Liège	14.7
GNU image finding tool	7Uni Geneva	20.6
32×32 images, Euclidean distance, nearest neighbor	--	36.8
sparse histograms (w/o position)	this work	
+ nearest neighbor		13.0
+ histogram distortion model, nearest neighbor		12.5
+ maximum entropy classification		11.6
+ support vector machine		11.3
sparse histograms (w/ position)	this work	
+ nearest neighbor		10.1
+ histogram distortion model, nearest neighbor		9.8
+ maximum entropy classification		<b>9.3</b>
+ support vector machine		10.0

model. The result obtained using maximum entropy training is again clearly improved for the case without position information. For the case with position information, the maximum entropy training cannot improve on the results.

In Table 2, results for the experiments on the three Caltech tasks are given. The first part of the table gives the best results we know for each of these tasks, the second part gives the results we obtained. We highlighted the best results in total and the best results we obtained with our method. Here again, using the histogram distortion model usually gave an improvement over the normal Jeffrey Divergence, and a further improvement can be achieved using the discriminatively trained log-linear model. Although the model we present is clearly much simpler than the models presented in [1, 2, 4, 8, 11], we achieve very competitive error rates. Using SVMs, the results are in the same area as those using the maximum entropy training. For both maximum entropy and SVM classifiers the results are better than those obtained using the nearest neighbor classification rule. This clearly shows that discriminative modeling can improve the results.

## 6 Conclusion

In this work we presented a part-based approach to object recognition that was evaluated on a database of medical radiographs and on three object recognition tasks. An advantage of this novel approach over other approaches is that it does not require large parts of the data to be disregarded, but instead almost

**Table 2.** Results for the Caltech data

method		error rate		
		airp.	faces	motb.
constellation model	[8]	9.8	3.6	7.5
improved constellation model	[2]	6.3	9.7	2.7
PCA SIFT features	[18]	2.1	0.3	5.0
patch-histograms, discriminative training	[11]	1.4	3.7	<b>1.1</b>
boosting weak hypotheses	[1]	2.5	<b>0.0</b>	5.7
texture features	[19]	<b>0.8</b>	1.6	7.4
sparse histograms (w/o position)				
+ nearest neighbor		4.9	12.7	6.1
+ histogram distortion model, nearest neighbor		4.8	13.6	7.0
+ maximum entropy classification		3.5	7.8	4.8
+ support vector machines		2.4	4.1	2.3
sparse histograms (w/ position)				
+ nearest neighbor		9.1	6.5	6.8
+ histogram distortion model, nearest neighbor		6.5	7.6	6.9
+ maximum entropy classification		1.9	<b>3.9</b>	1.8
+ support vector machines		<b>0.8</b>	4.4	<b>1.3</b>

arbitrary numbers of image patches can be handled by using a sparse histogram representation. Possible problems resulting from data sparseness are effectively counteracted by using a histogram distortion model which also improves the recognition results. Furthermore, the approach does not require an expensive training process, as the code-book is determined independently from the training data. The results obtained are the best published results for the task of radiograph recognition and are very competitive for the Caltech object recognition tasks. It was also shown that spatial information can easily be incorporated into the approach and that this information, although to the cost of losing translation invariance, can improve the results notably for the restricted domain task of radiograph recognition and in most cases for the Caltech tasks.

In the future we plan to extend the presented model to incorporate relative patch positions.

## References

1. Opelt, A., Pinz, A., Fussenegger, M., Auer, P.: Generic object recognition with boosting. **28**(3) (2006) 416-431
2. Fergus, R., Perona, P., Zisserman, A.: A sparse object category model for efficient learning and exhaustive recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 05). (2005) 380-389
3. Ulusoy, I., Bishop, C.M.: Generative versus discriminative methods for object recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 05). (2005) 258-265
4. Marée, R., Geurts, P., Piater, J., Wehenkel, L.: Random subwindows for robust image classification. In: IEEE Conference on Computer Vision and Pattern Recognition. (2005) 34-40

5. Deselaers, T., Keysers, D., Ney, H.: Discriminative training for object recognition using image patches. In: IEEE Conference on Computer Vision and Pattern Recognition. Volume 2., San Diego, CA (2005) 157-162
6. Bosch, A., Zissermann, A., Muñoz, X.: Scene classification via plsa. In: ECCV 2006. Volume 3954 of LNCS., Graz, Austria (2006) 517-530
7. Shotton, J., Winn, J., Rother, C., Criminisi, A.: TextonBoost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In: ECCV 2006. Volume 3951 of LNCS., Graz, Austria (2006) 1-15
8. Fergus, R., Perona, P., Zissermann, A.: Object class recognition by unsupervised scale-invariant learning. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 03), Blacksburg, VG (2003) 264-271
9. Leibe, B., Schiele, B.: Scale invariant object categorization using a scale-adaptive mean-shift search. In: DAGM. Number 3175 in LNCS (2004) 145-153
10. Linde, O., Lindberg, T.: Object recognition using composed repetitive field histograms of higher dimensionality. In: International Conference on Pattern Recognition, Cambridge, UK (2004)
11. Deselaers, T., Keysers, D., Ney, H.: Improving a discriminative approach to object recognition using image patches. In: DAGM 2005, Pattern Recognition, 26th DAGM Symposium. Number 3663 in Lecture Notes in Computer Science, Vienna, Austria (2005) 326-333
12. Puzicha, J., Rubner, Y., Tomasi, C., Buhmann, J.: Empirical evaluation of dissimilarity measures for color and texture. In: International Conference on Computer Vision. Volume 2., Corfu, Greece (1999) 1165-1173
13. Keysers, D., Gollan, C., Ney, H.: Local context in non-linear deformation models for handwritten character recognition. In: International Conference on Pattern Recognition. Volume 4., Cambridge, UK (2004) 511-514
14. Jeon, J., Manmatha, R.: Using maximum entropy for automatic image annotation. In: Proceedings of the 3rd International Conference on Image and Video Retrieval. (2004) 24-32
15. Darroch, J.N., Ratcliff, D.: Generalized iterative scaling for log-linear models. *Annals of Mathematical Statistics* 43(5) (1972) 1470-1480
16. Witten, I.H., Frank, E.: *Data Mining: Practical Machine Learning Tools and Techniques*. 2nd edn. Morgan Kaufmann, San Francisco (2005)
17. Clough, P., Mueller, H., Deselaers, T., Grubinger, M., Lehmann, T., Jensen, J., Hersh, W.: The clef 2005 cross-language image retrieval track. In: Working Notes of the CLEF Workshop, Vienna, Austria (2005)
18. Zhang, W., Yu, B., Zelinsky, G.J., Samarasinghe, D.: Object class recognition using multiple layer boosting with heterogeneous features. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 05). (2005) 323-330
19. Deselaers, T., Keysers, D., Ney, H.: Classification error rate for quantitative evaluation of content-based image retrieval systems. In: International Conference on Pattern Recognition 2004. Volume 2., Cambridge, UK (2004) 505-508

# An Object-Oriented Approach Using a Top-Down and Bottom-Up Process for Manipulative Action Recognition

Zhe Li, Jannik Fritsch\*, Sven Wachsmuth, and Gerhard Sagerer

Bielefeld University, 33594 Bielefeld, Germany

{lizhe, jannik, swachsmu, sagerer}@techfak.uni-bielefeld.de

<http://www.techfak.uni-bielefeld.de/ags/ai/>

**Abstract.** Different from many gesture-based human-robot interaction applications, which focused on the recognition of the *interactional* or the *pointing* gestures, this paper proposes a vision-based method for *manipulative* gesture recognition aiming to achieve natural, proactive, and non-intrusive interaction between humans and robots. The main contributions of the paper are an object-centered scheme for the segmentation and characterization of hand trajectory information, the use of particle filtering methods for an action primitive spotting, and the tight coupling of bottom-up and top-down processing that realizes a task-driven attention filter for low-level recognition steps. In contrast to purely trajectory based techniques, the presented approach is called object-oriented w.r.t. two different aspects: it is object-centered in terms of trajectory features that are defined relative to an object, and it uses object-specific models for action primitives. The system has a two-layer structure recognizing both the HMM-modeled manipulative primitives and the underlying task characterized by the manipulative primitive sequence. The proposed top-down and bottom-up mechanism between the two layers decreases the image processing load and improves the recognition rate.

## 1 Introduction

Recently, human-robot interaction is receiving more and more interest in the computer vision research community. With the development of cognitive robots which can serve humans as assistants or companions, a natural human-like communication between humans and robots comes into focus. As a consequence, the “hearing” as well as the “seeing” are becoming the most prominent and equally important modalities. In the near past, much work has been done in the area of gesture-based human-robot interaction (HRI) because of humans’ intensive use of their hands. Referring to Nehaniv’s gesture categorization [11], most of these approaches deal with *symbolic*, *interactional*, or *referential gestures* that have a communicative meaning on their own, i.e. they can be detected and described without considering the specific environment. In terms of Bobick’s taxonomy of *movements*, *activities*, and *actions* [2] they can be characterized as movements or, in more structured cases, activities. In this regard, object manipulations<sup>1</sup> are more complex because the hand trajectory needs to be interpreted in relation to the manipulated object. Due to Bobick this kind of context characterizes *actions*.

---

\* J. Fritsch is now with the Honda Research Institute Europe GmbH in Offenbach, Germany.

<sup>1</sup> Nehaniv refers to them as *manipulative gestures* [11].

In this paper, we aim at the vision-based recognition of simple actions that are defined by a non-deterministic sequence of object manipulations. As a manipulative gesture, this serves an important communicative function in human-robot interaction. First, the manipulation of an object draws the attention of the communication partner on the objects that are relevant for a performed task. Secondly, it serves the goal of a more pro-active behavior of the robot in passive, more observational situations. As Nehaniv states: “If the robot can recognize *what* humans are doing and to a limited extent *why* they are doing it, the robot may act appropriately” [11]. For example, in Fukuda’s work a cooking support robot is developed [6]. It can recognize human manipulations of objects by sensing the movements of the markers on the objects and give recommendations by speech or gesture. Dropping these kinds of artificial constraints, the recognition problem is becoming notoriously difficult. Assuming that a hand is manipulating a spatially near object, it becomes hard to decide if the object is just passed by the hand or manipulated. Besides this segmentation ambiguity, there is a large spatio-temporal variability of how hand trajectories reach different object types and the appearance of a hand trajectory in a 2D image will also heavily vary according to the position of the object and the view-angle. Finally, the mutual occlusion between the hand and the object causes even more difficulties for object detection and tracking.

In the present approach we will focus on two problems in the recognition of manipulative actions: (i) the segmentation ambiguity and (ii) spatio-temporal variability of the hand trajectory. We propose a unified graphical model with a two-layered recognition structure. On the lower layer, the object-specific manipulative primitives are represented as Hidden Markov Models (HMM) which are coupled with task-specific Markovian models on the upper level. A top-down processing mechanism predicts which kinds of objects are relevant according to the currently recognized tasks. Thereby, a dynamic attention mechanism is realized that reduces the number of considered objects and simplifies the segmentation task of the hand trajectory. Furthermore, the manipulative primitives are spotted by a particle filter (PF) realized HMM matching process. Due to an explicit modeling of an action abortion and resampling step, this method is more promising than traditional HMM forward-backward [13] processing and also could achieve more flexible transitions between model states than condensation-based trajectory recognition [1]. Afterwards, the results are fed back into the task level in order to predict the following primitives closing the bottom-up and top-down cycle.

## 2 Related Work

The concept of the action in the paper title is the same to that in Bobick’s categorization of motion recognition: movement, activity, and action [2]. It represents larger-scale events, which typically include interaction with the environment and causal relationships. In order to recognize these, more sophisticated schemes are needed that explicitly model such kind of contextual factors. Jo used a Finite State Machine (FSM) for modeling possible state transitions in the manipulative gesture [8]. Bobick developed a PNF (past-now-future) constraint network to model the temporal structure of actions and subactions [12]. These typically are pure semantic approaches, which have not used explicit motion models. In Chan’s work, a simple feature vector is used for modeling the

interaction primitive, e.g. *approach*. The transition of the semantic primitives are modeled by HMM [3]. Because of the early symbolic abstraction of trajectory information this method can only be applied in a restricted scenario. An approach that actually combines both types of information, sensory trajectory data and symbolic object data, in a structured framework is Moore's concept of objectspaces [10]. Here a camera mounted on the ceiling observes a human interacting with different objects. Certain image processing steps are carried out to obtain image-based, object-based, and action-based evidences for objects and actions, which are integrated using Bayesian networks. Action primitives are recognized from hand trajectories using HMMs that are trained offline on different activities related to the known objects. Our approach uses a similar object representation scheme but goes beyond this work because the spotting of meaningful parts in longer hand trajectories is seriously considered and a combined top-down and bottom-up mechanism solves the object attention problem. Furthermore, the proposed model enables the system to infer high-level intentions in the manipulative gesture detected.

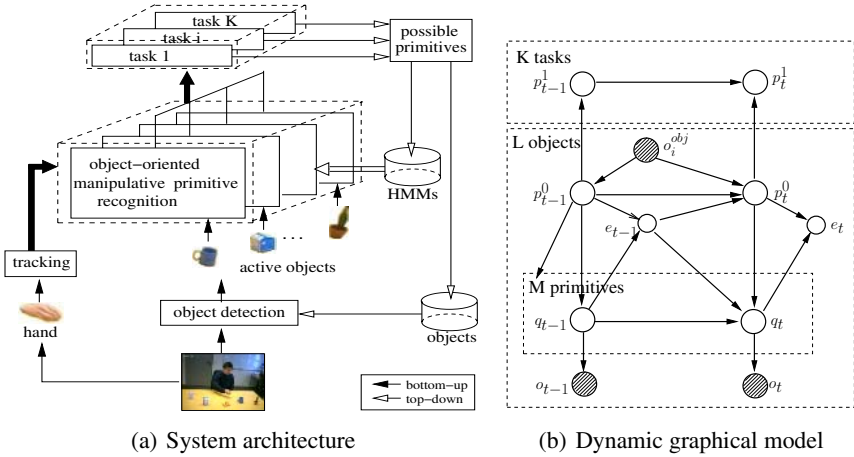
While these approaches center a context area around detected objects, hand-centered methods define context areas relative to a hand trajectory. Fritsch et al. [5] put forward such an approach. In this case, the trajectory information is augmented in each time step by contextual objects that are searched on-line using the context area bound to the moving hand. A hierarchical structure is used to model the manipulative sequence by [9]. In both works, the segmentation and spatio-temporal variability problems are coped with a particle filter. But the hand trajectory template, which is used as the primitive lacks the capability of generalization. For representing all possible hand trajectories in manipulation, a huge number of templates are needed.

Another specific application is presented by Yu et al. [15]. They argue that the eyes guide the hand in almost every action or object manipulation. In their work, the eye motion is measured by a head-mounted eye tracker and used for the segmentation of hand trajectories and the detection of objects. HMMs are used for action recognition which is purely based on trajectory information. Then object and action information is integrated on a symbolic level using action scripts.

### 3 Recognition System

In our definition, the manipulative action has two semantic layers. The bottom layer consists of the object-specific manipulative primitives. Each object has its own set of manipulative primitives because we argue that different object types serve different manipulative functions and even manipulations with the same functional meaning are performed differently on different objects. The top layer is used for representing the manipulative task, which are modeled by typical transitions between certain manipulative primitives. The system architecture and the dependency structure of the probabilistic model is shown in Figure 1. The architecture realizes a combined bottom-up top-down processing loop that utilizes the task-level prediction of possible primitives in order to restrict the object types possibly detected as well as the action primitives possibly recognized. In the bottom-up path, a processing thread is created for each detected object that consists of a trajectory segmentation, feature computation, and HMM-based recognition step. Thus, all three steps are performed differently for each object in parallel.





**Fig. 1.** Processing flow (a) and dependency structure of two time slices (b) in the recognition model. Each object-centered processing thread in (a) corresponds to one of the  $L$  plates in the dependency model.  $K$  is the number of different tasks modeled in the system and  $M$  is the number of possible primitives which each corresponds to one state of the variables  $p_t^0$  and  $p_t^1$ , respectively. The upper index of these variables denotes the primitive vs. task level.

Besides the detection of static objects, the hand is tracked over time and the trajectory information is passed to each object-centered processing thread. In the following, we will present both processing paths, bottom-up and top-down, in detail.

### 3.1 The Bottom-Up Process

This process builds up the bridge between the low-level image processing and the task knowledge by using HMMs for binding trajectory information to symbols representing action primitives. This subsection will explain the computation of low-level features and how these are matched to the models.

**Hand detection and tracking:** The hand is detected in a color image sequence by an adaptive skin-color segmentation algorithm (see [4] for detail) and tracked over time using Kalman filtering. The hand observation  $\mathbf{o}_t^{\text{hand}}$  is represented by the hand position  $(h_x, h_y)_t$  at time  $t$ .

**Preknowledge and detection of the objects:** The manipulative gesture is different to the face-to-face interactional gesture because it reflects the interaction between the human hand and the objects, not the pure hand movement with a meaningful trajectory. Therefore, a reliable detection of objects is crucial for the overall system performance. In order to avoid occlusion problems with interacting hands, we assume that a standard object recognizer, like the boosted cascade filter from Viola and Jones [14], is applied on the static scene. Then, object-dependent primitive actions are purely defined based on the hand trajectory that approaches an object instead of considering the object context while the object is being moved. If a moved object is applied to another object, the second object defines the object context. The observation vector of a detected object  $\mathbf{o}_i^{\text{obj}}$  contains its position  $(o_x, o_y)$ , a unique identifier (ID) for each different object type

in the scene and its height  $o_h$  and width  $o_w$ . As we can have several objects in the scene, the overall object observation vector contains multiple objects:

$$\mathbf{o}^{\text{obj}} = \{\mathbf{o}_1^{\text{obj}}, \dots, \mathbf{o}_i^{\text{obj}}, \dots, \mathbf{o}_L^{\text{obj}}\} \quad \text{with} \quad \mathbf{o}_i^{\text{obj}} = (o_x, o_y, \text{ID}, o_h, o_w). \quad (1)$$

A vicinity of an object is defined that is centered in the middle of the object detected and which is limited by the ratio  $\beta$  of its radius and the object size.

**Segmentation of trajectory:** Considering the possible occlusion in manipulation and the uncertainty in moving an object, a pre-segmentation step is only based on the vicinity of the static objects detected. A segment is started when the hand enters the vicinity or when an object is detected and the hand is already in the vicinity. The segment is ended when the hand goes out of the object's vicinity or when the object is lost after the hand moves away. As a consequence, the trajectory is segmented differently based on the different objects in the scene. To handle this multi-observation problem, one processing thread is started for each detected object. In the following, the processing in a single thread will be introduced. There, the final segmentation is directly coupled with the recognition step. The main goal of the pre-segmentation is to discard trajectories which are far away from the objects and contain less manipulative information.

**Interaction feature vector:** In the processing thread  $i$ , the interaction of the hand and the object is represented by a five-dimensional feature vector  $\mathbf{v}_f$  that is calculated from  $\mathbf{o}^{\text{hand}}$  and  $\mathbf{o}_i^{\text{obj}}$ . It contains the features: magnitude of hand speed  $v$ , change of the hand speed  $\Delta v$ , change of speed direction  $\Delta\alpha$ , distance  $r$  between the object and the operative hand scaled by object size, as well as the angle  $\gamma$  of the line connecting object and hand relative to the direction of the hand motion.

$$\mathbf{v}_f = (v, \Delta v, \Delta\alpha, r, \gamma) \quad (2)$$

The whole feature space is discretized into  $2 \times 2 \times 4 \times 3 = 48$  cells based on the following quantized dimensions:

$$\begin{aligned} \Delta v &: < 0, \geq 0; & \Delta\alpha &: < 90, \geq 90; & r &: [0..\beta/4), [\beta/4..\beta/2), [\beta/2..3\beta/4), [3\beta/4..\beta]; \\ (v, \gamma) &: v < v_{\text{threshold}}, (\geq v_{\text{threshold}}, < 90), (\geq v_{\text{threshold}}, \geq 90). \end{aligned}$$

These define the observation states for the following HMMs. The features are invariant with regard to translations, minor scale, and small rotations.

**HMM for manipulative primitive:** The typical manipulations related to one object type are named as the object-oriented manipulative primitives, e.g., “take a cup”. They are modeled by HMMs. Different to the normal parameter set  $\lambda = (A, B, \Pi)$  of an HMM, a terminal probability  $E$  is added. It reflects the terminal probability of an HMM given a state  $s_i$ . So the whole set consists of:

- $\Pi = \{\pi_i | \pi_i = P(q_1 = s_i)\}$ , initial probability of state  $s_i$ ;
- $A = \{a_{ij} | a_{ij} = P(q_{t+1} = s_j | q_t = s_i)\}$ , transition probability from state  $s_i$  to  $s_j$ ;
- $B = \{b_i(k) | b_i(k) = P(o_t = v_k | q_t = s_i)\}$ , probability of observing  $o_k$  given hidden state  $s_i$ ;
- $E = \{e_i | e_i = P(q_{\text{end}} = s_i)\}$ , terminal probability of state  $s_i$ .

The parameters are learned from manually segmented trajectories with the Baum-Welch algorithm,  $e_i$  is calculated similar to  $\pi_i$ , except using the last states.

**PF based HMM matching:** In order to spot the primitive from the pre-segmented trajectories, a PF called Sampling Importance Resampling (SIR) is used (better known

as CONDENSATION introduced by Isard and Blake [7]). The matching of the HMM and the observation are achieved by temporal propagation of a set of weighted particles:

$$\{(\mathbf{s}_t^{(1)}, w_t^{(1)}), \dots, (\mathbf{s}_t^{(N)}, w_t^{(N)})\} \quad \text{with} \quad \mathbf{s}_t^{(i)} = \{p_t^{0(i)}, q_t^{(i)}, e_t^{(i)}\}. \quad (3)$$

The number of particles is  $N$ . The sample  $\mathbf{s}_t^{(i)}$  contains the primitive index  $p_t^{0(i)}$ , the hidden state  $q_t^{(i)}$ , and the terminal state of this primitive  $e_t^{(i)}$  at time  $t$  (see Figure 1(b)). The resampling step reallocates a certain fraction of the particles with regard to the initial distribution  $\Pi$ . Consequently, the weight  $w_t^{(i)}$  of a sample can be calculated from

$$w_t^{(i)} = p(o_t | \mathbf{s}_t^{(i)}) / \sum_{j=1}^N p(o_t | \mathbf{s}_t^{(j)}). \quad (4)$$

The  $p(o_t | \mathbf{s}_t^{(i)})$  in it is the observation probability  $o_t$  given  $q_t^{(i)}$  and HMM  $p_t^{0(i)}$ . The propagation of the weighted samples over time consists of three steps:

**Select:** Selection of  $N - M$  samples  $\mathbf{s}_{t-1}^{(i)}$  according to their respective weight  $w_{t-1}^{(i)}$  and random initialization of  $M$  new samples. That means some particles which have high weights will be selected multiple times and some particles which have low weights will not be selected at all.

**Predict:** The current state of each sample  $\mathbf{s}_t^{(i)}$  is predicted from the samples from the select step according to the graphical model given in Fig. 1(b). The terminal state  $e_{t-1}^{(i)}$  is a bi-valued variable, 0 means the primitive is continuing and 1 means the primitive ends here. So if  $e_{t-1}^{(i)}$  is 0, the next hidden state  $q_t^{(i)}$  is sampled according to the transition probability of the HMM of primitive  $q_{t-1}^{(i)}$  and the primitive index  $p_t^{0(i)}$  keeps the same as  $p_{t-1}^{0(i)}$ . If the terminal state  $e_{t-1}^{(i)}$  is 1, the primitive index  $p_t^{0(i)}$  will be sampled according to the current possible primitives of this object. Then the hidden state  $q_t^{(i)}$  is sampled according to the initial probability of the HMM of the new primitive  $p_t^{0(i)}$ . At the end of this step, the terminal state of this particle  $e_t^{(i)}$  is sampled based on the terminal probability of the current primitive state  $q_t^{(i)}$ .

**Update:** Determination of the weights  $w_t^{(i)}$  of the predicted samples  $\mathbf{s}_t^{(i)}$  using Eq. 4.

The recognition of a manipulative primitive is achieved by calculating the *end-probability*  $P_{\text{end}}$  that a certain HMM model  $p_i$  is completed at time  $t$ :

$$P_{\text{end},t}(p_i) = \sum_n w_t^{(n)}, \quad \text{if } p_i \in \mathbf{s}_t^{(n)}. \quad (5)$$

A primitive model is considered recognized if the probability  $P_{\text{end},t}(p_k)$  of the primitive model  $p_k$  exceeds a threshold  $p_{\text{th}}^0$  which has been determined empirically.

The resampling step in the particle propagation is able to adapt the starting point of the model matching process if the beginning of the primitive does not match the beginning of the segment. The *end-probability* gives an estimation of the primitive's ending point. This combination to a certain extent solves the problem of the forward-backward algorithm which needs a clear segmentation of the pattern.

**Model of the tasks:** the manipulative tasks are modeled as the first-level Markovian process which is the same as Moore's definition [10]. Although this assumption vio-

lates certain domain dependencies, it is an efficient and practical way to deal with task knowledge. In the model  $A_i$  for a manipulative task  $i$ , a set of possible manipulative primitives  $P_i^1$  are defined, e.g., in the “prepare tea” task, the primitives “take cup”, “take tea can” could appear but not “take milk”. Because of the high effort needed for recording a huge amount of task sequences, the number of training examples for each complete task is too low for robustly estimating transition probabilities. Therefore, we model a task by a set of possible primitive pair transitions similar to a word pair grammar in automatic speech recognition. The set of transition rules  $A_i^1$ , the possible start symbols  $\Pi_i^1$ , and the set of possible end symbols  $E_i^1$  is learned from the output of the primitive recognition layer on a training set by thresholding the frequency of pairs observed in sequences of action primitives (see Sec. 4.2 for more details). Suppose the result from the manipulative primitive recognition is the sequence  $p_1^1, \dots, p_t^1$ . To calculate the acceptance of a task  $A_i = (P_i^1, \Pi_i^1, A_i^1, E_i^1)$ , only the primitives which are in the primitive list of the task  $A_i$  will be chosen because of the possible insertion in the primitive recognition.

$$(p_1^1, \dots, p_t^1 \mid p_j^1 \in P_i^1, j = 1 \dots t) \in \{\mathbf{P} \mid p_1^1 \xrightarrow{*}_{A_i^1} p_t^1, p_1^1 \in \Pi_i^1, p_t^1 \in E_i^1\} \quad (6)$$

where  $\mathbf{P}$  denotes the possible sequences from primitive  $p_1^1$  to  $p_t^1$  while considering transitions in  $A_i^1$ . Eq. 6 can easily be evaluated according to the parameter set  $A_i$ .

### 3.2 The Top-Down Process

Because of the object-specific primitive definition and its parallel processing for each affected object, the system confronts an attention problem when there are many objects appearing in the scene simultaneously. In order to solve this problem, a top-down process is introduced, in which the possible primitives coming next are predicted on the ground of the active task models and the previous results from the manipulative primitive recognition. This prediction is similar to the computation of a lookahead symbol in parsing strategies. For the prediction step different parsing alternatives are considered during the HMM matching process. For all primitives that gain an end probability  $P_{\text{end},t}(p_i) > 0$  a lookahead symbol is generated. If a primitive has been recognized this primitive is eliminated as a lookahead symbol. Because the predicted action primitives are specific for certain object types, the set of the next possibly manipulated object types can be calculated and be passed to the object detection component. This realizes a task driven attentional cue for early processing steps of the system (Fig. 1(a)). Additionally, the expectations from the predicted action primitives are used to restrict the HMMs applied in the PF based matching process.

## 4 Experiments and Results

In order to evaluate the quality of the manipulative gesture recognition, a scenario in an office environment has been designed as shown in Figure 2. A person is sitting behind a table and manipulates the objects that are located on it. She or he is assumed to perform one of three different manipulation tasks: (1) *water plant*: take cup, water plant, put cup; (2) *prepare tea*: consists of take/put cup, take tea can, pour tea into cup, put tea



**Fig. 2.** The office scenario used in the experiment

can; (3) *prepare coffee*: consists of take/put cup, take milk/sugar, pour milk/take sugar into cup, put milk. In the experiment, each task is performed 4-5 times by 8 different persons resulting in 36 sequences for each task and a total of 108 sequences. The images are recorded with a resolution of 320x240 pixels and with a frame-rate of 15 images per second. The object recognition results have been labeled because the evaluation experiment should concentrate on the performance of the action and task recognition. The object in the hand is ignored so that *pour milk into cup* and *pour tea into cup* are the same primitive actions. The scenario is restricted in so far that we assume a static camera, a known configuration of objects, and a camera view that is roughly orthogonal to the relevant movements.

#### 4.1 Manipulative Primitive Recognition

The first evaluation is used to test the performance of the object-oriented manipulative primitive recognition. There are five different objects used in the experiment: *tea can*, *milk*, *sugar*, *cup* and *plant*. Table 1 shows the primitives defined for each object type. The evaluation is done for all segments computed by the pre-segmentation step (see Section 3.1). These segments either contain a real manipulative primitive action which we call *positive segments* (PS) or contain just a hand passing by an object which we call *negative segments* (NS). For the positive segments, we calculate the false negative (FN) rate. For negative segments, the false positive (FP) rate is calculated. In order to achieve good performance results both rates should be low because both kinds of errors would seriously affect human-robot interaction. We randomly divided the 108 whole task sequences into a training set of 60, and a test set of 48 sequences. Because of the low number of training examples, we run the Baum-Welch algorithm used for

**Table 1.** The recognition results of the object-oriented manipulative primitives in both positive and negative segments

Objects	tea		milk		sugar	cup			plant
	take	put	take	put	take	take	put	pour	water
Num. PS	16	16	14	14	13	48	42	43	16
FN (%)	20.6 ±3.0	0.7 ±1.9	13.6 ±5.8	7.2 ±3.6	14.6 ±5.6	25.4 ±6.7	7.2 ±4.0	20.5 ±5.4	6.9 ±1.9
Num. NS	28		110		118	17			13
FP (%)	6.8 ± 2.0		9.8 ± 6.2		7.2 ± 8.7	17.6 ± 6.8			0

the HMM learning procedure 10 times with random initialization and give a standard deviation for the FN and FP rates. The results are computed using the parameter setting:  $N = 500$ ,  $M = 50$ ,  $p_{th}^0 = 0.2$ , and  $\beta = 3$ . From the results, it could be found that the *put* primitives are recognized with lower FN rate than the *take* and *pour* primitives because the variations of the latter two are much higher from person to person.

## 4.2 Manipulative Task Recognition

The second evaluation assesses the overall system performance. A manipulative task consists of the manipulative primitive sequence. However the ordering of the sequence is neither pre-determined nor completely fixed. For example some people may take sugar before taking milk, some will do it the other way around. But there probably will be an ordering between taking the cup and the watering action which needs to be learned from the data. For learning the possible transition pairs of each task model, the data set is divided into the set of 20 observation sequences, that was already used for learning the primitive action models, and a set of 16 sequences that are used for a one-leave-out experiment. Thus, each task model is learned from 35 task sequences in each experiment. The possible word pair transitions are extracted from the training data by a frequency threshold.

The task recognition results of the whole system are compared with (TD) and without (no TD) the top-down attention processing. The FN rate clearly shows a significant drop in case of top down processing for *prepare tea* and *prepare coffee*. Because sometimes an expected primitive was misrecognized in a way that was not covered by the task grammar, the rejection of these tasks caused relatively high FN rates but nearly no substitution errors (Sub.). The processing time for a 180-frame “prepare coffee” sequence with the former method is 54s running on MATLAB, which is much lower than the 86s needed by the pure bottom-up processing.

**Table 2.** The recognition results of the manipulative tasks with and without top-down processing

Name	Num.	FN(%, TD)	FN(%, no TD)	Sub.(%, TD)
<i>water plant</i>	16	2.5 ± 3.2	3.7 ± 4.4	2.5 ± 3.2
<i>prepare tea</i>	16	21.3 ± 4.4	37.5 ± 7.2	5.0 ± 5.3
<i>prepare coffee</i>	16	30.6 ± 6.2	38.1 ± 0.1	3.7 ± 3.2

## 5 Summary

The recognition of manipulative actions and tasks is an essential component for the natural, pro-active, and non-intrusive interaction between humans and robots. However, most techniques for the recognition of symbolic, interactional or referential gestures cannot be transferred because they ignore the object context and assume an object independent characteristic of the hand trajectory. Other approaches that focus on action recognition either use a pure semantic approach without considering motion models or simplify the trajectory segmentation problem in a pure bottom-up process.

The presented approach overcomes several of these deficiencies. The contextual objects are used for a pre-segmentation of the hand trajectory; the manipulative action

primitives are spotted by a particle filter approach that matches object specific HMMs in a more flexible way than the traditional forward-backward algorithm; tasks are defined by a set of possible transition rules similar to a word pair grammar that is automatically extracted from a small test set. By calculating a set of lookahead symbols on the task level, a task-driven attention filter is realized that tightly couples bottom-up and top-down processing. We were able to show first experiments that underline the potential of the presented approach. The action primitives were recognized quite robustly. The top-down attention filter significantly improves the computation time as well as the recognition performance. Further work will concentrate on an improved feature description of primitive actions, a more robust task model, and more sophisticated experiments.

## References

1. M. J. Black and A. D. Jepson. A probabilistic framework for matching temporal trajectories: Condensation-based recognition of gestures and expressions. In *European Conf. on Computer Vision, ECCV-98*, pages 909–924, Freiburg, Germany, 1998.
2. A. Bobick. Movement, activity, and action: The role of knowledge in the perception of motion. In *Royal Society Workshop on Knowledge-based Vision in Man and Machine*, 1998.
3. M.T. Chan, A. Hoogs, J. Schmiederer, and M. Petersen. Detecting rare events in video using semantic primitives with hmm. In *ICPR04*, pages IV: 150–154, 2004.
4. J. Fritsch. *Vision-based Recognition of Gestures with Context*. Dissertation, Bielefeld University, Technical Faculty, 2003.
5. J. Fritsch, N. Hofemann, and G. Sagerer. Combining Sensory and Symbolic Data for Manipulative Gesture Recognition. In *Proc. IEEE ICPR*, pages 930–933, Cambridge, UK, 2004.
6. T. Fukuda, Y. Nakauchi, K. Noguchi, and T. Matsubara. Time series action support by mobile robot in intelligent environment. In *Proc. IEEE Int'l Conf. Robotics and Automation*, pages 2908–2913, Barcelona, Spain, 2005.
7. M. Isard and A. Blake. Condensation – conditional density propagation for visual tracking. In *Int. J. Computer Vision*, pages 5–28, 1998.
8. K.H. Jo, Y. Kuno, and Y. Shirai. Manipulative hand gesture recognition using task knowledge for human computer interaction. In *Proc. Int'l Conf. on Automatic Face and Gesture Recognition*, pages 468–473, 1998.
9. Z. Li, N. Hofemann, J. Fritsch, and G. Sagerer. Hierarchical modeling and recognition of manipulative gesture. In *Proc. ICCV, Workshop on Modeling People and Human Interaction*, Beijing, China, 2005. IEEE.
10. D.J. Moore, I.A. Essa, and M.H. Hayes, III. Exploiting human actions and object context for recognition tasks. In *Proc. IEEE Int'l Conf. Computer Vision*, pages 20–27, 1999.
11. C. P. Nehaniv. Classifying types of gesture and inferring intent. In *Proceedings of the Symposium on Robot Companions: Hard problems and Open Challenges in Robot-Human Interaction AISB'05*, pages 74–81, Hatfield, UK, 2005.
12. C. S. Pinhanez and A. F. Bobick. Human action detection using pnf propagation of temporal constraints. In *Proc. IEEE CVPR*, pages 898–907, Washington, DC, USA, 1998.
13. L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Readings in speech recognition*, pages 267–296. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990.
14. P. Viola and M. Jones. Robust real-time object detection. In *Proc. IEEE Int. Workshop on Statistical and Computational Theories of Vision*, Vancouver, Canada, 2001.
15. C. Yu and D. H. Ballard. Learning to Recognize Human Action Sequences. In *2nd International Conference on Development and Learning (ICDL'02)*, pages 28–34, 2002.

# Detecting Intrinsically Two-Dimensional Image Structures Using Local Phase<sup>\*</sup>

Di Zang and Gerald Sommer

Cognitive Systems Group, Department of Computer Science  
Christian Albrechts University of Kiel, 24118 Kiel, Germany  
{zd, gs}@ks.informatik.uni-kiel.de

**Abstract.** This paper presents a novel approach towards detecting intrinsically two-dimensional (i2D) image structures using local phase information. The local phase of the i2D structure can be derived from a curvature tensor and its conjugate part in a rotation-invariant manner. By employing damped 2D spherical harmonics as basis functions, the local phase is unified with a scale concept. The i2D structures can be detected as points of stationary phases in this scale-space by means of the so call phase congruency. As a dimensionless quantity, phase congruency has the advantage of being invariant to illumination change. Experiments demonstrate that our approach outperforms Harris and Susan detectors under the illumination change and noise contamination.

## 1 Introduction

Local image structures play important roles in many computer vision tasks. They can be associated with the term intrinsic dimensionality [1], which, as a local property of multidimensional signal, expresses the number of degrees of freedom necessary to describe local structures. For 2D images, there exist three type of structures. The intrinsically zero dimensional (i0D) structures are constant signals. Intrinsically one dimensional (i1D) structures represent straight lines and edges. Corners, junctions, line ends, etc. are all intrinsically two dimensional (i2D) structures which all have certain degree of curvature. It is well know that these i2D structures are of high significance in object recognition, motion estimation, image retrieval, etc. Consequently, correct detection of i2D structures under image deformations is very important.

There exist a lot of work concerning the detection of i2D structures based on intensity information, see [2,3,4,5]. These intensity based approaches are sensitive to variations in image illumination. Hence, it is necessary to find some features of local structures which are invariant with respect to image brightness change for a robust and reliable detection. Phase is such a good candidate, which carries most essential structure information of the original signal and has the advantage of being invariant to illumination variation [6]. Detecting local structures can be

---

<sup>\*</sup> This work was supported by German Research Association (DFG) Graduiertenkolleg No. 357 (Di Zang) and DFG grant So-320/2-3 (Gerald Sommer).



realized by means of the phase congruency. Using phase congruency to detect edges has been reported in [7,8]. However, i2D structure detection based on its local phase has not yet been well investigated, although Kovessi proposed to use i1D local phase to detect i2D points by constructing the phase moments [9].

In this paper, we present a novel approach to detect i2D image structures using local phase information. The local phase of the i2D structure is derived from a curvature tensor and its conjugate part in a rotationally invariant way. By employing damped spherical harmonics as basis functions, the local phase is unified with a scale concept. The i2D structures can be detected as points of stationary phases in this scale-space by means of the so called phase congruency. Experimental results illustrate that our approach outperforms Harris and Susan detectors under illumination change and noise contamination.

## 2 Phase Estimation of Intrinsically Two-Dimensional Image Structures

The local phase of an i2D structure can be derived from a tensor pair, namely, the curvature tensor and its conjugate part, see also [10] for details. By employing damped 2D spherical harmonics [11] as basis functions, the local phase is unified with a scale-space framework. An  $n$ th order damped 2D spherical harmonic  $H_n$  has a much simpler representation in the spectral domain than that of the spatial domain. It takes the following form

$$H_n(\rho, \alpha; s) = \exp(in\alpha)\exp(-2\pi\rho s) = [\cos(n\alpha) + i \sin(n\alpha)]\exp(-2\pi\rho s) \quad (1)$$

where  $\rho$  and  $\alpha$  denote the polar coordinates in the Fourier domain,  $s$  refers to the scale parameter. The damped 2D spherical harmonics are actually 2D spherical harmonics  $\exp(in\alpha)$  combined with the Poisson kernel  $\exp(-2\pi\rho s)$  [8]. The first order damped 2D spherical harmonic is basically identical to the conjugate Poisson kernel [8]. When the scale parameter is zero, it is exactly the Riesz transform [12]. In order to evaluate the local phase information, the curvature tensor and its conjugate part are designed to capture the even and odd information of 2D image structures. Designing the curvature tensor is motivated by the second order fundamental theorem of the differential geometry, that is the second derivatives or Hessian matrix which contains curvature information of the original signal. Let  $f$  be a 2D signal, its Hessian matrix is correspondingly given by

$$H = \begin{bmatrix} f_{xx} & f_{xy} \\ f_{xy} & f_{yy} \end{bmatrix} \quad (2)$$

where  $x$  and  $y$  are the Cartesian coordinates. According to the derivative theorem of the Fourier theory [13], the Hessian matrix in the spectral domain reads

$$\mathcal{F}\{H\} = \begin{bmatrix} -4\pi^2 \rho^2 \frac{1+\cos(2\alpha)}{2} F & -4\pi^2 \rho^2 \frac{\sin(2\alpha)}{2} F \\ -4\pi^2 \rho^2 \frac{\sin(2\alpha)}{2} F & -4\pi^2 \rho^2 \frac{1-\cos(2\alpha)}{2} F \end{bmatrix} \quad (3)$$

where  $F$  is the Fourier transform of the original signal  $f$ . It is obvious that angular parts of the second order derivatives in the Fourier domain are related to 2D spherical harmonics of even order 0 and 2. Hence, these harmonics represent the even information of 2D structures. Therefore, we are motivated to construct a tensor  $T_e$ , which is related to the Hessian matrix. This tensor is called a curvature tensor, because it is similar to the curvature tensor of the second fundamental form of the differential geometry. This curvature tensor  $T_e$  indicates the even information of 2D image structures and can be obtained from a tensor-valued filter  $H_e$  in the frequency domain, i.e.  $T_e = \mathcal{F}^{-1} \{FH_e\}$ , where  $\mathcal{F}^{-1}$  means the inverse Fourier transform. Hence, the tensor-valued filter  $H_e$ , called the even filter reads

$$\begin{aligned}
 H_e &= \begin{bmatrix} \frac{H_0 + \text{real}(H_2)}{2} & \frac{\text{imag}(H_2)}{2} \\ \frac{\text{imag}(H_2)}{2} & \frac{H_0 - \text{real}(H_2)}{2} \end{bmatrix} = \begin{bmatrix} \frac{1 + \cos(2\alpha)}{2} & \frac{\sin(2\alpha)}{2} \\ \frac{\sin(2\alpha)}{2} & \frac{1 - \cos(2\alpha)}{2} \end{bmatrix} \exp(-2\pi\rho s) \quad (4) \\
 &= \begin{bmatrix} \cos^2(\alpha) & \frac{1}{2} \sin(2\alpha) \\ \frac{1}{2} \sin(2\alpha) & \sin^2(\alpha) \end{bmatrix} \exp(-2\pi\rho s)
 \end{aligned}$$

where  $\text{real}(\cdot)$  and  $\text{imag}(\cdot)$  indicate the real and imaginary parts of the expression.

In this filter, two components  $\cos^2(\alpha)$  and  $\sin^2(\alpha)$  can be considered as two angular windowing functions. These angular windowing functions provide a measure of the angular distance. From them, two perpendicular i1D components of the 2D image, oriented along the  $x$  and  $y$  coordinates, can be obtained. The other component of the filter is also the combination of two angular windowing functions, i.e.  $\frac{1}{2} \sin(2\alpha) = \frac{1}{2}(\cos^2(\alpha - \frac{\pi}{4}) - \sin^2(\alpha - \frac{\pi}{4}))$ . These two angular windowing functions yield again two i1D components of the 2D image, which are oriented along the diagonals. These four angular windowing functions can also be considered as four differently oriented filters, which are basis functions to steer a filter [14]. They make sure that i1D components along different orientations are extracted. Consequently, the even filter  $H_e$  enables the extraction of differently oriented i1D components of the 2D image.

The conjugate Poisson kernel, which evaluates the corresponding odd information of the i1D signal, is in quadrature phase relation with the i1D signal. Therefore, the odd representation of the curvature tensor, called the conjugate curvature tensor  $T_o$ , is obtained by employing the conjugate Poisson kernel to elements of  $T_e$ . Besides, the conjugate curvature tensor  $T_o$  results also from a tensor-valued odd filter  $H_o$ , i.e.  $T_o = h_1 * T_e = \mathcal{F}^{-1} \{H_1 H_e F\} = \mathcal{F}^{-1} \{H_o F\}$ , where  $h_1$  denotes the conjugate Poisson kernel in the spatial domain. Hence, the odd filter  $H_o$  in the spectral domain is given by

$$H_o = \frac{1}{2} \begin{bmatrix} H_1(H_0 + \text{real}(H_2)) & H_1(\text{imag}(H_2)) \\ H_1(\text{imag}(H_2)) & H_1(H_0 - \text{real}(H_2)) \end{bmatrix} \quad (5)$$

Similar as the Hessian matrix, we are able to compute the determinant of  $T_e$  and  $T_o$  for knowing the existence of the i2D structure. Combing the determinants of  $T_e$  and  $T_o$  results in a novel model for the i2D structure, which is called the monogenic curvature scale-space  $\mathbf{f}_{i2D}(\mathbf{x}; s)$ ,

$$\mathbf{f}_{i2D}(\mathbf{x}; s) = \det(T_e(\mathbf{x}; s)) + \det(T_o(\mathbf{x}; s)) \quad (6)$$

From it, the local amplitude for the i2D structure is given by

$$a(\mathbf{x}; s) = \sqrt{\det^2(T_e(\mathbf{x}; s)) + \det^2(T_o(\mathbf{x}; s))} \tag{7}$$

and the local phase can be obtained as

$$\varphi(\mathbf{x}; s) = \frac{\det(T_o(\mathbf{x}; s))}{|\det(T_o(\mathbf{x}; s))|} \text{atan} \left( \frac{|\det(T_o(\mathbf{x}; s))|}{\det(T_e(\mathbf{x}; s))} \right) \tag{8}$$

where  $\frac{\det(T_o(\mathbf{x}; s))}{|\det(T_o(\mathbf{x}; s))|}$  decides the local main orientation of the i2D structure. Hence, the local phase information of the i2D structure contains not only phase information but also the local main orientation. Therefore, the evaluation of the i2D structure can be realized in a rotation-invariant way.

### 3 Phase Congruency

Since the local phase is independent of the local amplitude, it thus has the advantage of being not sensitive to illumination change. Hence, detecting i2D image structures can be done by looking for points of stationary phase in the scale-space. This approach is commonly called phase congruency and is based on comparisons of the local phase at certain distinct scales [7]. In this paper, we take a similar idea as those reported in [7,9]. However, there are some differences. First, our local phase information can be evaluated in a rotation-invariant manner. Therefore, no orientation sampling is required. Second, the local phase directly indicates the phase information of the i2D structure. Thus, there is no need to construct principal moments of the phase congruency to determine i2D structures.

Morrone and Owens [15] define the phase congruency function in terms of the Fourier series expansion of a signal at a local position  $\mathbf{x}$  as

$$PC = \max_{\bar{\phi} \in (0, 2\pi]} \frac{\sum_n A_n \cos(\phi_n - \bar{\phi})}{\sum_n A_n} \tag{9}$$

where  $A_n$  represents the amplitude of the  $n$ th Fourier component,  $\phi_n$  denotes the local phase of the Fourier component at position  $\mathbf{x}$  and  $\bar{\phi}$  is the amplitude weighted mean local phase angle of all the Fourier terms at the position being considered. The measure has a value between zero and one. A phase congruency of value one means that there is an edge or a line, zero phase congruency indicates there is no structure. However, this measure results in poor localization and is also sensitive to noise. Hence, Kovesi [7] developed a modified version of the phase congruency. In this measure, the local phase is obtained from the logarithmic Gabor wavelet. Due to its lack of rotation invariance, orientation sampling must be employed to make sure that features at all possible orientations are treated equally. Hence, the new measure of phase congruency reads

$$PC = \frac{\sum_o \sum_n W_o [A_{no} (\cos(\phi_{no} - \bar{\phi}_o) - |\sin(\phi_{no} - \bar{\phi}_o)|) - T_o]}{\sum_o \sum_n A_{no} + \varepsilon} \tag{10}$$

where  $n$  and  $o$  refer to the scale parameter and the index over orientations, respectively. And  $W_o$  denotes a factor that weights for frequency spread along certain orientation and  $\varepsilon$  is added to avoid division by zero. The terms  $A_{no}$  and  $\phi_{no}$  are the local amplitude and local phase at a certain scale and orientation, respectively. The mean local phase at a certain orientation is represented as  $\bar{\phi}_o$ . Only energy values that exceed the estimated noise influence  $T_o$  can be taken into consideration. The symbols  $[$  and  $]$  indicate that the enclosed entity equals itself when its value is positive and zero otherwise. This new phase congruency measure produces a more localized response and it also incorporates noise compensation. However, the estimated local phase is only valid for the i1D signal. Hence, using phase congruency to detect i2D structures requires the construction of principal moments of the phase congruency, see [9].

In contrast to this, we have now a rotationally invariant evaluation of the local phase for the i2D structure, no orientation sampling is needed. Hence, the computation of phase congruency can be simplified as the following

$$PC = \frac{\sum_n W [A_n (\cos(\phi_n - \bar{\phi}) - |\sin(\phi_n - \bar{\phi})| - T)]}{\sum_n A_n + \varepsilon} \quad (11)$$

where  $n$  denotes the scale parameter,  $W$  is also a factor weighting for frequency spread,  $A_n$  and  $\phi_n$  represent the local amplitude and local phase of the i2D structure point, respectively. This new measure can be directly applied to detect i2D image structures. Any point with a phase congruency value higher than a certain threshold can be considered as an i2D point.

## 4 Performance Evaluation Criteria

In the literature, many detectors are designed for detecting i2D image structures. However, most of them show only qualitative experimental results. Because computer vision tasks require more robust and reliable detection results, there has been an increasing emphasis on quantitative performance evaluation. There also exists a number of research for assessing the detector performance. The measure suggested by Schmid et al. [16] is based on the idea of repeatability. Rockett [17] and Martinez-Fonte et al. [18] proposed a more empirical method for accessing. In their research, examples of true corners and non-corners are provided. For each threshold level, the corner detection probability and the false alarm rate are estimated to plot an ROC curve. In [19], Carneiro et al. assessed the detector performance by two measures, namely, the precision and recall rates.

The repeatability evaluation delivers the number of points repeated between two images with respect to the total number of detected points. However, this measure does not consider those correctly or wrongly detected points which do not repeat at all. The ROC curve plots the relation between the detection rate and false alarm rate with respect to the threshold variation, but it is not easy to show the detection performance with respect to image deformations like illumination change, rotation change and so on. In this paper, we follow the measures in [19].

The recall rate measures the probability of finding an i2D point in a deformed image given that it is detected in the reference image. The definition of the recall rate is given by

$$R = \frac{TP}{TP + FN} \tag{12}$$

where  $TP$  denotes the true positive and  $FN$  is the false negative. Since it is not easy to identify the ground truth, in this case, the true positive means the number of correctly matched points. Given a point  $\mathbf{x}_i$  in the reference image and a point  $\mathbf{x}_j$  in the deformed image, let  $M(\cdot)$  represent the deformation transform, if the Euclidean norm condition is satisfied, i.e.  $\|M(\mathbf{x}_i) - \mathbf{x}_j\| < 1.5$ , then these two points are correctly matched. False negative is the number of points in the reference image which cannot be matched with any points in the deformed image.

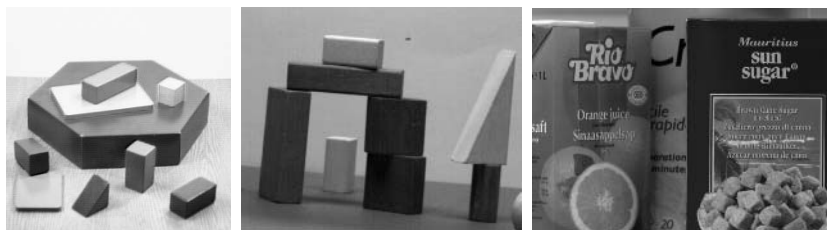
The precision rate indicates the probability that an i2D point detected in a deformed image is actually an i2D point in the reference image. Its definition reads

$$P = \frac{TP}{TP + FP} \tag{13}$$

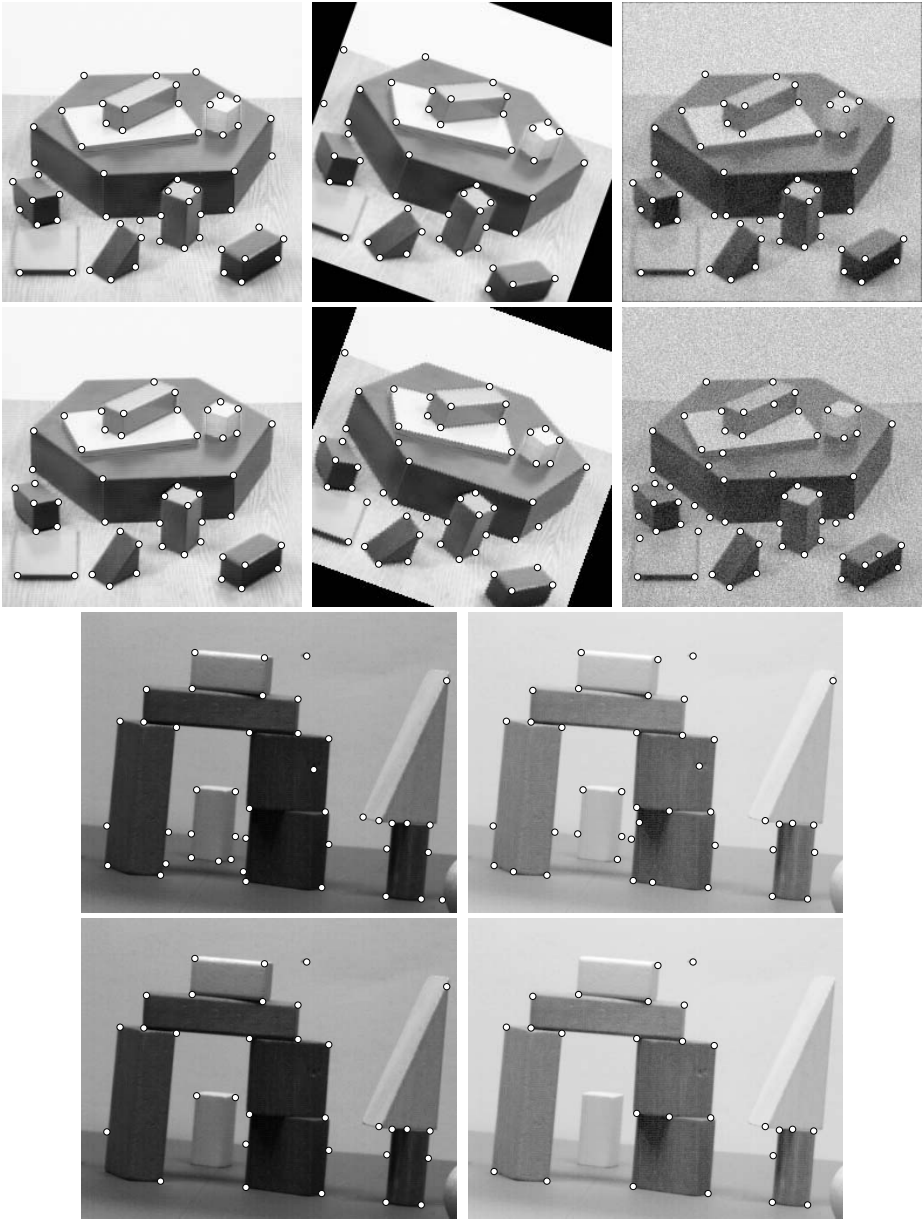
where  $FP$  is false positive, it means the number of points in the deformed image which cannot be matched with any points in the reference image. Both the recall and precision rates have values between zero and one. If the rate is higher, the detection performance is better.

## 5 Experimental Results

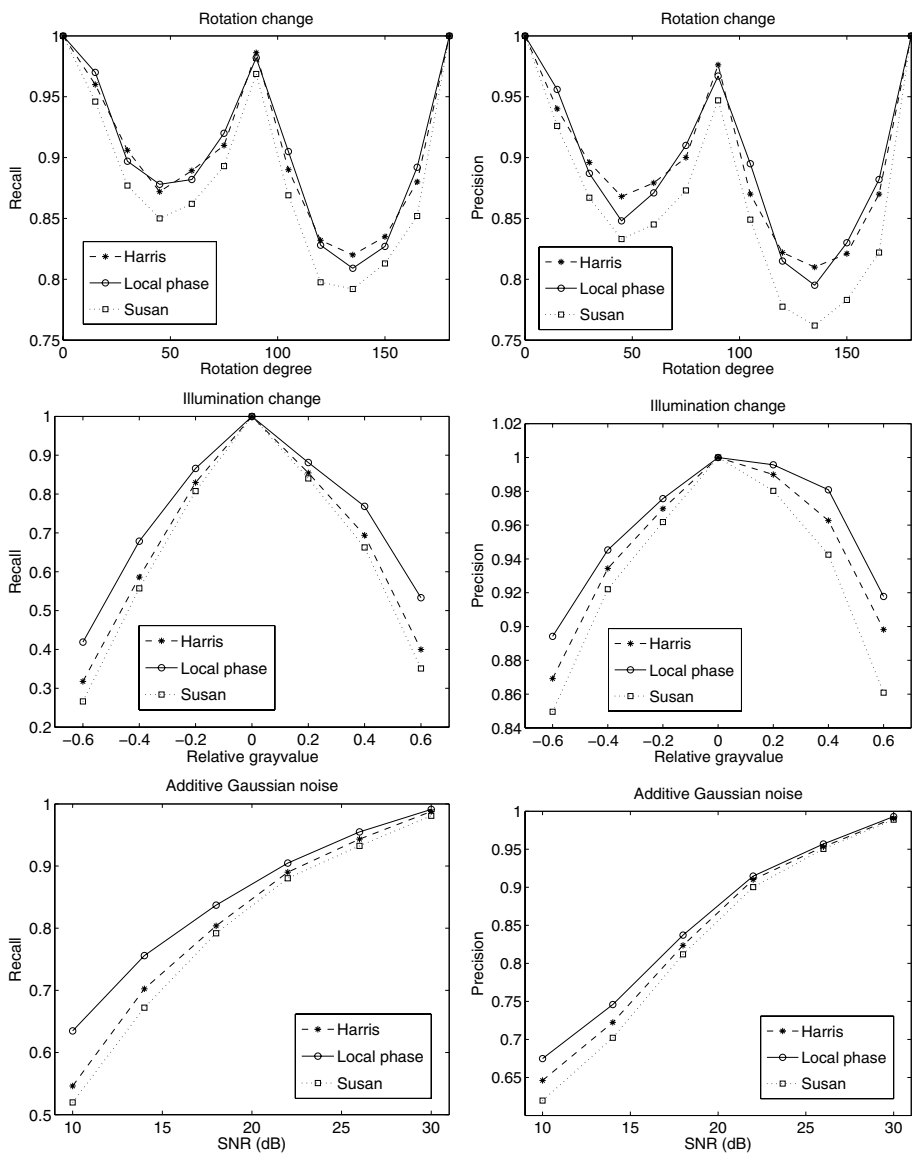
In this section, we present some experimental results. As shown in Fig. 1, two test images and one image sequence are employed for the experiments. The first experiment aims to illustrate some qualitative comparison results between our approach and the well-known Harris detector. The blox image is used for detection under the rotation change and the additive Gaussian noise contamination (standard derivation is 10). For the illumination change, we use the blocks image to show the detection difference. Fig. 2 demonstrates the detection results of our approach and the Harris detector under the rotation change, the noise contamination and the illumination change. According to the false positives and



**Fig. 1.** Two test images (blox and blocks) and one frame of a boxes image sequence



**Fig. 2.** Top row shows the detection results using our approach for the original image, the rotated image and the Gaussian noise contaminated image. The second row demonstrates the results from the Harris detector for the original image, the rotated one and the noise contaminated one. Results shown in the third row are detections for the original image and the illumination varied one by using our approach. Bottom row illustrates results from the Harris detector for the original image and the illumination changed one.



**Fig. 3.** First column: from top to bottom are recall rates under the rotation change, illumination variation and the additive Gaussian noise contamination. Second column: from top to bottom are precision rates under the rotation change, illumination variation and the additive Gaussian noise contamination.

false negatives, it can be shown that our approach performs better than the Harris detector when the illumination changes and the noise is added to some degree.

The second experiment is to show some quantitative comparison results. We follow the evaluation criteria of recall and precision rates to compare the performances of our approach, the Harris detector and also the well-known Susan detector. Ten frames of the boxes image sequence are employed for this experiment. Image deformations of rotation change, additive Gaussian noise contamination and the illumination variation are considered. For each deformation, the averaged values of ten frames are recorded to plot the recall and precision rates. Fig. 3 demonstrates comparison results between our approach, the Harris detector and the Susan detector according to the performance assessment criteria of recall and precision rates. Note that recall and precision rates have different scales for different image deformations. The top row shows detection results under the rotation change. Our approach has a comparable result with the Harris detector, and the Susan detector performs worse than these two approaches. Because of the discretization errors, curves for recall and precision show some local minima at rotations of 45 and 135 degrees. The second row are recall and precision rates for the illumination change. The phase congruency is a dimensionless quantity which is in theory invariant to the illumination change, although it is not absolutely invariant to brightness variation in practice, it is still less sensitive to the illumination variation than those intensity based approaches. Results indicate that our approach performs much better than the Harris and Susan detectors especially in the case of higher illumination change. Bottom row shows the additive Gaussian noise contaminated results. Since the phase congruency takes several scales into consideration and it also incorporates noise compensation, our approach demonstrates a better performance than that of the Harris detector. And the Harris detector is less sensitive to the noise when compared with that of the Susan detector due to the Gaussian smoothing in the local neighborhood.

## 6 Conclusions

We present a novel approach towards detecting i2D image structures using local phase information. The local phase of the i2D structure can be derived from a curvature tensor and its conjugate part in a rotation invariant manner. The i2D image structures are detected as those points with stationary phases in the scale-space by means of phase congruency. The recall and precision rates are employed as detection performance assessment criteria. Experimental results illustrate that our approach outperforms the Harris and Susan detectors when the illumination changes and the images are contaminated by the additive Gaussian noise. For the deformation of rotation change, our approach shows a comparable result with the Harris detector.



## References

1. Zetsche, C., Barth, E.: Fundamental limits of linear filters in the visual processing of two-dimensional signals. *Vision Research* **30** (1990) 1111–1117
2. Beaudet, P.: Rotationally invariant image operators. In: *Proceedings of International Joint Conference on Artificial Intelligence*. (1978) 579–583
3. Förstner, W., Gülch, E.: A fast operator for detection and precise location of distinct points, corners and centers of circular features. In: *Proc. ISPRS Intercommission Conference on Fast Processing of Photogrammetric Data, Interlaken, Switzerland* (1987) 281–305
4. Harris, C., Stephen, M.: A combined corner and edge detector. In: *Proceedings of 4th Alvey Vision Conference, Manchester* (1988) 147–151
5. Smith, S.M., Brady, J.M.: Susan – a new approach to low level image processing. *International Journal of Computer Vision* **23**(1) (1997) 45–78
6. Oppenheim, A.V., Lim, J.S.: The importance of phase in signals. *IEEE Proceedings* **69** (1981) 529–541
7. Kovese, P.: Videre: A image features from phase congruency. *Journal of Computer Vision Research* **1**(3) (1999)
8. Felsberg, M., Sommer, G.: The monogenic scale-space: A unifying approach to phase-based image processing in scale-space. *Journal of Mathematical Imaging and Vision* **21** (2004) 5–26
9. Kovese, P.: Phase congruency detects corners and edges. In: *Proc. The Australian Pattern Recognition Society Conference*. (2003) 309–318
10. Zang, D., Sommer, G.: The monogenic curvature scale-space. In Reulke, R., Eckardt, U., Flach, B., Knauer, U., Polthier, K., eds.: *Proc. of the 11th International Workshop on Combinatorial Image Analysis (IWCIA'06)*, Berlin. Volume 4040 of LNCS., Springer-Verlag, Berlin, Heidelberg (2006) 320–332
11. Felsberg, M.: Low-level image processing with the structure multivector. Technical Report 2016, Christian-Albrechts-Universität zu Kiel, Institut für Informatik und Praktische Mathematik (2002)
12. Felsberg, M., Sommer, G.: The monogenic signal. *IEEE Transactions on Signal Processing* **49**(12) (2001) 3136–3144
13. Papaulis, A.: *The Fourier Integral and its Application*. McGraw-Hill, New York (1962)
14. Freeman, W.T., Adelson, E.H.: The design and use of steerable filters. *IEEE Trans. Pattern Analysis and Machine Intelligence* **13**(9) (1991) 891–906
15. Morrone, M.C., Owens, R.A.: Feature detection from local energy. *Pattern Recognition Letters* **6** (1987) 303–313
16. Schmid, C., Mohr, R., Bauckhage, C.: Evaluation of interest point detectors. *International Journal of Computer Vision* **37**(2) (2000) 151–172
17. Rockett, P.: Performance assessment of feature detection algorithms: A methodology and case study on corner detectors. *IEEE Transactions on Image Processing* **12**(12) (2003) 1668–1676
18. Martinez-Fonte, L., Gautama, S., Philips, W.: An empirical study on corner detection to extract buildings in very high resolution satellite images. In: *Proc. of IEEE-ProRisc*. (2004) 288–293
19. Carneiro, G., Jepson, A.D.: Multi-scale phase-based local features. In: *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR03)*. (2003) 736–743

# Towards Unsupervised Discovery of Visual Categories

Mario Fritz and Bernt Schiele

Multimodal Interactive Systems, TU-Darmstadt, Germany  
{fritz, schiele}@mis.tu-darmstadt.de

**Abstract.** Recently, many approaches have been proposed for visual object category detection. They vary greatly in terms of how much supervision is needed. High performance object detection methods tend to be trained in a supervised manner from relatively clean data. In order to deal with a large number of object classes and large amounts of training data, there is a clear desire to use as little supervision as possible. This paper proposes a new approach for unsupervised learning of visual categories based on a scheme to detect reoccurring structure in sets of images. The approach finds the locations as well as the scales of such reoccurring structures in an unsupervised manner. In the experiments those reoccurring structures correspond to object categories which can be used to directly learn object category models. Experimental results show the effectiveness of the new approach and compare the performance to previous fully-supervised methods.

## 1 Introduction

Over the years various approaches have been proposed for the recognition of object categories often based on models learned directly from image data. The approaches, however, vary greatly in the specific task they address: from simple present/absent decision [1,2] over object class detection and localization [3] to pixel-level segmentation [4]. In this paper we deal with the problem of object detection and localization. Another difference between the proposed methods is the amount of supervision used and provided for the training data. The types of annotation varies from pixel-level segmentations [5], over bounding-box annotations [3] to unsupervised methods [2,6,7]. Very recent approaches for learning multiple categories do not even require the information which category is presented in which image [8]. While approaches using more supervision tend to require less training data, there is a clear desire to use less supervision typically at the price to use more unlabeled training data.

The central problem addressed in this paper is to discover and learn objects category models as reoccurring patterns of local appearance in sets of training data. It may seem quite unrealistic to discover object categories in this way. However, many appearance-based approaches explicitly or implicitly rely on the fact that both the local appearance as well as its structural layout exhibit reoccurring patterns that can be learned and modeled (e.g. [2,4,9]). A key idea

of our approach is therefore to discover reoccurring patterns in multiple images without the model of any particular object. Finding the locations and scales of such reoccurring structures effectively corresponds to unsupervised annotations of the training data. As we will show, the proposed approach enables effective object class discovery in unlabeled images. Using those estimated annotations a model of an object class can be learned.

Learning object models in an unsupervised fashion may be formulated in one single EM-loop as in e.g. Fergus et al [2]. In that method, appearance and structure are learned simultaneously making the learning computationally expensive and thus restricting the complexity of the model. Recently a new approach for object discovery has been proposed based on a pLSA-model [8]. Since the underlying model is a bag-of-words representation, the object discovery is based on local appearance alone neglecting structural information. [7] extends the initial approach to also include some structural information on top of the pLSA model, but the object discovery is still based on appearance only.

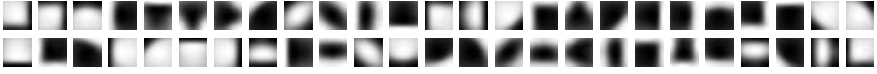
The main contributions of this paper are the following: First, we propose a novel scheme to discover object category members in images, which is based on the idea of estimating the locations and scales of reoccurring patterns. The estimates can be seen as an automatic annotation procedure of the training data. Second, we experimentally show the applicability of this idea for object discovery on several object classes. Third, we use the estimated annotations to learn object class models for object detection and localization. Fourth, we analyze the performance of such object class models on standard datasets.

The paper is organized as follows: Section 2 describes a method for locating reoccurring structure for which in Section 3 we present a method to robustly estimate the intrinsic scale of the associated objects. Section 4 shows how a model like [4] can be learnt from the estimated annotations. Finally, we show in Section 5 the usefulness of the obtained information on a image ranking and an object detection task.

## 2 Object Discovery

Our new approach to unsupervised object discovery is based on efficiently finding reoccurring spatial patterns of local appearances in a set of training images. We use a generic codebook representation which is also the basis of the object scale estimation procedure as presented in Section 3.

**Generic Codebook Representation.** Similar to other approaches for recognition [1], material classification [10] and detection [4], we use an initial clustering procedure to obtain a visual codebook. Since we do not want to assume a priori information on parts or common structure of the object category, we use a fixed generic codebook produced on unrelated background images. We extract image patches on the Caltech background images [11] using a scale-invariant Hessian-Laplace interest point detector [12]. Those image patches are clustered by k-means using normalized gray-scale correlation as similarity measure. The result looks as follows:



**Scale-Invariant Patterns.** We define a pattern  $\Psi_{k,r}$  in image  $k$  with reference point  $r$  to be characterized by a set of distributions  $\{p(h|\Psi_{k,r,c})\}_c$ . Each of the  $p(h|\Psi_{k,r,c})$  encodes the spatial distribution of the features in image  $k$  that match to a certain codebook  $c$ . The coordinates  $h = (h_x, h_y)$  are scale-normalized with the intrinsic feature scales  $\sigma$  (obtained from the scale-invariant interest point detector) and computed relative to a reference point  $r = (r_x, r_y)$

$$h = \left( \frac{x - r_x}{\sigma}, \frac{y - r_y}{\sigma} \right). \quad (1)$$

Using this scale-normalized coordinates is beneficial, as the pattern becomes characteristic for the particular reference point  $r$ . This allows to locate reoccurring patterns even though they appear at different global scales.

**Method.** We formulate the unsupervised discovery of reoccurring spatial patterns of local appearances as finding for each image the most likely pattern given all observed patterns in the training data. Therefore we are interested in finding the reference point  $\hat{q}_j$  associated with the most likely pattern in each image  $j$  given all observed patterns  $\Psi = \{\Psi_{k,r}\}_{k,r}$

$$\hat{q}_j = \arg \max_q p(\Psi_{j,q}|\Psi). \quad (2)$$

To simplify notation, the reference points  $q$  and  $r$  are assumed to be quantized. The likelihood estimate is obtained by marginalizing over the codebook entries  $c$ , scale-normalized coordinates  $h$ , reference points  $r$ , and images  $k$

$$p(\Psi_{j,q}|\Psi) = \sum_c \sum_h \sum_r \sum_k p(\Psi_{j,q,c}|h)p(h|\Psi_{k,r,c})p(\Psi_{k,r,c}).$$

Using Bayes' formula we obtain

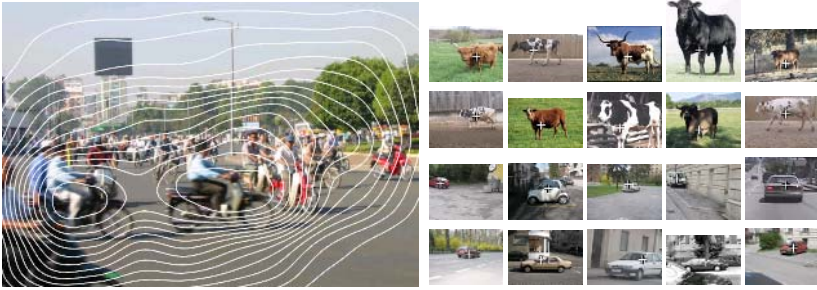
$$p(\Psi_{j,q,c}|h) = \frac{p(h|\Psi_{j,q,c})p(\Psi_{j,q,c})}{p(h)}. \quad (3)$$

By assuming uniform priors,  $p(\Psi_{k,r,c})$  and  $p(h)$  can be written as constant  $\frac{1}{Z}$ . This assumption is justified, by a uniform partitioning of our data using k-means clustering. Eq. 3 simplifies to

$$p(\Psi_{j,q}|\Psi) = \frac{1}{Z} \sum_c \sum_h \sum_r \sum_k p(h|\Psi_{j,q,c})p(h|\Psi_{k,r,c}). \quad (4)$$

An example of this likelihood estimate on the multi-scale TUD motorbikes [11] is overlaid on one of the images in Figure 1 as iso-lines. In this image we can clearly see two maxima which correspond to two motorbikes.

Eq. 4 can be interpreted as collecting evidence for pattern  $\Psi_{j,q}$  with respect to all other patterns  $\Psi$  by searching for matching feature with appearance  $c$



**Fig. 1.** (Left) Example of the computed likelihood on the multi-scale TUD motorbikes. (Right) Example result of our procedure for object discovery on car and cow images including varying position, scale and viewpoint and heterogeneous background.

and scale-normalized position  $h$ . Although this seems computationally infeasible, we introduce an efficient method to evaluate eq. 4 using scale-invariant feature hashing - similar to the idea of geometric hashing [13]. The idea is to index all features of the image database by quantized scale-normalized coordinates  $h$ , and store them in the hashes  $\mathcal{H}_c$  for each matching codebook cluster  $c$ . Features which are similar in appearance and scale-normalized position  $h$  are now stored in the same hash bin. More importantly, the matches can be used to backproject the support of all patterns  $\Psi_{j,q}$  with respect to all patterns. As a result, all  $p(\Psi_{j,q}|\Psi)$  given by the complex eq. 4 can be computed by a single loop over the hash bins of hashes  $\mathcal{H}_c$ .

**Evaluation.** To test the proposed procedure for object discovery with respect to robustness against translation, scaling, occlusion, and background clutter we ran tests on three object categories: motorbikes, cows, and cars. For the cows we used the training set of the TUD cows [11], as well as the cows from [14]. For the cars we used the training set of the PASCAL challenge [11]. Examples for the estimated object centers are shown in Figure 1. Despite the strong variations in appearance and view-point, the objects were successfully localized. The reference locations were quantized on a  $10 \times 10$  grid.

To gain more insights, we perform a more detailed quantitative analysis on the Caltech motorbike training set [11] which consists of 400 images. We compute the distance between our estimate and the center of the groundtruth bounding box annotation normalized by the object width. The average distance is 0.10, which we consider to be very good, as the groundtruth annotations are not really accurate themselves. Nearly all errors are below a normalized distance of 0.3, which is well below the noise level assumed in the evaluation of the scale estimation method in Section 3.

### 3 Object Scale Estimation

From the procedure for object discovery described in the previous section we obtain localized patterns  $\Psi_{j,q}$  at reference points  $\hat{q}_j$  for each image  $j$ . However, since these reoccurring patterns are obtained in a scale-invariant fashion, they

are of unknown scale  $s$ . While it is advantageous, that no explicit knowledge of the object scale is required for discovering reoccurring patterns, tasks like training an object model for detection need an estimate of the object scale to learn a model across the training instances.

**Method.** The proposed method matches scale-invariant patterns to collect evidence for their associated global object scale. Different methods to obtain a robust estimate are proposed and evaluated. As the absolute, global object scale only exists with respect to a reference scale, we formulate the scale estimation problem as finding the pairwise relative scale  $\hat{\rho}_{k,l} = s_k/s_l$  between two discovered patterns  $\Psi_k$  and  $\Psi_l$  in a pair of images  $k$  and  $l$ . In analogy to eq. 2 we describe the problem of finding the most likely relative scale  $\hat{\rho}_{k,l}$  with respect to the two patterns of the image pair as

$$\hat{\rho}_{k,l} = \arg \max_{\rho_{k,l}} p(\rho_{k,l} | \Psi_k, \Psi_l) \quad (5)$$

We assume that for matching features the ratio of the intrinsic scale  $\sigma$  of the matched structures is equal to the ratio of the global scales  $s$  between the patterns and their associated objects  $\rho_{k,l} = s_k/s_l = \sigma_k/\sigma_l$ . According to this we factor eq. 5 and marginalize over the codebook entries  $c$  and the scale-normalized coordinates  $h$

$$p(\rho_{k,l} | \Psi_k, \Psi_l) = \sum_{\sigma_l} p((\rho_{k,l}\sigma_l) | \Psi_k) p(\sigma_l | \Psi_l) = \sum_c \sum_h \sum_{\sigma_l} p((\rho_{k,l}\sigma_l), h | \Psi_{k,c}) p(\sigma_l, h | \Psi_{l,c})$$

As in Section 2 we store all features in the hashes  $\mathcal{H}_c$ . Our efficient data structure allows to compute all these likelihoods in one loop over the hash bins.

The estimates from eq. 5 can be interpreted as a fully connected graph, where the patterns in the images are the nodes and the relative scales of the patterns are attached to the edges. To make our method robust with respect to outliers, we compute confidence scores for all estimated relative scales. These are computed by indentifying image triplets with consistent relative scale estimates: Given three images  $I_a, I_b, I_c$  with their relative scales  $\rho_{a,b}, \rho_{b,c}, \rho_{a,c}$ , the confidence for all three scale estimates is increased if the equation  $\rho_{a,b}\rho_{b,c} = \rho_{a,c}$  is fulfilled.

In this paper we investigate three different methods to derive a unique scale estimate for each pattern from the pairwise relative scale information: *least squares*, *maximum spanning tree*, and *min-linkage method*.

The *least squares method* is based on a linear system of equations to estimate the unknown scales without using the computed confidences. Considering two patterns  $\Psi_k, \Psi_l$  with the global scale of the patterns  $s_k, s_l$  of the associated object instances, we compute a least-squares fit for the global scales  $s$  from all the estimated relative scale according to:

$$\frac{s_k}{s_l} = \rho_{k,l} \implies \log s_k - \log s_l = \log \rho_{k,l}. \quad (6)$$

This method is computational expensive, because the number of equations grows quadratically in the number of images, and its estimates are sensitive to outliers.

The *maximum spanning tree method* computes a maximum spanning tree on the graph of confidences. The scale estimates can be directly computed from this

tree by fixing one scale. Although this method has low computational complexity, the estimates are rather unstable, as shown in Section 3.

As a compromise between efficient computation and robust estimation, we propose a third method. The *min-linkage method* considers for every image the  $n$  most confident relative scales to all other images and therefore the number of equations grows only linearly with the number of images. The estimate of the scales is still robust due to the least-squares estimation.

The above described methods estimate relative scales, however, for the detection experiments (Section 5) an absolute scale based on the extent of the object is required. One possibility is to specify a reference scale for one image. In the experimental evaluation it turned out that this is not necessary, as the absolute object radius can be chosen to be twice the mean feature distance to the center *after* aligning all objects with the computed relative scale estimates.

**Evaluation.** To evaluate the accuracy of our new scale estimation scheme, we again use the Caltech motorbike database with annotated object centers, which has a scale variation of 2 octaves. The mean deviation of the estimated scales from the true scales is roughly  $\frac{1}{9}$  of an octave for the least-squares and  $\frac{1}{5}$  for the min-linkage method (minimum linkage preserves 40 = 10% of the most confident scales). Additionally, we evaluated the robustness of the system with respect to Gaussian noise in the center point annotation. Even when the noise is amplified until the  $3\sigma$ -radius reaches  $\frac{2}{3}$  of the object radius - which is twice the noise level we measured for the center point estimate in Section 2 - the mean deviation of the estimated scales from the true scale is roughly  $\frac{1}{4}$  of an octave for least-squares and minimum linkage. The maximum spanning tree method reaches this error already at half the noise level. As a conclusion we use the minimum linkage method in our following experiments, as it shows about the same accuracy as the full least-squares, but with a much lower computational cost.

## 4 Model Estimation for Detection

**Object Category Detection with ISM.** The *Implicit Shape Model (ISM)* [5] is a versatile framework for scale-invariant detection of object categories, which has shown good performance on challenging detections tasks [11]. It uses a flexible non-parametric representation for modeling visual object categories by spatial feature occurrence distributions with respect to a visual codebook. For details we refer to [5]. Additionally the method allows for back-projecting the support of the hypotheses to infer figure-ground segmentation masks and performing an MDL-based reasoning to resolve multiple and ambiguous hypotheses [4]. However, the generation of an object specific visual codebook and the MDL-based reasoning step require figure-ground segmentations for the training images which introduce high annotation effort.

**Unsupervised Learning of Models for Detection.** One of our contributions is to show that one can achieve high recognition performance by using the estimated center point (Section 2) and scale (Section 3) instead of manually produced segmentations. As we do not have a detailed segmentation mask at

our disposal when using those location and scale estimates, we use a simple but (as will be seen in the experiments) effective approximation. Figure 3 shows our rough approximation by assuming the segmentation to be a circle specified by the estimated center point and scale.

To learn the ISM model, we first switch from the generic codebook (Section 2) to an object specific SIFT representation [15] computed on Hessian-Laplace interest points [12]. We use the approximated segmentation (circles) to determine the object features for clustering. Given the approximated segmentation and the new codebook, we can proceed training the ISM as described in [5]. Despite the crude approximation of the segmentations with circles, it is possible to infer segmentations for the hypothesis on test images as shown in Figure 3.

## 5 Experiments

Whereas the previous sections analyzed the proposed object discovery and object scale estimation separately, this section shows the applicability to image ranking and object category detection. While the ranking task also shows the scalability to large numbers of images, the detection experiments evaluate how the proposed method generalizes to different categories. In addition we will show that the approximated segmentation masks from Section 4 are effective and even crucial to obtain high level detection performance.

**Image Ranking.** In the following, experiments we show that the proposed method for unsupervised object discovery from Section 2 can be used on its own for an image ranking task. Using a keyword search for motorbikes we downloaded 5246 images containing a wide range of different motorbike types (e.g. cruiser, sportbike, touring, scooter, moped, off-road, combination) captured from different viewpoints. Naturally quite a number of those images only show close-ups, parts or even unrelated objects. Our task is to sort these images out. We use our method for object discovery to rank the images by the likelihood (eq. 4). Note, that this ranking is obtained in an totally unsupervised and no validation set as in [7] is needed. Figure 4(left) shows the ROC curves obtained by running our approach with and without spatial information. If the spatial information of the features is discarded, our representation reduces to a bag-of-words representation. The use of spatial information improves the results significantly, which demonstrates the improvement of our model over purely appearance-based approaches. Qualitative results for our new approach using appearance and spatial structure are shown in Figure 2. As scooters were the dominating motorbike type in the set (1169 of 5246), they also appear first in the ranking.

**Visual Category Detection Task.** In the detection experiments we train a model according to Section 4 and use it to localize objects in the test images. Detections are only accepted as correct if the hypothesized bounding box fits the groundtruth annotation. Multiple detections are counted as false positives. For better comparability we use the acceptance criterion described in [16]. We want to emphasize, that no parameters had to be tuned for the proposed approach for unsupervised learning. In terms of efficiency, the approach for object discovery





**Fig. 2.** The proposed method for Object Discovery also facilitates ranking of the images. (left) best ranked images (right) worst ranked images.

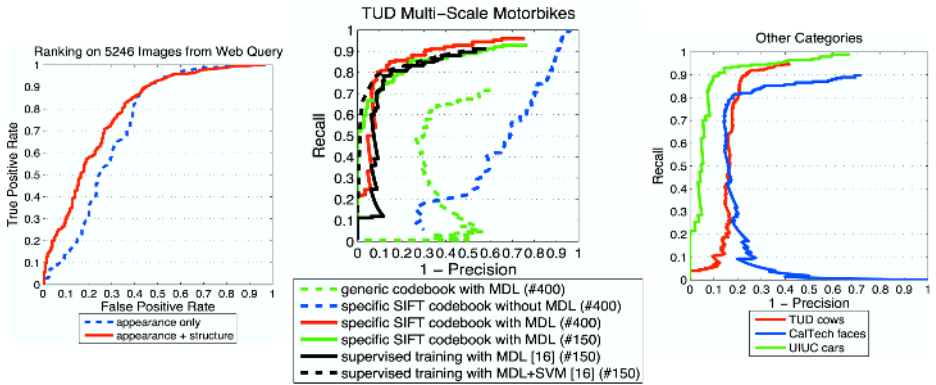


**Fig. 3.** (a) training image (b) estimated approximation of object segmentation (c) test image (d)+(e) inferred segmentation for hypothesis (f) final detections

can estimate object locations in 200 images in 11 minutes on a 3Ghz Pentium4, whereas the object scale estimation takes 6 minutes.

*Unsupervised Learning for Detection.* Figure 4(middle) shows results on the multi-scale TUD motorbike test set [11], which includes significant scale variations, partial occlusions and multiple instances per image. The models are trained on the Caltech motorbikes [11]. The best results published on this data-set are 80% EER using accurate pixel-level segmentation and ISM (supervised training with MDL) and 81% adding an additional SVM-stage (supervised training with MDL+SVM) [16]. Quite importantly, the performance of the proposed unsupervised object discovery method (specific SIFT codebook with MDL - 150) is very similar to the supervised training of ISM. The EER of 81% can be further increased to 84% by using 400 instead of 150 training images (again in an unsupervised fashion) and which is the best performance presented so far on the test set. Compared to the SVM approach [16] the precision is slightly worse, but the achievable recall is higher. So adding an SVM classifier in a similar fashion has the potential to further increase the overall performance. Overall the results are highly encouraging as they indicate that high annotation effort can be replaced by using a larger amount of training data.

*Evaluation of the Approximated Segmentation Masks.* Figure 3 shows a test image with the estimated segmentation masks and the final detections. While the mask is far from being perfect, the computed support of the hypotheses is approximately correct. Figure 4(middle) shows how the performance increases significantly when this approximation is used to perform the MDL-based hypothesis verification. The results support our claim, that the estimated segmentation masks are accurate enough and facilitate the training of a model that gives competitive performance. The figure also shows the importance of switching to an object class specific SIFT codebook (Section 4).



**Fig. 4.** (left) ROC-curve of ranking task (middle) performance comparison to supervised baseline (right) generalization to other categories and data sets

*Generalization to other Categories.* To investigate how this approach generalizes to other categories and compare our method to previous work, we conduct experiments on cows, faces, and cars. The results are reported in Figure 4(right). The training sets TUD cows and Caltech faces [11] are selected, as they include a significant amount of variation of the object position in the training data to underline the performance of the proposed method for object discovery. For the cows we use the same test setting as in the supervised approach of [16]. Our unsupervised approach achieves an equal error rate performance of 79.9% whereas the supervised reference achieved 93.2% [16]. As the background is for some training images the same, we learnt it as reoccurring structure. As it is part of the model, we get some strong hypotheses on these background structures which also occur in the test set and that are responsible for the decrease in performance. On the UIUC car and caltech face database we compare to the unsupervised method of Fergus [2]. On the cars we get an equal error rate performance of 89.5% in comparison to 88.5% in [2] using the same evaluation criterion. We achieve this performance training on only 50 car images and their mirrored versions from the TUD car database [11]. The best performance on this dataset is reported by the supervised method in [4] achieving 97% equal error rate performance. In [17] a detection performance for the model of [2] of 78% equal error rate is presented on the caltech face database. Our approach achieves a significant improvement by an equal error rate performance of 81.1%.

## 6 Conclusion

We have proposed an efficient and flexible framework for discovering visual object categories in an unsupervised manner which makes use of appearance and spatial structure at the same time. The approach is based on two new components for object discovery and object scale estimation, that extract information about reoccurring spatial patterns of local appearance. The experimental results show that our system facilitates unsupervised training of an model for object

class detection that has equal or even better performance than previous unsupervised approaches. In addition, the method was used to rank images without any supervision or validation. Results are presented on a large image database of over 5000 images including a significant amount of noise. Finally, we obtained comparable results w.r.t. a strongly supervised state-of-the-art detection system on a challenging multi-scale test set. We showed that we can compensate for the decrease in performance by adding more training examples, which results in the best performance shown so far on this test set.

**Acknowledgments.** This work has been funded, in part, by the EU project CoSy (IST-2002-004250).

## References

1. Csurka, G., Dance, C., Fan, L., Willarnowski, J., Bray, C.: Visual categorization with bags of keypoints. In: ECCV'04 Workshop on Stat. Learn. in Comp. Vis., Prague, Czech Republic (2004) 59–74
2. Fergus, R., Perona, P., Zisserman, A.: Object class recognition by unsupervised scale-invariant learning. In: CVPR'03, Madison, WI (2003)
3. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: CVPR'01. (2001)
4. Leibe, B., Leonardis, A., Schiele, B.: Combined object categorization and segmentation with an implicit shape model. In: ECCV'04 Workshop on Stat. Learn. in Comp. Vis., Prague, Czech Republic (2004) 17–32
5. Leibe, B., Schiele, B.: Scale invariant object categorization using a scale-adaptive mean-shift search. In: DAGM'04, Tuebingen, Germany (2004)
6. Winn, J.M., Jojic, N.: Locus: Learning object classes with unsupervised segmentation. In: ICCV. (2005) 756–763
7. Fergus, R., Fei-Fei, L., Perona, P., Zisserman, A.: Learning object categories from google's image search. In: ICCV'05, Beijing, China (2005)
8. Sivic, J., Russell, B.C., Efros, A.A., Zisserman, A., Freeman, W.T.: Discovering objects and their locations in images. In: ICCV'05, Beijing, China (2005)
9. Felzenszwalb, P.F., Huttenlocher, D.P.: Pictorial structures for object recognition. *IJCV* **61**(1) (2005) 55–79
10. Leung, T., Malik, J.: Representing and recognizing the visual appearance of materials using three-dimensional textons. *IJCV* **43**(1) (2001) 29–44
11. Pascal: The PASCAL Object Recognition Database Collection (2005) <http://www.pascal-network.org/challenges/VOC>.
12. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.* **27**(10) (2005) 1615–1630
13. Wolfson, H.J., Rigoutsos, I.: Geometric hashing: An overview. *IEEE Comput. Sci. Eng.* **4**(4) (1997) 10–21
14. Hillel, A.B., Hertz, T., Weinshall, D.: Efficient learning of relational object class models. In: ICCV'05, Beijing, China (2005)
15. Lowe, D.: Distinctive image features from scale-invariant keypoints. *IJCV* **60**(2) (2004) 91–110
16. Fritz, M., Leibe, B., Caputo, B., Schiele, B.: Integrating representative and discriminant models for object category detection. In: ICCV'05, Beijing, China (2005)
17. Fergus, R., Perona, P., Zisserman, A.: Weakly supervised scale-invariant learning of models for visual recognition. (2005) Under review

# Cross-Articulation Learning for Robust Detection of Pedestrians

Edgar Seemann and Bernt Schiele

Technical University of Darmstadt  
lastname@mis.tu-darmstadt.de

<http://www.mis.informatik.tu-darmstadt.de>

**Abstract.** Recognizing categories of articulated objects in real-world scenarios is a challenging problem for today's vision algorithms. Due to the large appearance changes and intra-class variability of these objects, it is hard to define a model, which is both general and discriminative enough to capture the properties of the category. In this work, we propose an approach, which aims for a suitable trade-off for this problem. On the one hand, the approach is made more discriminant by explicitly distinguishing typical object shapes. On the other hand, the method generalizes well and requires relatively few training samples by cross-articulation learning. The effectiveness of the approach is shown and compared to previous approaches on two datasets containing pedestrians with different articulations.

## 1 Introduction

In recent years a large number of approaches have been proposed for the detection of object categories in still images. Categories of non-rigid objects such as pedestrians have proven to be particularly challenging. The high intra-class variability, which is caused by global appearance changes and object articulations, requires recognition approaches that are both highly discriminative and also generalize well.

In the literature, several approaches focus on the global structure of the object [5,11,17,2], while others detect individual parts [4,3,10,8,16]. Gravila [5] uses a hierarchy of object silhouettes and applies Chamfer matching to obtain detection hypotheses. Papageorgiou & Poggio [11] train an SVM based on wavelet features. Zhao & Thorbe [17] perform detection with a neural network and exploit stereo information to pre-segment images. Dalal & Triggs [2] compute a global gradient-based descriptor, similar to SIFT, to train a linear SVM. Forsyth & Fleck [4] introduce the general methodology of body plans for finding people in images. Felzenszwalb and Huttenlocher [3] learn simplistic detectors for individual body parts. Ronfard *et al.* [19] extended this work by using stronger classifiers such as SVMs. Mohan and Papageorgiou [10] apply the wavelet-based detectors from [11] to detect body parts and then use body geometry to infer a person's position and pose. Viola *et al.* [16] use simple local features and a boosting scheme to train a cascade of classifiers. Mikolajczyk *et al.* train body part classifiers with

boosting and combine them in a probabilistic framework. In this work, instead of modeling individual object parts, we identify and model typical object articulations or shapes. These typical shapes are learnt automatically from motion segmentations, which can be computed from video sequences with a Grimson-Stauffer background model [14]. The advantage of this approach is that we do not need manual labeling of object parts.

The main contributions of this paper are the following. We introduce a novel scheme to learn the relationship between arbitrary object parts or, as we call it, *local contexts* and the global object shape. As a result, we obtain an approach, which captures large appearance variations in a single model and implements a suitable trade-off between generalization performance and discriminative power of the model. The method is able to share features between typical object shapes and therefore requires relatively few training images. In a sense, the approach generalizes the idea of sharing features [15] to the sharing of local appearance across object instances and shapes. A thorough evaluation shows that the proposed model outperforms previously published methods on two challenging data sets for the task of pedestrian recognition.

## 2 Recognition Algorithm

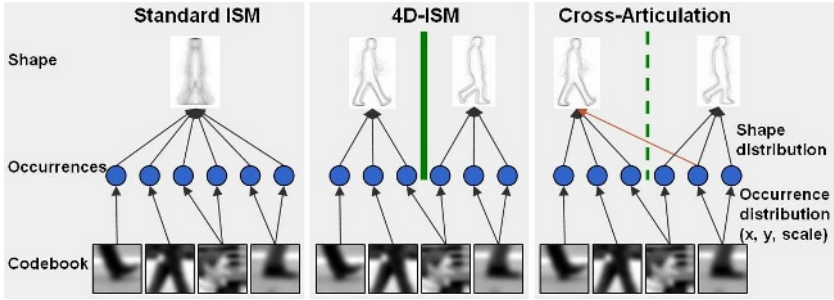
The recognition approach proposed in this paper extends the Implicit Shape Model (ISM) developed by Leibe & Schiele [6]. This section introduces the basic algorithm and discusses extensions to explicitly handle global appearance changes and object articulations.

### 2.1 Standard ISM

The ISM is a voting framework, which accumulates local image evidences to find the most promising object hypotheses. It is capable of multi-scale detection and pixel-wise segmentation masks can be inferred for each hypothesis. An additional reasoning step based on the Minimum Description Length (MDL) principle makes the method more robust in the presence of clutter and overlapping objects. The following gives a brief overview of the methods.

**Codebook Representation.** For representing an object category with an ISM, a *codebook* or visual vocabulary of local appearances is built [6]. Therefore, a scale-invariant interest point detector is applied to each training image and local descriptors are extracted. These descriptors are subsequently clustered with an agglomerative clustering scheme. The resulting set of local appearances represents typical structures on an object category.

**Spatial Occurrence Distribution.** Once a *codebook* on an object category has been learnt, we model the spatial occurrence distribution of its elements. In order to do this, we record all locations (x-, y-position and scale) on which a codebook entry matches the training instances.



**Fig. 1.** Schematic overview of the different object models. Both the standard ISM and 4D-ISM models are special cases of the proposed approach. By learning the *shape distribution* from local contexts, we combine the strength of the two other models.

**Hypotheses Voting.** In the recognition process, we apply the same feature extraction procedure as during training. Thus, we obtain a set of local descriptors at various scales on the test image. Each extracted descriptor casts votes for object hypotheses in a probabilistic extension of the generalized Hough transform. The maxima of the 3D voting space  $(x, y, scale)$  are back-projected to the image to retrieve the supporting local features of each hypotheses. We present details on an improved version of this probabilistic formulation when we introduce the extensions to deal with different object shapes in section 2.2.

## 2.2 Consistent Shape Voting

Figure 1 shows a schematic illustration of the standard ISM-model on the left. While the ISM allows for cross-instance learning and therefore requires relatively little training data it has no notion of possible object articulations within the category. Local appearances are learnt from all possible variations, which ensures good generalization performance, but results in relatively weak discriminative power, e.g. with respect to background structures. By adding a 4<sup>th</sup> dimension for object articulations to the ISM voting space (Figure 1 center), the model is able to distinguish between object shapes and is thus more discriminant [13]. This, however, requires an association of each training example to one of the typical articulations or shapes.

**Learning Object Shapes.** Manual labelling of object shapes in the training data is both time consuming and difficult for more complex objects. We therefore automatically learn the most prominent shapes from object silhouettes. Therefore, we apply agglomerative clustering with global Chamfer distance as similarity measure. The silhouettes are extracted from video sequences with a motion-segmentation algorithm [14]. For the object category of pedestrians the silhouette is often a good indication of the current body articulation. As an example, Figure 3 shows the identified articulation clusters for side-view pedestrians generated by this method.

**4D Voting.** In this paragraph we describe the probabilistic formulation of the extended 4D voting procedure. Let  $e$  be a local descriptor computed at location  $\ell$ . Each descriptor is compared to the codebook and may be matched to several codebook entries. One can think of these matches as multiple valid interpretations  $I_i$  for the descriptor, each of which holds with the probability  $p(I_i|e)$ . Each interpretation then casts votes for different object instances  $o_n$ , locations  $\lambda_x, \lambda_y$ , scales  $\lambda_\sigma$  and shape clusters  $s$  according to its learned occurrence distribution  $P(o_n, \lambda, s|I_i, \ell)$  with  $\lambda = (\lambda_x, \lambda_y, \lambda_\sigma)$ . Thus, any single vote has the weight  $P(o_n, \lambda, s|I_i, \ell)p(I_i|e)$  and the descriptor's contribution to the hypothesis can be expressed by the following marginalization:

$$P(o_n, \lambda, s|e, \ell) = \sum_i P(o_n, \lambda, s|I_i, \ell)p(I_i|e, \ell) \quad (1)$$

$$= \sum_i P(\lambda, s|o_n, I_i, \ell)p(o_n|I_i, \ell)p(I_i|e)$$

$$P(o_n, \lambda, s) \sim \sum_k P(o_n, \lambda, s|e_k, \ell_k) \quad (2)$$

There are, however, several issues with this formulation. First, it is difficult to estimate the probability density  $P(\lambda, s|o_n, I_i, \ell)$  reliably due to the increased dimensionality, in particular from a relatively small set of data. Second and quite importantly, the shape dimension  $s$  is neither continuous nor ordered. It is therefore unclear, how the maximum search can be efficiently formulated. Applying a Mean-Shift search with a scale-adapted kernel, as in the standard ISM approach, is no longer feasible. Therefore, the following factorization is used to obtain a tractable solution:

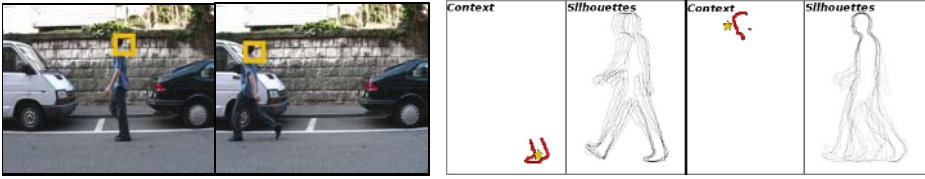
$$P(o_n, \lambda, s|e, \ell) = \sum_i P(s|\lambda, o_n, I_i, \ell)P(\lambda|o_n, I_i, \ell)p(o_n|I_i, \ell)p(I_i|e) \quad (3)$$

Please note, that all but the first term ( $P(s|\lambda, o_n, I_i, \ell)$ ) are the same as in [6]. Therefore we can use the following simple yet effective strategy to find the maxima of equation 2. By first searching the  $K$  maxima in the marginalized 3D voting space, we can not only reduce the computational complexity but also constrain our search to those areas of the probability density with enough evidence and training data. Choosing  $K$  sufficiently large, we can find all maxima with high probability. For those  $K$  maxima we then retrieve the contributing votes and use the following calculation (for simplicity of notation we use  $P(s|H) = P(s|\lambda, o_n, I_i, \ell)$ ):

$$P(s|H) = \sum_j P(s|c_j, H)p(c_j|H) = \sum_j P(s|c_j)p(c_j|H) \quad (4)$$

where  $c_j$  corresponds to the individual silhouettes present in the training data and  $s$  is a shape cluster.  $P(s|c_j)$  represents the probability that silhouette  $c_j$  is assigned to cluster  $s$ .  $P(s|c_j)$  is 1 if silhouette  $c_j$  is contained in shape cluster  $s$ .

By following the above procedure, we can obtain the 4D-maxima of  $P(o_n, \lambda, s)$ . This means in particular, that the votes corresponding to these maxima conform



**Fig. 2.** (Left) The same local feature can occur on globally dissimilar object shapes. (Right) The comparison of local contexts (red) around interest points (yellow star), influences the choice of object shapes considered in the recognition process.

with a common shape cluster. As a result, the voting scheme produces hypotheses, which have a consistent shape.

### 2.3 Cross-Articulation Learning Using Local Contexts

As will be seen in the experiments, the 4D voting procedure for individual object shapes improves the recognition performance w.r.t. the original ISM approach. In particular, the discriminative power of the learned object model is increased, since it enables to distinguish typical object articulations. While this is a desirable goal, it involves a number of side effects.

On the one hand, we reduce the statistical significance of the object hypotheses, since the number of features contributing to each hypothesis has been reduced. In essence, the votes are distributed over a range of articulation clusters. This can be easily seen from the schematic views of the original ISM-model (Fig. 1 left) and the 4D voting approach (Fig. 1 center). In the standard ISM model feature occurrences from all training instances can be combined for a final hypothesis. This is a desirable property, which we call cross-instance learning, that uses the training images effectively and allows to obtain high recognition performance with a relatively small number of training images. Even though, in the case of the 4D-ISM, codebook entries are shared and some limited cross-articulation learning is achieved, the feature occurrences and therefore the votes are basically limited to a certain shape cluster. The goal of the following is therefore to introduce an algorithm that allows for more effective cross-articulation learning and thereby increasing the generalization power of the approach without losing the gained discriminative power of the 4D-ISM.

To illustrate the underlying idea, consider the images shown in figure 2 (left). Assume that we are observing a head-feature (shown as the yellow square in the two left images). In that case, the observation of the head puts very little constraints on the particular position and shape of the legs. In terms of the 4D-ISM, this means that we should not restrict our votes to one particular articulation but rather vote for a range of different but compatible articulations.

While, in principle, an increase of the number of training instances should compensate for the limited cross-instance learning in the 4D-ISM, we, motivated



by the discussion above, propose another strategy. Our strategy re-enables cross-instance learning without the need of more training data. The principle idea is that object shapes, while being globally dissimilar, are often very similar in a more local context. So instead of considering only the global shape for the assignment of feature occurrences to articulation clusters, we propose to compare the local context of an interest point. This is illustrated in Figure 2 (right). There, we consider a local context (represented as local silhouette segment in red) extracted around an interest point (yellow star) and depict locally similar object silhouettes. As can be seen, an occurrence at the head (here the front of the head) is compatible with many different articulations. An occurrence on the foot, on the contrary, constrains the range of compatible articulations considerably.

**Learning the Shape Distribution.** In order to integrate this idea into our probabilistic voting framework, we adapt equation 4 with information about the similarity between local contexts on different object shapes.

The *Shape Distribution*  $P(s|c_j, H)$  associates a silhouette with a corresponding shape cluster depending on the location  $\ell$  of a codebook occurrence (remember  $H = (\lambda, o_n, I_i, \ell)$ ). Thus the equality  $P(s|c_j, H) = P(s|c_j)$ , as used in section 2.2, does not hold any longer. Instead we define  $P(s|c_j, H)$  in the following manner:

$$P(s|c_j, H) = \begin{cases} \frac{1}{Z} & \text{if } \exists c_i : P(s|c_i) = 1 \text{ and } \text{dist}(r_i(H, d), r_j(H, d)) < t \\ 0 & \text{else} \end{cases} \quad (5)$$

with  $Z$  a normalization factor and  $r_i(H, d)$ ,  $r_j(H, d)$  local contexts with radius  $d$  around  $\ell$  on the training instances corresponding to the silhouettes  $c_i$  and  $c_j$ .

In other words, for each codebook occurrence at an interest point, we look for similar local contexts in all other training instances. If we have found such a matching context, we adjust the probability distribution, which associates object silhouettes with shape clusters. Note, that if no matching local contexts are found, we obtain the original equality  $P(s|c_j, H) = P(s|c_j)$  and thus the 4D voting approach of section 2.2. On the other hand, if we always find matching contexts in all shape clusters, the model is equivalent to the standard ISM approach. Figure 1 (right) shows a schematic illustration of the newly proposed object model.

A major advantage of this approach is, that we can vary its behavior, by choosing different context radii. The larger the local context are, the more global the decision process becomes. Thus we can find the level of locality, which is appropriate for an object category. The choice of appropriate local context representations and distance measures is another point of consideration. We can simply use local silhouette segments as in Figure 2 to describe a context or use more sophisticated descriptors, which might include information about background structures present in the training data.



**Fig. 3.** (Left) Learned articulation clusters. (Right) Example images from our training set.

### 3 Experimental Evaluation

This section evaluates the performance of the newly proposed recognition scheme and analyzes the influence of the involved parameters. In particular we analyze the influence of both context radius and context representation.

We conduct our experiments on two challenging test sets of pedestrians. Test set *A* consists of 181 images of pedestrian side-views. Each image contains a single pedestrian. Pedestrian appearances vary considerably, and people often take up only a small portion of the image. Also the backgrounds exhibit a huge variability and contain significant amount of clutter. Test set *B* consists of 206 images, with a total of 595 pedestrians. Pedestrians in this test set frequently overlap or are partially occluded by bags or other objects.

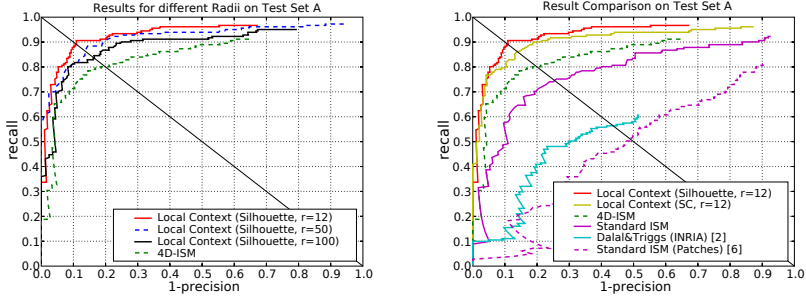
For training we use a set of 210 images from two video sequences, which are mirrored in order to have the same amount of pedestrians heading left and right. The images contain two different backgrounds and 27 subjects in various articulations with a height of approximately 200 pixels (see Figure 3 for some example images). Please note, that image backgrounds and visual appearances of the pedestrians differ considerably between training and test sets.

Images in the training set are annotated with segmentation masks. These are typically computed from the recorded video sequences with a Grimson-Stauffer background model [14]. From these segmentation masks we additionally compute the shape silhouettes, which are used for the shape clustering step during model training. The resulting articulation clusters consist of 5 articulations from a typical walking cycle and their respective mirrored articulation, which results in a total of 10 clusters (see Figure 3).

#### 3.1 Test Set *A* - Single Pedestrians

On test set *A* we show the detection performance when pedestrians are fully visible. For this case, hypotheses should be rather accurate, which makes it easier to analyze and separate the effects of the proposed improvements.

The radius of the local context determines, how local or global the method operates. We evaluate a set of 4 radii ranging from 12 pixel to the whole object size. For the context representation, we consider local silhouette segments and a modified *Shape Context* (SC) descriptor [1,9], which has been proposed for pedestrian detection in [12].



**Fig. 4.** Recognition performance on side-view pedestrians (Test Set *A*). In the left plot different context radii are evaluated. In the right plot we compare the results to previous approaches.

Figure 4 (left) shows the obtained results for various radii and the local silhouette representation. The green curve is our reference performance of the 4D-ISM, which achieves an equal error rate (EER) of 80%. Note, that the 4D-ISM is equivalent to choosing a radius of the whole object size. As can be seen, the performance improves with small context radius, reaching an EER of nearly 90% for a radius of 12 pixels.

Using SC to describe the local context, we achieve very similar results with the same ranking from local to global. The recognition rates are, however, a little lower than those obtained with local silhouette segments. This is probably due to the fact, that by using silhouette segments for comparison, we discard disturbing background structures, which are present in the SC descriptors.

Figure 4 (right) compares the obtained results to previous approaches. The conventional ISM approach [6] using image patches and the Difference-of-Gaussians detector achieves an equal error rate (EER) of 50%. This is probably due to the large appearance differences between training and test set and the difficult data, which includes heavily cluttered backgrounds. Using more appropriate feature descriptors along with the Hessian-Laplace detector [12] improves performance to an EER of 74%. The 4D-ISM produces consistent hypotheses and can thus improve the performance to 80% in EER. Extending the 4D-ISM with the proposed cross-articulation learning further increases both detection precision and recall. We obtain equal error rates of 86% and nearly 90%, depending on the choice of context representation. This is remarkable considering the difficulty of the test set. To stress this, we compare the results to the state-of-the-art detector of Dalal & Triggs [2] using the detector available on the authors' webpage. [2] achieves an EER performance of 57% on test set *A*. To be fair, it should be mentioned, that our detector was trained in this case on side-views only, whereas their system was built for multi-viewpoint detection. However, Dalal & Triggs also use an order of magnitude more training data.

In conclusion, the proposed cross-articulation learning is, while separating the influences of different articulations, able to exploit the information present in the

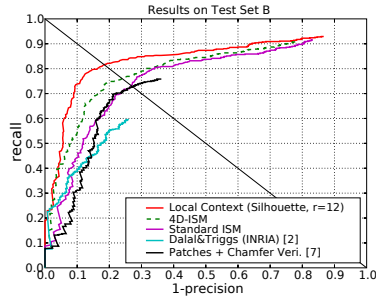


Fig. 5. Recognition performance on side-view pedestrians (Test Set B)

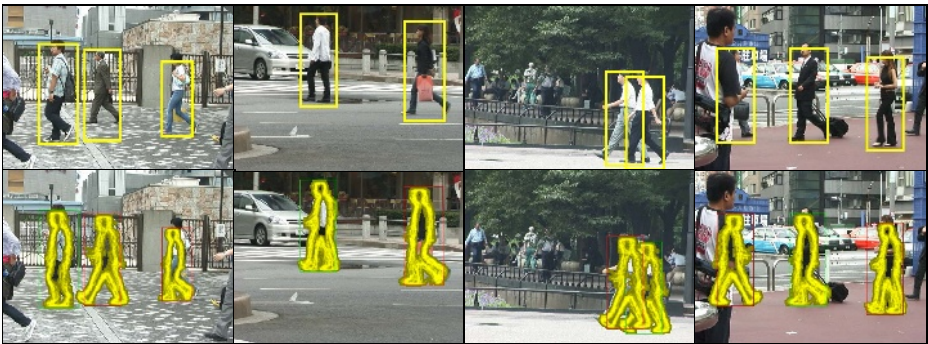


Fig. 6. Example detections at the EER (upper row) with corresponding articulation estimates (bottom row)

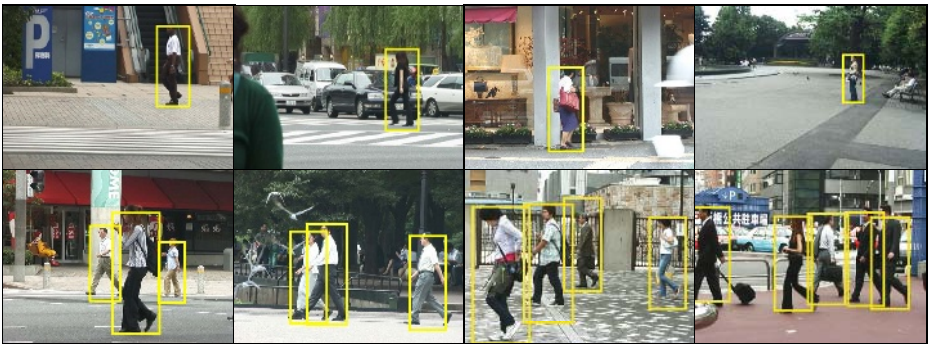


Fig. 7. Example detections on test set A (upper row) and testset B (lower row) at the EER

training data to a large extent. Thus it represents a considerable improvement over previous approaches.

### 3.2 Test Set *B* - Crowded Scenes

Test set *B* contains occluded and overlapping pedestrian. As a result, the estimation of a person’s articulation is more difficult. This section analyzes whether the proposed approach can improve detection performance in this challenging setting. Similar to section 3.1, we compare the results with previous approaches, as well as the pedestrian detector of [2].

Figure 5 depicts the respective results. The pedestrian detection system of Dalal & Triggs achieves a recall of 60% at a precision of 75%. Note that, the curve stops there, since the available binary has a fixed threshold, which cannot be changed. The approach of [7] achieves an EER of 73%. In this work hypotheses resulting from local evidences are subsequently verified globally with a set of silhouettes. A combined score is computed of the global Chamfer distance and a measure based on overlap between silhouette and segmentation.

Using the standard ISM with SC descriptors results in the same EER of 73%. By applying the extended 4D-voting for the shape clusters, the performance is mainly improved with respect to detection precision. The improvement in EER is 3%. Even on this challenging data set, cross-articulation learning significantly increases the recognition performance. The best result is again obtained for a very small radius of the context. The approach achieves an EER of 81%, which corresponds to a 5% increase.

Figure 6 shows example detections of the proposed approach with their corresponding articulation estimates on test set *B*. Both detection bounding boxes and body articulations are predicted correctly, when people are overlapping or occluded. Figure 7 displays more example results for both test set *A* and *B*.

## 4 Conclusion

In this paper we have introduced a new approach to enable cross-articulation learning for robust detection of pedestrians in difficult real-word scenes. The experiments suggest that the approach enables a sensible trade-off between generalization performance and discriminative power. In particular, the new approach makes effective use of the available training data through the proposed cross-articulation learning scheme. By altering the radius of the considered contexts, we are able change the approach to operate more locally or more globally. A thorough evaluation has shown, that the proposed approach makes object detection more robust in realistic environments and in the presence of overlapping and partially occluded objects. The new approach outperforms previously published results of state-of-art pedestrian detectors on two challenging multi-scale data sets, underlining the effectiveness of the approach.

**Acknowledgements.** This work has been funded, in part, by the EU project CoSy (IST-2002-004250) and Toyota Motor Europe.

## References

1. S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *PAMI*, 2002.
2. N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.
3. P. Felzenszwalb and D. Huttenlocher. Efficient matching of pictorial structures. In *CVPR*, 2000.
4. D. Forsyth and M. Fleck. Body plans. In *CVPR*, 1997.
5. D. Gavrilu. Multi-feature hierarchical template matching using distance transforms. In *ICPR*, volume 1, pages 439–444, 1998.
6. B. Leibe and B. Schiele. Scale invariant object categorization using a scale-adaptive mean-shift search. In *DAGM*, pages 145–153, 2004.
7. B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. In *CVPR*, 2005.
8. C. Mikolajczyk, C. Schmid, and A. Zisserman. Human detection based on a probabilistic assembly of robust part detectors. In *ECCV*, pages 69–82, 2004.
9. K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *PAMI*, 27(10):1615–1630, 2005.
10. A. Mohan, C. Papageorgiou, and T. Poggio. Example-based object detection in images by components. *PAMI*, 23(4):349–361, 2001.
11. C. Papageorgiou and T. Poggio. A trainable system for object detection. *IJCV*, 38(1):15–33, 2000.
12. E. Seemann, B. Leibe, K. Mikolajczyk, and B. Schiele. An evaluation of local shape-based features for pedestrian detection. In *BMVC*, 2005.
13. E. Seemann, B. Leibe, and B. Schiele. Multi-aspect detection of articulated objects. In *CVPR*, 2006.
14. C. Stauffer and W. Grimson. Adaptive background mixture models for realtime tracking. In *CVPR*, 1999.
15. A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing visual features for multiclass and multiview object detection. In *submitted to PAMI*, 2005.
16. P. Viola, M. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *ICCV*, pages 734–741, 2003.
17. L. Zhao and C. Thorpe. Stereo and neural network-based pedestrian detection. *IEEE Transactions on Intelligent Transportation Systems*, 1(3):148–154, 2000.

# Analysis on a Local Approach to 3D Object Recognition

Elisabetta Delponte, Elise Arnaud, Francesca Odone, and Alessandro Verri

DISI - Università degli Studi di Genova - Italy

**Abstract.** We present a method for 3D object modeling and recognition which is robust to scale and illumination changes, and to viewpoint variations. The object model is derived from the local features extracted and tracked on an image sequence of the object. The recognition phase is based on an SVM classifier. We analyse in depth all the crucial steps of the method, and report very promising results on a dataset of 11 objects, that show how the method is also tolerant to occlusions and moderate scene clutter.

## 1 Introduction

This paper proposes a method based on local keypoints that allows us to recognize 3D objects in real environments, under different viewing and scene conditions. The method falls into the view-based approaches that have been widely used in the past for object recognition, as they offer a simple but principled way to model viewpoint variation.

Our method can be summarized as follows: for each object we acquire an image sequence that describes the 2D appearance of the object from different viewpoints. We extract local keypoints from the sequence, describe them with SIFT descriptors, and track them over the sequence with an Unscented Kalman filter. For each SIFT trajectory we compute a compact representation, or *virtual feature*, that becomes the delegate for all the keypoints of the trajectory and hopefully for the same feature belonging to yet to be seen images of the same object. The collection of virtual features form a model of the object, or *vocabulary*. The actual recognition is based on learning from examples: we represent all the images of a training set with respect to the vocabulary estimating the degree of similarity between each image and the vocabulary, and train a binary SVM classifier. The multiclass nature of the problem is captured with a one-vs-all approach. We carry out our analysis on a dataset of 11 objects (see Fig. 1).

The paper is organized as follows. Section 2 reviews related work on object recognition with local approaches. Section 3 describes how the object model is built, while Section 4 discusses how to represent an image with respect to the model. Section 5 describes the training stage, and Section 6 reports the recognition experiments. Section 7 is left to a final discussion.



**Fig. 1.** The 11 objects of the dataset. From top left: *bambi*, *box*, *duck*, *biscuit*, *book1*, *book2*, *dino*, *teddy*, *pino*, *tele*, *tommy*.

## 2 Related Work

In the last few years there has been an increasing interest on object recognition systems based on local keypoints. Among the many possible approaches we focus on those combining local descriptions with learning from examples.

Statistical learning methods have been often coupled to local keypoints through the design of *ad hoc* kernels. Since local representations are variable-length and usually they do not carry internal ordering, local kernels are derived from the studies on kernels for sets. One of the first works proposed is the *matching kernel* [10], that has been proved very effective for object recognition. Unfortunately, it has been demonstrated that it is not a Mercer kernel [2]. More recently a modification of the matching kernel, called *intermediate matching kernel* has been proposed [2]. The new kernel is based on the concept of *virtual features*, which is reminiscent of the ones we will define.

An alternative approach to combining local descriptors with learning from examples is the so called *bags of keypoints* approach. The idea is inspired to the bag of words used for text categorization, and it was first proposed in [3] for visual categorization. The method can be summarized as follows: (1) extract interesting points for all images of the training set mixing keypoints from all the classes; (2) cluster all the keypoints and identify the keypoints “bags”, or equivalence classes; the collection of bags form a vocabulary of keypoints; (3) represent all images with respect to the bags with a histogram-like approach. With this approach keypoints belonging to different objects may fall in the same equivalence class, as long as they are more similar to it than to other classes.

Our method is close to bags of keypoints, but it is also somewhat related to the intermediate matching kernel. Similarly to [3] we look for keypoints equivalence classes, but since our input datum (an image sequence) is more informative than a single image, our classes will only contain different instances of the same feature. To do so, we exploit the image sequence temporal coherence. A similar idea can be found in [4], where a local spatio-temporal description is computed using SIFT and the KLT tracker; since their tracker has no prediction ability a more complex trajectory selection is used.



### 3 The Object Model

For each object of interest the first stage of our method consists of finding a model based on local keypoints taken from an image sequence of the object. First, we extract local keypoints from all the images of the training sequence and exploit temporal coherence by tracking them, obtaining a list of trajectories, or *trains of keypoints*. Second, we represent the trains in a compact way that we call a *virtual feature* and build a *vocabulary* of such compact representations. This vocabulary is the model of the object, since it is the information about the object that we will use for recognition.

#### 3.1 The Local Keypoints

For each image we locate interesting points on a difference of Gaussians pyramid. These points are centers of blob-like structures. We represent them with SIFT descriptors [6]. This descriptor contains the following information about the keypoint: (a) position  $\mathbf{p}$ , (b) scale  $\mathbf{s}$ , (c) main orientation  $\mathbf{d}$ , and (d) a vector  $\mathbf{H}$  containing local orientation histograms around the keypoint position. We will use the first three elements for SIFT tracking and the orientation histograms  $\mathbf{H}$  for computing the similarities. Scale and main orientation are also implicitly used for computing the similarities, as  $\mathbf{H}$  is built on an image patch centered at the keypoint position, and scaled and rotated according to scale and main orientation.

#### 3.2 The SIFT Tracker

The selected keypoints are tracked over time with an Unscented Kalman filter [5,11]. This method belongs to the filtering algorithms family that are well-known for their simplicity and robustness. Such an algorithm allows us to cope with temporal detection failures, and as a consequence avoids redundancies in the vocabulary.

Filtering methods consist of a dynamic system tracked by a hidden Markov process. The goal is to estimate the values of the state  $\mathbf{x}_k$  from a set of observations  $\mathbf{z}_{1:n} = \{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_n\}$ . The system is described by a dynamic equation  $p(\mathbf{x}_k|\mathbf{x}_{k-1})$  modeling the evolution of the state and a measurement model  $p(\mathbf{z}_k|\mathbf{x}_k)$  that links the observation to the state. The goal is then to estimate the *filtering distribution*  $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ , that carries the whole information on the process to be estimated. The choice of the estimation algorithm (Kalman filter, Extended Kalman filter, Particle filter, etc.) depends on the characteristics of the model (if it is linear, Gaussian, etc.).

The system we consider here – whose unknown state is  $\mathbf{x}_k = \{\mathbf{p}_k, \mathbf{s}_k, \mathbf{d}_k\}$ , where  $\mathbf{p}_k$  is the SIFT position,  $\mathbf{s}_k$  its scale and  $\mathbf{d}_k$  its main orientation – is composed of the following dynamic and measurements models.

The **dynamic equation** describes the evolution in time of the keypoint. A constant model is associated to  $\mathbf{s}_k$  and  $\mathbf{d}_k$ :

$$\begin{pmatrix} \mathbf{s}_k \\ \mathbf{d}_k \end{pmatrix} = \begin{pmatrix} \mathbf{s}_{k-1} \\ \mathbf{d}_{k-1} \end{pmatrix} + \gamma_k, \quad (1)$$

where  $\gamma_k$  is a zero-mean Gaussian white noise of covariance matrix  $\Gamma_k$  (set *a priori*). As for  $\mathbf{p}_k$ , its dynamic model has to describe the motion of the keypoint along the image sequence. Since no *a priori* information is available, and in order to be reactive to any change of speed and direction, we define the state equation as [1]:

$$\mathbf{p}_k = \mathbf{p}_{k-1} + \mathbf{u}_k(\mathbf{p}_{k-1}) + \psi_k, \quad (2)$$

where  $\psi_k$  is assumed to be zero-mean Gaussian white noise of covariance  $\Psi_k$  (set *a priori*). The variable  $\mathbf{u}_k(\mathbf{s})$  denotes the motion vector associated to a pixel  $\mathbf{s}$ . It is estimated with a robust parametric technique [7] that computes a 2D parametric model representing the dominant image motion within a considered support  $\mathcal{R}$ . For the computation of  $\mathbf{u}_k(\mathbf{p}_{k-1})$  between images  $\mathbf{I}_{k-1}$  and  $\mathbf{I}_k$ ,  $\mathcal{R}$  is chosen as a small region around  $\mathbf{p}_{k-1}$ , introducing a non linearity in the system.

Given a search window, the **measurement**  $\mathbf{z}_k$  is the keypoint that is nearest to the prediction. The measurement and the state are defined in the same space, then the following linear observation model can be set:

$$\mathbf{z}_k = \mathbf{x}_k + \mathbf{v}_k, \quad (3)$$

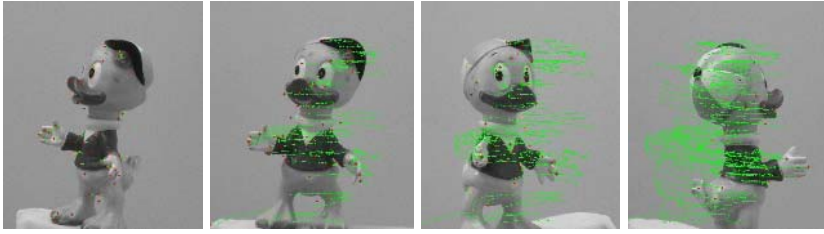
where  $\mathbf{v}_k$  is a zero-mean Gaussian white noise of covariance  $R_k$  (set *a priori*). If no keypoint is detected in the search window,  $R_k$  is set to  $\infty \times Id$  ( $Id$  is the identity matrix) so that the current estimation only relies on the dynamic equation.

As the dynamic equation is non linear because of Eq. (2), the Kalman filter is not appropriate. Recently Particle filters have been extensively used to deal with the non linearity of a system. These methods are very interesting because they enable an accurate approximation of the distribution of interest even if it is highly multimodal. However, their interest can decrease if the system under consideration is weakly non linear as the one we propose here. In this case, the use of an algorithm that assumes a Gaussian approximation of the filtering density can be both sufficient and efficient. We choose the Unscented Kalman filter that describes the Gaussian approximation of the posterior density by carefully selected weighted sample points. These points capture the mean and covariance of the approximation accurately to the 3rd order and are propagated through the non linear system. To implement our SIFT tracker we apply the Unscented Kalman Filter to Eq. (1, 2, 3).

### 3.3 The Vocabulary

All keypoints linked by a tracking trajectory, or train, belong to the same equivalence class. A *virtual feature*  $\mathcal{V}_i$  is the average of all local orientation histograms  $\mathbf{H}_k$ , with  $k$  running through the train. We use the virtual feature as a delegate for the train. Average values are good representatives of the original keypoints as the tracking procedure is robust and leads to a class of keypoints with a small variance. Being an average, some histogram peculiarities are smoothed or suppressed, but empirical evidence shows that the information that it carries is enough to describe the object.

The set of virtual features form a vocabulary of keypoints for the object:  $\mathcal{V} = \{\mathcal{V}_1, \dots, \mathcal{V}_N\}$ . *The vocabulary  $\mathcal{V}$  is the model for the object.*



**Fig. 2.** SIFT tracking on the image sequence of object *duck*: SIFT point trajectories are displayed at different steps of the image sequence

## 4 Image Representation

Once the model for the object has been computed, each image is represented with respect to the model. This representation carries information on how much related the image is related to the object. It is based on first extracting local keypoints from the image, then comparing this list of keypoints with the object vocabulary.

### 4.1 The Choice of the Similarity Measure

A crucial point is to decide how to compare the local orientation histogram of a keypoint with the average orientation histogram of a virtual feature. Then, a comparison criterion for histograms seems to be appropriate. We consider (1) Euclidean distance  $D$ , (2) Chi-square distance  $\chi^2$ , (3) Kullback-Leibler divergence  $\mathcal{K}$ , (4) Histogram intersection  $\cap$  [9].

Since 1-3 are distance measures we will use the exponent version:  $D_{exp} = \exp(-D)$ ,  $\chi_{exp}^2 = \exp(-\chi^2)$ ,  $\mathcal{K}_{exp} = \exp(-\mathcal{K})$ . Also, since the keypoint descriptions may not be normalized, instead than measure 4 we will use

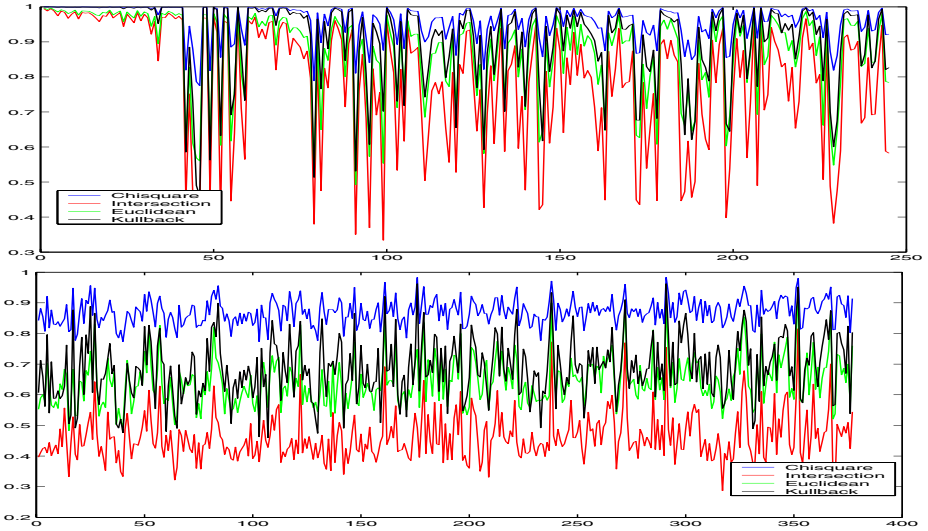
$$\cap_{norm}(H, H') = \frac{\cap(H, H')}{\cup(H, H')} = \frac{\sum_{i=1}^n (\min(H_i, H'_i))}{\sum_{i=1}^n (\max(H_i, H'_i))}.$$

If the histograms are normalized, this similarity measure is equivalent to histogram intersection.

Let us reason on what we would ask to a similarity measure: high scores on similar keypoints, low scores on different keypoints. Figure 4 shows the results of comparing two similar images (Figure 3, left and center) and two very different images (Figure 3, center and right), with the four similarity measures. The plots are obtained as follows: for each keypoint of the first image we compute the highest match value with respect to keypoints of the other image. The results show that Chi-square returns uniformly high scores in both cases. The best compromise between intraclass and interclass keypoints is obtained with normalized histogram intersection  $\cap_{norm}$ , which will be used in the rest of the experiments.



**Fig. 3.** Example images used for the comparative analysis of similarity measures



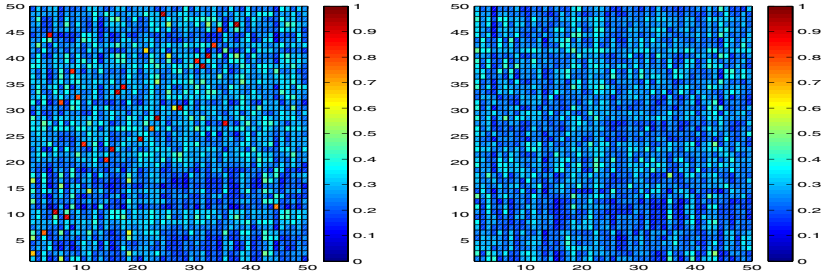
**Fig. 4.** Match values obtained comparing 2 images with the 4 similarity measures. Top: results from two similar images (Fig. 3 left and center). Bottom: results from two different images (Fig. 3 center and right). On the x axis are the indices of keypoints in the first image, on the y axis the corresponding match values with the most similar keypoints of the second image.

### 4.2 Building the Representation

An image  $F_i$ , after we extract local interest points, can be seen as a collection of keypoints  $F_i = \{\mathcal{F}_1^i, \dots, \mathcal{F}_M^i\}$ , where  $M$  will vary. The vocabulary helps us to avoid the problem of variable length representations: each image  $F_i$  is represented with a vector  $R_i$  of length  $N$ .

Each entry  $k$  of  $R_i$  carries the contribution of the keypoint  $\mathcal{F}_j^l$  most similar to  $\mathcal{V}_k$ , if there is one. Possible choices on how to build  $R_i$  include:

1. **Binary entries**, with  $R_i^k = 1$  if there exist a keypoint  $\mathcal{F}_j^l$  closer to  $\mathcal{V}_k$  than a threshold.
2. **Real value entries** describing the degree of similarity between  $\mathcal{V}_k$  and the most similar keypoint  $\mathcal{F}_j^l$ .
3. **SIFT entries**, with  $R_i^k = \mathcal{F}_j^l$ , where  $\mathcal{F}_j^l$  is the most similar keypoint to  $\mathcal{V}_k$ .



**Fig. 5.** Similarity matrices obtained comparing the vocabulary of object  $\mathcal{O}$  with one image of object  $\mathcal{O}$  (on the left) and one image of another object (on the right). On the rows: the image keypoints, on the columns: the virtual features (see text).

Our image representations will be based on choice 2, as it is the best compromise between effectiveness and simplicity. It is worth mentioning that choice 3 corresponds to an explicit mapping of the intermediate matching kernel [2].

We compute the similarity values between all keypoints of image  $F_i$  and all virtual features of the vocabulary  $\mathcal{V}_k$ . An explicit computation would lead to a similarity matrix as the ones shown in Figure 5. The final description is obtained by taking the maximum values column-wise. While finding the association between  $\mathcal{V}_k$  and  $\mathcal{F}_i^j$ , keypoint that appear similar to more than one virtual feature are penalized. Figure 5 considers a vocabulary for object  $\mathcal{O}$  and includes the comparison with one image of object  $\mathcal{O}$  (on the left) and one image of another object (on the right). On the left matrix are clearly visible the high match values corresponding to the most similar keypoint.

## 5 Object Representation

We acquired a dataset of 11 different objects (Figure 1) that include examples of similar objects (5 plastic toys, 2 books), but at the same time are variable enough to represent a possible selection of things of a real indoor environment.

Each object is represented by an image sequence of about 200 frames acquired by placing the object on a turntable. We use these sequences both for building the vocabulary and as positive examples for training the recognition system. The training set is acquired in a neutral but real environment. No segmentation or background subtraction is applied.

For each object we acquired six different test sets: (1) similar conditions to the training, (2) moderated illumination changes, (3) different scale, (4) allowing for severe occlusions of the object (5) placing the object against a plain, but different background, (6) placing the object against a complex and highly textured background (see Figure 6). We also acquired background images, and images of other objects to be used as negative examples. For each object we use about 200 positive training examples, 300 negative training examples. Each object has about 18 000 images of test examples. For each object we build the vocabulary



**Fig. 6.** The different conditions under which the test data have been acquired (see text)

and then represent the training data with respect to it. We then train a binary SVM classifier with a histogram intersection kernel [8], as it was proved effective on a number of applications and does not depend on any parameter. We deal with the multiclass nature of the problem with a *one against all* approach.

## 6 Experiments on Object Recognition

The recognition rates obtained over test sets (1-4) are summarized in Table 1. The column **simple** refers to the results obtained on test sets (1) and (2), the column **scale** refers to test set (3), while the column **occlusions** refers to test set (4). The very good results confirm how SIFT keypoints combined with a robust feature tracking produce a model which is robust to illumination and scale changes, and to occlusions. The description proposed captures the peculiarity of objects, and allows us to recognize them correctly even if the possible classes contain many similar objects. The drop obtained for object “book2” is due to the fact that in many test images the object was entirely hidden. In the case of more complex backgrounds, instead, it is worth showing the confusion matrices (Tables 2 and 3). They show how, if the amount of clutter is small, the recognition rates are still very satisfactory. In the case of very complex and textured backgrounds the performance drops because of the high number of keypoints detected (of which only a small number belong to the object).

**Table 1.** Hit percentages of the 11 classifiers against test sets (1-4)

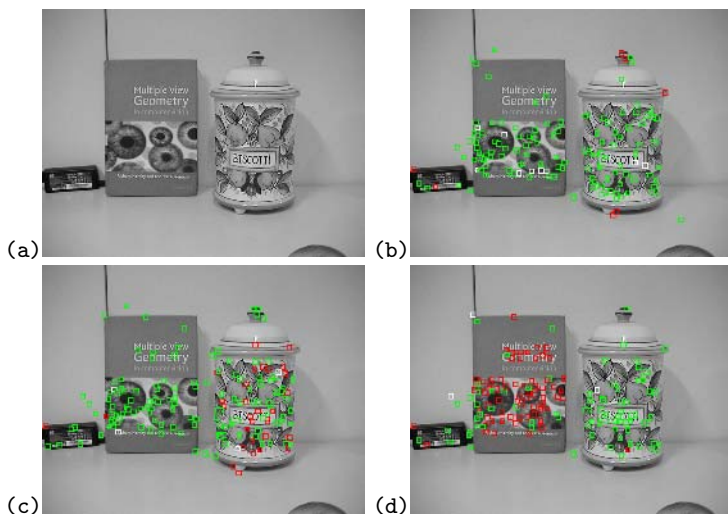
<b>objects</b>	<b>simple</b>	<b>scale</b>	<b>occlusions</b>
bambi	99.50	92.34	100.00
box	100.00	100.00	100.00
duck	100.00	98.90	100.00
biscuit	100.00	100.00	100.00
book1	100.00	100.00	100.00
book2	100.00	95.53	77.78
dino	100.00	100.00	100.00
teddy	100.00	98.86	100.00
pino	100.00	99.59	92.96
tele	100.00	100.00	100.00
tommy	100.00	100.00	100.00

**Table 2.** Confusion matrix for test set (5), with moderate quantities of clutter on the background

	bambi	box	duck	biscuit	book1	book2	dino	teddy	pino	tele	tommy
bambi	71.46	0.00	13.69	0.00	0.00	0.00	0.00	3.48	2.55	0.23	8.58
box	0.34	98.65	1.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
duck	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
biscuit	0.00	0.00	0.22	99.78	0.00	0.00	0.00	0.00	0.00	0.00	0.00
book1	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00
book2	5.22	0.00	0.37	0.00	0.00	91.04	0.00	1.49	0.00	1.49	0.37
dino	9.48	0.00	13.73	0.00	0.00	0.00	58.17	0.33	0.00	0.00	18.30
teddy	0.00	0.00	3.13	0.00	0.00	0.00	0.00	96.87	0.00	0.00	0.00
pino	15.66	0.00	15.93	0.00	0.00	0.00	7.42	1.92	41.48	0.00	17.58
tele	0.93	0.93	6.48	0.00	0.00	0.00	0.00	0.93	0.00	90.28	0.46
tommy	4.86	0.00	2.86	0.00	0.00	0.00	4.29	0.57	2.29	0.00	85.14

**Table 3.** Confusion matrix for test set (6), with a very complex background

	bambi	box	duck	biscuit	book1	book2	dino	teddy	pino	tele	tommy
bambi	2.11	0.00	5.15	19.67	2.11	11.01	0.00	11.94	0.00	8.90	39.11
box	0.00	85.81	0.65	0.65	0.00	8.06	0.65	0.00	0.00	0.65	3.55
duck	0.53	0.00	40.74	9.52	1.06	0.53	0.53	4.23	0.53	4.23	38.10
biscuit	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
book1	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00
book2	0.37	0.00	0.74	0.00	0.37	96.68	0.00	0.37	0.00	0.74	0.74
dino	1.08	0.00	0.65	16.85	33.69	3.46	2.38	11.45	1.08	2.81	26.57
teddy	1.24	0.00	3.96	0.25	1.73	4.70	0.50	36.14	7.43	14.60	29.46
pino	0.00	0.63	8.15	25.08	13.48	7.52	0.00	4.70	0.63	10.34	29.47
tele	0.00	0.47	0.47	0.00	0.94	12.21	0.00	0.00	0.00	81.22	4.69
tommy	2.07	0.00	0.00	1.38	7.24	6.55	0.00	33.79	1.72	6.55	40.69


**Fig. 7.** A test image (a), and the SIFT points matched with vocabularies of different objects: (b) duck, (c) biscuits, (d) book1 (high similarity score are red)

## 7 Discussion

We proposed a method for 3D object recognition which is robust to scale, illumination changes and viewpoint variation. The results we presented show that

the method is also tolerant to occlusions and moderate clutter. We are currently dealing with the presence of multiple objects. Preliminary results indicate that the local approach coupled with analysis on image sub-regions allows us to focus on one object at a time and achieve good recognition results. Figure 7 shows a test image with two objects and the keypoints that match 3 different object vocabularies. The high scores (in red) are positioned on the correct object. We also considered embedding additional information in the keypoint description, such as color information, but the modest increase in the performance does not justify the choice.

**Acknowledgements.** This research is partially supported by the FIRB project RBIN04PARL *Learning Theory and Applications*. The authors thank Francesco Isgro for proofreading.

## References

1. E. Arnaud, E. Mémin, and B. Cernuschi-Frías. Conditional filters for image sequence based tracking - application to point tracking. *IEEE Tr. on Im. Proc.*, 1(14), 2005.
2. S. Boughorbel, J. P. Tarel, and N. Boujemaa. The intermediate matching kernel for image local features. In *International Joint Conference on Neural Networks*, pages 889–894, Montreal, Canada, 2005.
3. G. Csurka, C. Dance, L. Fan, J. Willamowsky, and C. Bray. Visual categorization with bags of keypoints. In *Int. Work. on Stat. Learn. in CV, ECCV*, 2004.
4. M. Grabner and H. Bischof. Object recognition based on local feature trajectories. In *I Cognitive Vision Work.*, 2005.
5. S. Julier and J. Uhlmann. A new extension of the kalman filter to non linear systems. In *Int. Symp. Aerospace/Defense Sens., Sim. and Cont.*, 1997.
6. D. Lowe. Distinctive image features from scale invariant keypoints. *IJCV*, 60(2):91–110, 2004.
7. J.-M. Odobez and P. Bouthemy. Robust multiresolution estimation of parametric motion models. *Journal of Vis. Comm. and Image Rep.*, 6(4), 1995.
8. F. Odone, A. Barla, and A. Verri. Building kernels from binary strings for image matching. *IEEE Trans. on Image Processing*, 14(2):169–180, 2005.
9. M. J. Swain and D. H. Ballard. Color indexing. *IJCV*, 7(1):11–32, 1991.
10. C. Wallraven, B. Caputo, and A. Graf. Recognition with local features: the kernel recipe. In *ICCV*, page 257ff, 2003.
11. E.A. Wan and R. van der Merwe. The Unscented Kalman filter for nonlinear estimation. In *IEEE Symp. on Adapt. Sys. for Sig. Proc., Communication and Control*, 2000.



# Phase Based 3D Texture Features

Janis Fehr and Hans Burkhardt

Albert-Ludwigs-Universität Freiburg, Institut für Informatik, Lehrstuhl für  
Mustererkennung und Bildverarbeitung, Georges-Koehler-Allee Geb. 052,  
79110 Freiburg, Deutschland  
fehr@informatik.uni-freiburg.de  
<http://lmb.informatik.uni-freiburg.de/>

**Abstract.** In this paper, we present a novel method for the voxel-wise extraction of rotation and gray-scale invariant features. These features are used for simultaneous segmentation and classification of anisotropic textured objects in 3D volume data. The proposed new class of phase based voxel-wise features achieves two major properties which can not be achieved by the previously known Haar-Integral based gray-scale features [1]: invariance towards non-linear gray-scale changes and a easy to handle data driven feature selection. In addition, the phase based features are specialized to encode 3D textures, while texture and shape information interfere in the Haar-Integral approach. Analog to the Haar-Integral features, the phase based approach uses convolution methods in the spherical harmonic domain in order to achieve a fast feature extraction.

The proposed features were evaluated and compared to existing methods on a database of volumetric data sets containing cell nuclei recorded in tissue by use of a 3D laser scanning microscope.

## 1 Introduction

Segmentation and classification of anisotropic objects in 3D volume data, especially of biological structures in 3D laser scanning microscope (LSM) images, has recently become a fast rising topic. Life sciences take more and more advantage of 3D imaging techniques like LSM, combined with fluorescent antibody markers or auto-fluorescent probes. For a broad band of research topics from cellular anatomy to gene expression experiments, 3D volumetric imaging methods are used. Microscopes of the latest generation allow very fast, high resolution, multi-channel recordings which produce high amounts of data. At this stage, there is a rising demand for (semi)automatic image analysis methods which on one hand would allow high-throughput experiments, and on the other hand, provide a tool for quantitative data analysis. Most of the demanded automatic analysis tasks include the "basic" but difficult operations of segmentation, classification, or landmark detection for registration of textured objects in 3D volume data. All these operations have in common, that a rotational invariant representation of the 3D data is needed.

This paper is structured as follows: first we give a brief overview of related work, especially the Haar-Integral based gray-scale features are revised. Then

we motivate the phase based approach. In section 2 the proposed features are discussed in detail. Section 3 introduces the data driven selection of phase based features. Experiments are presented in section 4.

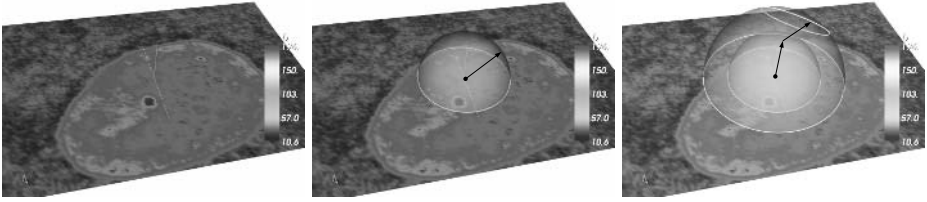
## 1.1 Related Work

To our best knowledge, there are not many publications available regarding the segmentation or classification of anisotropic 3D objects in LSM volume data. [2] presented a method for automatic segmentation of cell nuclei in dilution using region-growing (watershed) techniques. However, this method fails for recordings of nuclei in dense tissue probes as we present in our experiments. In general, we found that standard edge based, contour based, or region-growing segmentation methods deliver poor results on tissue probes. This is due the fact, that difficult segmentation tasks require a great amount of a-priori knowledge, which can not be encoded in these methods. Model driven segmentation methods, such as snakes or level-sets, are capable to encode a-priori knowledge and therefore might work well on single examples of this data. But, since there is a high variability in biological structures like cell nuclei and the demand is going towards very general and easy adaptable methods, the use of such models is complex and not flexible enough for many biological tasks.

**Learning Segmantion by Example:** An other way of incorporating a-priori knowledge into a segmentation task was presented in [3]. Here segmentation and classification is performed voxel-wise and in a single step using training examples given by a human expert. In an iterative process, the expert labels some voxels of different object classes and background. Then voxel-wise invariant features are extracted and each voxel is classified based on the given training samples. Neighboring voxels of the same label are then grouped to objects. The resulting segmentation/classification is adjusted by the expert until the model reaches a stable state. Afterwards new datasets can be segmented using this model. Models can be adopted to new cell types or even totally new data by retraining the model with additional training samples. We use this method combined with a Support Vector Machine (SVM) [4][5] classifier and our proposed phase based features for the experiments in section 4.

**Voxel-Wise Haar-Integration Features:** [1] and [6] introduced voxel-wise rotation invariant gray-scale features for combination with the previously described simultaneous segmentation and classification algorithm. In general, (rotation) invariance can be achieved via integration over the transformation (rotation) group:

$$T[f](\mathbf{X}) := \int_G f(g\mathbf{X})dg \quad \begin{array}{l} G : \text{transformation group (rotation)} \\ g : \text{element of the transformation group} \\ f : \text{non-linear mapping} \\ \mathbf{X} : n\text{-dim, multi channel data} \\ g\mathbf{X} : \text{transformed } n\text{-dim data} \end{array} \quad (1)$$



**Fig. 1.** Schematic calculation of Haar-Integral features. Left: slice through cell nuclei in original data. Center: calculation of "2-Point" invariants via Haar-Integration along the surface of a sphere. Right: "3-Point" features incorporate the relationship of gray-values at three points (center and two on concentric spheres). In this case many degrees of freedom have to be covered in order to achieve rotational invariance.

[1] and [6] formulated this approach for the special case of 3D "2-point" and "3-point" gray-scale invariants of the form (here given for "3-point"):

$$T[f](\mathbf{X}) = f_a(\mathbf{X}(\mathbf{0})) \cdot f_b(\mathbf{X}(\mathbf{q}_1)) \cdot f_c(\mathbf{X}(\mathbf{q}_2))$$

$f_a, f_b, f_c$  : arbitrary gray-scale mapping  
 $\mathbf{q}_i$  : radius

and showed a fast way of voxel-wise calculation via convolution in the spherical harmonic domain. Fig. (1) illustrates the calculation of gray-scale Haar-Integration features.

The results which can be achieved with this method are very reasonable (as shown in [1]), but the Haar-Integration approach also has some drawbacks: first, the features are not invariant towards gray-scale shifts, which appear in recordings moving deeper into the specimen. Due to the integral nature, the mean gray-value tends to dominate the value of the invariants and only a gray-scale robustness can be achieved via elaborate normalization techniques. Second, the Haar-Features have many degrees of freedom ( $\mathbf{q}_1, \mathbf{q}_2, f_a, f_b, f_c$ ), which makes an extensive feature selection necessary. And last but not least, the integration step makes it almost impossible to conduct an inverse inference from discriminating features to the original structure, which would be very useful for a deeper understanding and further improvements of the method.

## 2 Phase Based 3D Texture Features

In order to overcome the drawbacks of the Haar-Features while utilizing its strengths, we propose a new phase based approach towards voxel-wise rotation and gray-scale invariant features. As for the Haar-Features, we encode the spherical neighborhood of a voxel to an invariant feature vector. Since this feature calculation is conducted in the spherical harmonic domain, we first give a brief introduction to the harmonic methods used for our approach.

### 2.1 Spherical Harmonics

To represent the neighborhood of some point in a 3D Euclidean space as a function  $f$  on the surface of a sphere (parameterized over the two angles  $\theta$  and  $\phi$ ), the original 3D signal can be expanded in terms of spherical harmonics [7]. These provide an orthogonal basis for such functions analog to the Fourier transform in Euclidean space. This way, every spherical function can be represented by the sum of its harmonics:

$$f(\theta, \phi) = \sum_{l=0}^{\infty} \sum_{m=0}^l a_{lm} Y_l^m(\theta, \phi) \tag{2}$$

where  $l$  denotes the band of expansion,  $m$  the number of components for the  $l$ -th band and  $a_{lm}$  the harmonic coefficient. The harmonic base functions  $Y_l^m(\theta, \phi)$  are calculated as follows:

$$Y_l^m(\theta, \phi) = \sqrt{\frac{2l+1}{4\pi} \frac{(l-m)!}{(l+m)!}} \cdot P_l^m(\cos \theta) e^{im\phi} \tag{3}$$

with the associated Legendre polynomial  $P_l^m$ .

Note that in this formulation we take advantage of the symmetry in the harmonic representation, neglecting the negative coefficients. For practical reasons we also split the base components into their real and imaginary parts following the notation  $Y_l^{mr}$  and  $Y_l^{mc}$  respectively. Fig. (2) shows the first few spherical harmonics.

The transformation  $\widehat{D(l, m)}$  of the original volumetric data  $D$  into the harmonic domain is easily computed via fast convolution:

$$\widehat{D(l, m)} = Y_l^m(\theta, \phi) * D \tag{4}$$

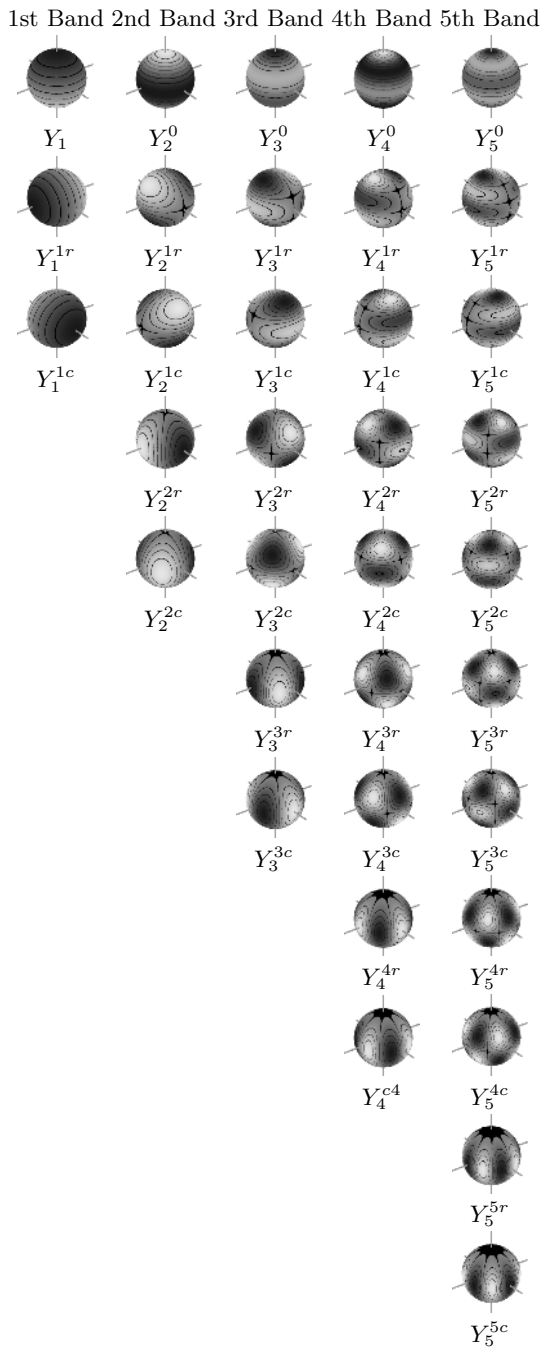
where  $*$  denotes a convolution in Euclidean space and  $Y_l^m(\theta, \phi)$  a spherical harmonic base component.

### 2.2 Feature Calculation

In order to obtain local features  $\widehat{D_r(l, m)}$  which encode the spherical neighborhood of each voxel at different consecutive radii  $r$ , we restrict the harmonic expansion to the surface of spheres  $S_r$  smoothed by a Gaussian filter  $\mathcal{G}$ .

$$\widehat{D_r(l, m)} = (Y_l^m(\theta, \phi) \cdot (S_r * \mathcal{G})) * D \tag{5}$$

Once the original volume data is transformed to the harmonic domain, there are different ways of calculating a rotational invariant representation. The simplest and well known approach is to take the band-wise absolute value of the harmonic coefficients, also known as harmonic descriptors, which for example have been used in [8] for 3D object recognition. However, this method yields a major drawback: by taking the absolute values one totally neglects the the relations between



**Fig. 2.** Spherical harmonics on a sphere surface from 1st to 5th band

the bands of the harmonic representation. This leads to ambiguous features with decreased separability.

For the Haar-Integration features the rotation invariance is achieved by integration over all possible rotations (in terms of convolutions in the harmonic domain). Here the band relations are implicitly conserved, but as mentioned before, the integration inhibits a gray-scale invariance. As the name "phase-based" features suggests, our new method uses only the relation of the harmonic bands as feature representation. This approach is motivated by results known from Fourier transform, which showed that the characteristic information is dominant in the phase of a signal's spectrum rather than in the pure magnitude of it's coefficients. Following this strategy has the nice side-effect that the overall gray-value intensity is only encoded in the amplitude, making a phase-only method directly gray-scale invariant.

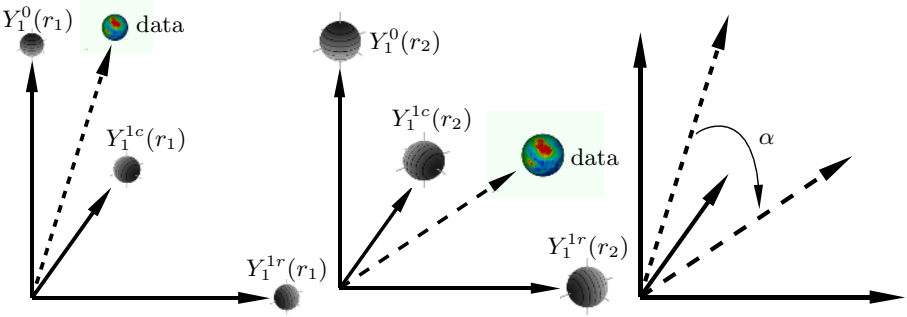
**Phase:** In this paper, the phase of a local harmonic expansion in band  $l$  a radius  $r$  is denoted by the orientation of the vector  $\mathbf{P}_{l,r}$ , containing the  $l$  harmonic coefficient components of the band-wise local expansion (Equ. 6). Since the coefficients are changing when the underlying data is rotated, the phase itself is not a rotational invariant feature.

$$\mathbf{P}_{l,r}(m) := D_r(\widehat{l}, m) \quad (6)$$

Due to the orthogonality of the harmonic base, the rotation of a spherical neighborhood can be calculated in the harmonic domain via a band-wise matrix multiplication of a symmetric and orthogonal rotation matrix  $\mathbf{R}_l$  of size  $l \times l$  with the harmonic coefficients in the  $l$ -th band. Unfortunately it turns out, that the actual calculation of this rotation matrix is getting more and more complicated and time consuming for higher bands. This would make it very expensive to achieve rotation invariance of the phase via pre-alignment.

But, there is another way to realize rotational invariant phase-only features: since we are interested in encoding the neighborhood at many consecutive radii, we can take advantage of this additional information and construct a phase-only rotational invariant feature based on the band-wise relations of phases between the different concentric harmonic series.

Fig. (3) illustrates the basic idea: the relation (angle) between phases of harmonic expansions at different radii, but in the same harmonic band, are invariant towards rotation around the center of the expansion. Intuitively phases in the same harmonic band undergo the same changes under rotation of the underlying data, keeping the angle between the phases of different radii constant. We encode this angle in terms of the dot product of band-wise spherical harmonic expansions. The resulting phase-only features can be interpreted as a description of the change in the 3D data texture, when moving from one spherical neighborhood to the next concentric neighborhood. We use this texture encoding property in the next section to find discriminatory texture elements for classes of 3D anisotropic volumetric objects. The formalization of the band-wise phase based feature vector  $T[f_l]$  calculation is given as the dot product between two band-wise expansions at radii  $r_1$  and  $r_2$ :



**Fig. 3.** Schematic example of the phase based feature calculation. Left: representation of the original data as combination of the 3D base functions of an expansion in the 1st band at radius  $r_1$ . Center: representation at radius  $r_2$ . Right: the feature is encoding the 1st band phase angle  $\alpha$  between the two concentric harmonic expansions.

$$T[f_l] := \langle P_{l_{r_1}}, P_{l_{r_2}} \rangle \tag{7}$$

**Proof** of rotational invariance is rather straight forward basic linear algebra: Since the phases of both radii are in the same band, a rotation of the underlying data can now be expressed in terms of matrix multiplications with the same orthogonal rotation matrix  $\mathbf{R}_l$ :

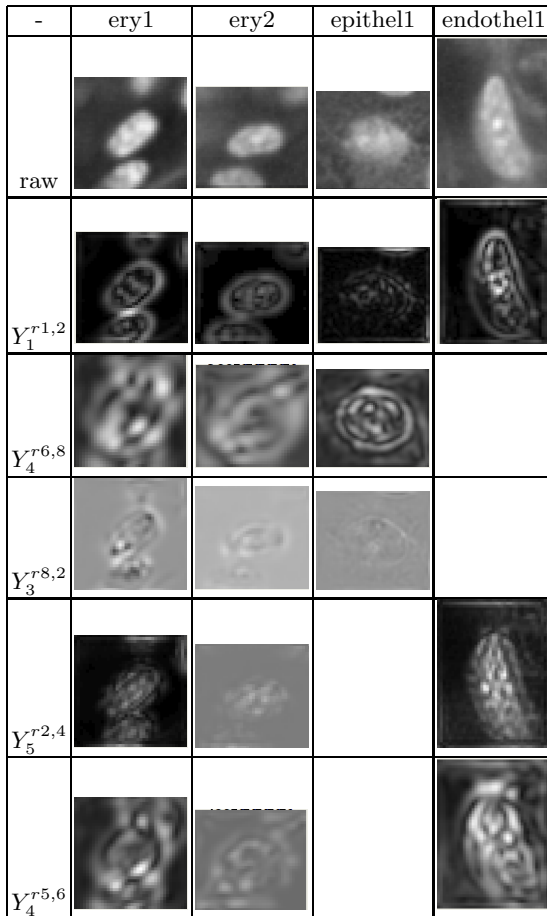
$$\begin{aligned} T'[f_l] &= \langle \mathbf{R}_l P_{l_{r_1}}, \mathbf{R}_l P_{l_{r_2}} \rangle \\ &= (\mathbf{R}_l P_{l_{r_1}})^T (\mathbf{R}_l P_{l_{r_2}}) && \text{rewrite as matrix multiplication} \\ &= (P_{l_{r_1}})^T (\mathbf{R}_l^T \mathbf{R}_l) P_{l_{r_2}} && \text{resolve transposition} \\ &= (P_{l_{r_1}})^T (\mathbf{R}_l^T \mathbf{R}_l) (P_{l_{r_2}}) && \text{comutativity} \\ &= (P_{l_{r_1}})^T \underbrace{(\mathbf{R}_l^T \mathbf{R}_l)}_{=I} (P_{l_{r_2}}) && \text{use orthogonality of } \mathbf{R}_l \\ &= (P_{l_{r_1}}^T P_{l_{r_2}}) \\ &= \langle P_{l_{r_1}}, P_{l_{r_2}} \rangle \\ &= T[f_l] \end{aligned} \tag{8}$$

Since the rotational invariance is achieved band wise, the approximation of the original data via harmonic expansion can be cut off at an arbitrary band, encoding just the level of detail needed for the application.

**Computational Complexity:** Compared to the Haar-Features the calculation of a single phase based feature has about the same computational complexity, both include expansions of two different neighborhoods in harmonics and their dot-product. However, a significant speedup can be achieved by the usage of phase based features. Our experiments showed, that due to the higher discrimination capability, the number of needed features is significant lower compared to the usage of Haar-Features.

### 3 Data Driven Features

As Fig. (4) shows an example of how different 3D textures (of nuclei types) lead to distinguishable representations in feature space. Fine, high frequency textures have an large impact on features in higher bands and at small radii while a more raw textured object is predominantly represented in the lower bands and larger radii. In order to determine the most distinguishable features, we apply a strong gauss filter to the features. This way the very local but strong texture responses are distributed to the local neighborhood. Then we apply the maximum marginal diversity algorithm [9] to calculate to most separating features. Since the phase based features are independent for every band, the data representation can be reduced to arbitrary band and radii combinations without a full transformation to the harmonic domain.



**Fig. 4.** Distinguishing feature response of different cell nuclei.  $Y_1^{r1,2}$  denotes a feature in band 1 encoding the phase change from radius 1 to 2.



## 4 Experiments

To verify our new method, we conducted some first experiments on a database of 3D laser-scanning microscope (LSM) recordings from cell nuclei in tissue.

**Data:** The database consists of 236 nuclei samples divided in 5 different classes (erythrocyte, endothelia cells, pericyte, fibroblast and macrophage). The samples were recored from tissue probes of the chicken chorioallantoic membrane which were treated as described in [10]. Human experts manually segmented and labeled the sample nuclei recordings as ground truth.

**Methods:** We extracted 16 features involving 8 different radii and expansions up to the 6th band. All features were selected from a larger number of initial features by the data driven selection method. Two reference models were trained on two small disjunct subsets of the database, containing samples from two different recording depths. The different depths cause shifts in the gray values of the recordings, as described in section 1. The remaining samples were also split into two sets, according to the recording depths, and were then classified with both models.

**Results:** We compared non gray-scale invariant 3-point Haar-Integral (gsi) features (as described in [3]) and the new, gray-scale invariant, phase based features. While the gsi features performed very well for constant gray-scales (94.53 %), the classification rate dropped to poor 46.2% for the subsets from a different recording depth. The phase based features on the other hand performed slightly worse on constant gray-scales (91,58%), but delivered stable results for varying gray values: 90.1%.



**Fig. 5.** Slice of a sample 3D database entry (erythrocyte). Left: YoPro stained channel. Center: SNAAlexa stained channel. Right: ground truth segmentation and label.

## 5 Conclusion and Outlook

In this paper we presented a novel approach of calculating rotational and gray-scale invariant 3D texture features based on the phase information of a spherical harmonic expansion of the original data. Our first experiments showed promising results and pointed out the strengths of these new features, especially concerning

gray-scale changes in 3D textures as well as the possibility to construct data driven features in comparison to Haar-Integration features.

For future work, we will continue to focus on methods for data driven features. Seeking for ways of learning the most discriminative features for larger local areas, moving towards a 3D patch based approach.

## References

1. Ronneberger, O., Fehr, J., Burkhardt, H.: Voxel-wise gray scale invariants for simultaneous segmentation and classification. In Proceedings of the 27th DAGM Symposium, in number 3663 LNCS, Springer, Vienna, Austria, 30.8 - 2.9. 2005. (2005)
2. Wählby, C., e.a.: Compining intensity, edge, and shape information for 2d and 3d segmentation of cell nuclei in tissue sections. (Journal of Microscopy, July 2004, vol. 215, no. 1, pp. 67-76(10))
3. Fehr, J., Ronneberger, O., Kurz, H., Burkhardt, H.: Self-learning segmentation and classification of cell-nuclei in 3d volumetric data using voxel-wise gray scale invariants. Proceedings of the 27th DAGM Symposium, in number 3663 LNCS, Springer, Vienna, Austria, 30.8 - 2.9. 2005. (2005)
4. Ronneberger, O.: Libsvm1 - a support vector machine template library. download at: <http://lmb.informatik.uni-freiburg.de/lmbsoft/libsvm1/> (2004)
5. Vapnik, V.N.: The nature of statistical learning theory. Springer (1995)
6. Ronneberger, O., Fehr, J., Burkhardt, H.: Voxel-wise gray scale invariants for simultaneous segmentation and classification – theory and application to cell-nuclei in 3d volumetric data. Internal report 2/05, IIF-LMB, University Freiburg (2005)
7. Groemer, H.: Geometric Applications of Fourier Series and Spherical Harmonics. Cambridge University Press (1996)
8. M. Kazhdan, T.F., Rusinkiewicz, S.: Rotation invariant spherical harmonic representation of 3d shape descriptors. in Symposium on Geometry Processing (2003)
9. Vasconcelos, N.: Feature selection by maximum marginal diversity: optimality and implications for visual recognition. Proceedings of IEEE Conf. on Computer Vision and Pattern Recogniton, Madison, USA (2003)
10. Kurz, H., Papoutsis, M., Wilting, J., Christ, B.: Pericytes in experimental mdamb231 tumor angiogenesis. *Histochem Cell Biol* (2002) 117:527–534

# Learning of Graphical Models and Efficient Inference for Object Class Recognition

Martin Bergtholdt, Jörg H. Kappes, and Christoph Schnörr

Computer Vision, Graphics, and Pattern Recognition Group  
Department of Mathematics and Computer Science  
University of Mannheim, 68131 Mannheim, Germany  
{bergtholdt, jkappes, schnoerr}@uni-mannheim.de

**Abstract.** We focus on learning graphical models of object classes from arbitrary instances of objects. Large intra-class variability of object appearance is dealt with by combining statistical local part detection with relations between object parts in a probabilistic network. Inference for view-based object recognition is done either with  $A^*$ -search employing a novel and dedicated admissible heuristic, or with Belief Propagation, depending on the network size.

Our approach is applicable to arbitrary object classes. We validate this for “faces” and for “articulated humans”. In the former case, our approach shows performance equal or superior to dedicated face recognition approaches. In the latter case, widely different poses and object appearances in front of cluttered backgrounds can be recognized.

## 1 Introduction

Recent research on class-specific object recognition from arbitrary viewpoints has focused on the high intra-class variability of object instances in connection with the recognition of cars, airplanes, motor-bikes [1,2], quadrupeds (cows and horses) [3], faces [1,2], and humans [4,5,6,7,8,9,10,11].

Approaches can be roughly classified into global/holistic and local methods. Global methods model the distribution of objects as a whole using learned templates [4,5] for example, while local methods use local object features and parts in order to better cope with false detections due to occlusions, image clutter, and noise by exploiting recent research on interest point detection and distinctive image features [12,13]. In this context, object features or parts may be organized as “bags of keypoints” ignoring geometric structure entirely [14], or with additional structural constraints between parts [1,2,6,7,8,9,10,11], enabled by the recent progress concerning the inference in graphical models [15]. Often, the relative geometric locations of parts are distinctive for an object class.

In our work, we exploit both local parts and structure for object class recognition. Rather than using computationally convenient tree-models [10,6] which capture only a small fraction of dependencies explicitly, we employ more powerful graphical models to represent relevant relations between parts and to cope

with uncertainties due to clutter, occlusion, and noise. While the corresponding increased computational complexity of inference for object recognition was an obstacle in previous work relying on conventional methods, up-to-date approximate inference algorithms, including Loopy Belief-Propagation or Tree-Reweighted Belief-Propagation, have proved to yield high-quality maximum a posteriori (MAP) optima at moderate computational costs [16].



**Fig. 1. Left, Middle:** Recognition of humans in cluttered background. Edges indicate relations between parts, not pose (see text). **Right:** Recognition of faces.

In this paper, we present a general approach to object class recognition. Based on the probabilistic graphical model described in section 2, we explain how part detectors are learned as well as relations between parts in terms of *geometry and appearance* (section 3). The inference algorithms are described in section 4. Besides the well-known belief propagation (BP), and related to [17], we contribute a *novel admissible heuristic* for applying  $A^*$ -search as an alternative to BP. For sufficiently small-sized networks, the latter always converges, thus returns the global optimum, and with less run-time than BP. On the other hand, BP reliably infers highly probable configurations also for larger networks in fixed time (for fixed problem size). The general applicability of our framework is validated in section 5 for two object classes, “faces” and “articulated humans”. Despite its generality, our approach compares favorably with dedicated face detection algorithms.

## 2 Probabilistic Graphical Model

We want to locate an object with  $S$  parts in an Image  $I$ , with image domain  $\Omega_I \subset \mathbb{C} \times \mathbb{Z}$ . The location of part  $s$  is denoted as  $\mathbf{x}_s \in \Omega_I$ . The configuration of the entire model is therefore  $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_S) \in \Omega = \Omega_I \times \dots \times \Omega_I = \Omega_I^S$  and we want to find the best configuration  $\hat{\mathbf{X}}$  as an MAP-estimate:

$$\hat{\mathbf{X}} = \arg \max_{\mathbf{X} \in \Omega} P(\mathbf{X}|I, G) \quad (1)$$

$G$  refers here to our prior model hypothesis that an object is defined by a pairwise Markov Random Field (MRF) with associated probabilistic graphical structure  $G = (V, E, \lambda)$  where object parts are nodes in  $V$  and relations between parts are edges in  $E$ ;  $\lambda$  denotes a parameter vector for the geometric prior, which is learned using training data. We use *dense* graphs to model the complex relations between parts.

To simplify presentation, we omit  $G$  in the following derivations. Using Bayes' rule, we can factor the posterior probability for the configuration  $P(\mathbf{X}|I)$  as

$$P(\mathbf{X}|I) = \frac{P(I|\mathbf{X})P(\mathbf{X})}{\sum_{\mathbf{X} \in \Omega} P(I|\mathbf{X})P(\mathbf{X})} \propto P(I|\mathbf{X})P(\mathbf{X}) \quad (2)$$

The first term will be denoted as the *appearance* or *data term*, the second as *geometry* or *shape term*. Because we only use *unary* and *binary constraints*, the posterior can also be written as Gibbs distribution  $p(\mathbf{X}|I) \propto \exp(-\mathcal{E}(\mathbf{X}|I))$  with corresponding energy  $\mathcal{E}$  and potential functions  $\psi_s, \psi_{st}$ :

$$\mathcal{E}(\mathbf{X}|I) = \sum_{s \in V} \psi_s(\mathbf{x}_s) + \sum_{st \in E} \psi_{st}(\mathbf{x}_s, \mathbf{x}_t) \quad (3)$$

Section 3 explains how  $\psi_s, \psi_{st}$  depend on  $I, G$ . We point out that each sample space  $\Omega_s$  comprises all locations in the image and that  $\psi_s, \psi_{st}$  are *general* functions learned from data. Therefore, global optimization with polynomial complexity, e.g. by computing graph cuts [18], cannot be applied, and we have to resort to approximate inference (cf. section 4).

**Geometry term.** The geometry for our MRF-representation of the object comprises pairwise terms on edges only

$$P(\mathbf{X}) \propto \prod_{st \in E} (H_{d_{st}}(|\mathbf{x}_s - \mathbf{x}_t|) H_{\gamma_{st}}(\angle(\mathbf{x}_s - \mathbf{x}_t))) \quad (4)$$

where  $H(\cdot)$  denote independent 1D histograms for relative edge-length  $d_{st} = |\mathbf{x}_s - \mathbf{x}_t|$  and absolute edge-direction  $\gamma_{st} = \angle(\mathbf{x}_s - \mathbf{x}_t)$  with respect to the  $x$ -axis, learned from the training set with 30 bins each.

Concerning global object parameters (scale, rotation, and translation), we note, that by using only pairwise relations, our representation is already invariant to global translation. The use of absolute angles makes our model rotation variant. We assume, however, that our images are registered with respect to the ground-plane such that the horizon is parallel to the  $x$ -axis. To account for the scale dependency of relative lengths, we scale-normalize the training images. In a new image we treat scale as hidden and consequently compute the MAP-estimate over a set of discrete scales  $\sigma \in \{\sigma_1, \dots, \sigma_L\}$ .

**Appearance term.** We assume that the image likelihood factors as

$$P(I|\mathbf{X}) \propto \prod_{s \in V} p(I|\mathbf{x}_s) \prod_{st \in E} p(I|\mathbf{x}_s, \mathbf{x}_t) \quad (5)$$

Where the individual terms are functions learned from extracted features

$$\begin{aligned} p(I|\mathbf{x}_s) &\approx \text{Prob}_s(f_s(I, \mathbf{x}_s)) \\ p(I|\mathbf{x}_s, \mathbf{x}_t) &\approx \text{Prob}_{st}(f_{st}(I, \mathbf{x}_s, \mathbf{x}_t)) \end{aligned} \quad (6)$$

$\text{Prob}_s(f_s(I, \mathbf{x}))$  is our approximation to the image likelihood for observing part  $s$  at location  $\mathbf{x}$  against background (likewise for edges  $st$ ). Under the assumption that the presence or absence of a part at a certain image location only depends on a small neighborhood of that location, we compute features  $f_s(I, \mathbf{x})$  from *image patches* in windows of fixed size, see section 3.1, and use a support vector machine (SVM) with Gaussian kernel and probabilistic outputs [19] to compute  $\text{Prob}_s(f_s(I, \mathbf{x}_s))$ . We have used the implementation of [20], performing grid-search to learn optimal SVM-parameters ( $C$  and  $\gamma$ ) using cross-validation.

Assuming independence of part-appearance is certainly not true for very self-similar object parts, e.g. symmetrical body parts like eyes, hands and feet. But the assumption keeps the model tractable and with the additional geometric-information, these ambiguities can in most cases be resolved. Additionally, our SVM-detector will (and should!) give positive detections around the true location of parts due to strong local correlation. To remedy for this effect, we use non-maxima suppression when sampling candidates from the image, see section 3.2, so that the assumptions hold approximately.

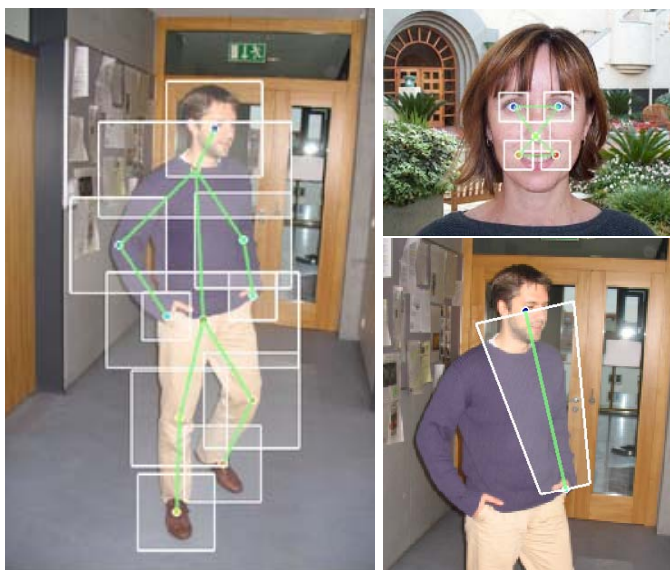
## 3 Supervised Learning and Implementation Details

### 3.1 Appearance Features

Features suitable for our general setting have to meet certain criteria: To facilitate implementation, one type of feature-extractor is to be used for all parts. We also require robustness for: changes in illumination and color; small occlusions and clutter; and minor variations in spatial transformations (translation, rotation, and scale).

A suitable feature descriptor meeting these criteria has been proposed in [12], and variants have already proved successful for object detection compared to other descriptors [5]. The features we use are defined as follows: for each pixel in a sliding window, we compute its gradient orientation  $\theta \in [0, \pi]$ , i.e. modulo  $\pi$ , and for each block of  $8 \times 8$  pixels we accumulate the orientation into one histogram with 8 orientation bins.

Each image patch located over an object part, see fig. 2, is resized to  $32 \times 32$  pixels for which we compute  $4 \times 4$  blocks with 8 orientations, yielding a feature vector of size  $4 \times 4 \times 8 = 128$ . As proposed in [12], we used trilinear interpolation among neighboring bins in  $(x, y, \theta)$  to obtain smooth histograms and normalize the feature vectors to unit length. This whole procedure significantly reduces the dimensionality (e.g. 128 vs.  $32 \times 32 \times 3 = 3072$ ), while meeting our requirements. See fig. 2 for an example labeling of a human and a face image. The white frames correspond to the image patches used for learning.



**Fig. 2.** **Left:** Human with 11 labeled parts. **Right-top:** Face with 5 labeled parts. **Right-bottom:** Patch geometry for edge “head/left hand”.

We have not precomputed local orientation or scale information, but rely on learning these remaining variations from the training set using the SVM. To this end, we increased the number of training samples by factor 10 for faces and by factor 20 for humans, randomly varying the scale in the interval  $[0.8, 1.2]$  and the orientation in  $\pm 10^\circ$  and  $\pm 20^\circ$  for faces and humans respectively. We then computed probabilistic SV-classifiers for each part against background.

For the *pairwise appearance information*, we propose the following: For each edge in the graph, we sample an *oriented* patch using the image locations of the two incident parts  $\mathbf{x}_s$ ,  $\mathbf{x}_t$  and their respective diameters, see the bottom right image in fig. 2 for an illustration. Each patch is then resized to  $32 \times 32$  pixels. The feature vector is computed in the same way as for the single parts, and SVM-learning yields then pairwise appearance probabilities. Note that appearance is computed on all edges of the model graph, not only for the physical links, thus adding necessary redundant information to the model representation. Moreover, the geometry for the pairwise sampling is defined by the incident part-candidates, so features are invariant to rotation and foreshortening along the edge, yielding in general stronger classifiers than the individual part-classifiers.

We have found, that for a multi-scale image analysis, it is necessary to speed up the process of feature generation. We have therefore changed the order of computation in that we first computed for the entire image at each  $8 \times 8$  pixel block the corresponding histogram of orientations. For a single image location we then used linear interpolation in  $(x, y, \theta)$  to obtain the  $4 \times 4 \times 8$  feature-vector.

### 3.2 Determining the Effective Configuration Space

Based on the probabilistic model, we compute a feasible subset of the entire space, the *effective configuration space*. We sample candidate part-locations in the image using non-maxima suppression to account for local correlation of the image likelihood terms (6), and compress probabilistically the remaining hypotheses into a single node for each part, where the missing information is provided by prior estimates as

$$\begin{aligned} P_s(\cdot|I) &= \alpha \mathbb{E}_{\mathbf{x}_s} \{ \exp(-\psi_s(\mathbf{x}_s)) \} \\ P_{st}(\cdot, \cdot|I) &= \alpha \mathbb{E}_{\mathbf{x}_s, \mathbf{x}_t} \{ \exp(-\psi_{st}(\mathbf{x}_s, \mathbf{x}_t)) \} \end{aligned} \quad (7)$$

We take the expectation over our training set and set the penalty parameter  $\alpha$  in our experiments manually (see section 5).

## 4 Inference

We consider the MAP-configuration  $\hat{\mathbf{X}}$  as the best fit of the model to an image. We used two approaches to the combinatorial inference problem (1): Loopy Belief Propagation [21] (BP) and  $A^*$ -search [22,23] ( $A^*$ ) using a novel tree-based admissible heuristic. Concerning BP, we refer the reader to the literature [21,23].

**$A^*$ -Algorithm with a Novel Admissible Tree-Heuristic.** The  $A^*$ -algorithm is an established technique for searching the optimal solution in terms of the shortest path within a graph, representing the whole configuration space [22].

Its performance depends on devising a heuristic for estimating the “future costs” of unexplored paths between two nodes (configurations) for the problem at hand. In order to find the *global* MAP optimum, the heuristic has to be *admissible*, i.e., it always returns a lower bound for the cost of some unexplored path of configurations. While this guaranty for global optimality holds once the search terminates, we do not have *polynomial* time complexity guaranteed.

In previous work [23] we introduced this technique for graphical models with the admissible heuristic (8).

$$\min_{\mathbf{x}_{V \setminus B}, \mathbf{x}_B = b} \left\{ \sum_{s \in V \setminus B} \psi_s(\mathbf{x}_s) + \sum_{st \in E_{11}} \psi_{st}(\mathbf{x}_s, \mathbf{x}_t) \right\} + \sum_{st \in E_{12}} \min_{\mathbf{x}_{st}} \psi_{st}(\mathbf{x}_s, \mathbf{x}_t) \quad (8)$$

Here,  $B$  denotes the subset of already processed nodes in  $V$ .  $E_{11}$  is the set of tree-edges which are not in  $B \times B$ , and  $E_{12}$  contains all edges neither in  $B \times B$  nor in  $E_{11}$ .

A much tighter lower bound is achieved, however, by defining  $E_{21}$  as the union of  $E_{11}$  and all edges in  $B \times (V \setminus B)$ , and  $E_{22}$  as the set of all edges neither in  $B \times B$  nor in  $E_{21}$ . This leads to the novel admissible heuristic:

$$\min_{\mathbf{x}_{V \setminus B}, \mathbf{x}_B = b} \left\{ \sum_{s \in V \setminus B} \psi_s(\mathbf{x}_s) + \sum_{st \in E_{21}} \psi_{st}(\mathbf{x}_s, \mathbf{x}_t) \right\} + \sum_{st \in E_{22}} \min_{\mathbf{x}_{st}} \psi_{st}(\mathbf{x}_s, \mathbf{x}_t) \quad (9)$$



Whereas for (8) it is possible to compute lookup tables in advance, (9) requires re-computation in every exploration step of the  $A^*$ -algorithm. In spite of this apparent disadvantage, the gained tightness of the bound more than compensates for the computational cost as far less exploration steps are necessary to compute the MAP. Moreover, with (8) it is very difficult to cope with hidden/missing nodes.

## 5 Experiments and Discussion

**Data sets.** We have used three object data sets and one background set for learning and evaluation of our model: **The Caltech face dataset** [2] consisting of 450 frontal faces. We used the first 216 frames (14 subjects) as training and the last 234 frames (14 subjects with 3 additional artificial paintings) for testing. **The BioID face dataset** [24] consisting of 1521 face images of 23 subjects, featuring more variation in pose than the Caltech dataset. This dataset was used for testing only, using the model learned from the Caltech training set. **The human dataset**, consisting of 894 images, from various consumer cameras and google image search. From the 894 frames we used 624 for training and 270 for testing. **Background** was obtained from 45 images without people/faces, but featuring scenes where people normally occur. For faces we chose  $\alpha = 0.01$  and for humans  $\alpha = 1$ .



**Fig. 3.** Recognition examples. **Top row:** BioID faces frame#(rank): B11(1465), B416(1130), B428(568), B1464(484). **Bottom row:** Caltech faces C289(166), C327(38), C402(202), C429(92). A  $\circ$  denotes a found part,  $\times$  are geometrically inferred missed parts. None of these persons was part of the training set. Note the difficulty of these particular images due to partial occlusion (C289, C327), illumination (C327), and “abstractness” (C429).

**Optimization.** We applied  $A^*$ -search and BP to all recognition experiments. While  $A^*$  always converged and detects faces quickly (mean: 0.008 seconds), BP needs less run-time on average for the larger network (complete graph with  $|V| = 11$ ) used to recognize humans. Run-times vary between 20 seconds and 0.5 hours for  $A^*$ , and between 3 seconds and 2 minutes for BP.

**Quality measure.** To measure the quality of our results, let  $m_s = \frac{|\mathbf{x}_s - \mathbf{x}_s^*|}{|\mathbf{x}_{l\text{-eye}}^* - \mathbf{x}_{r\text{-eye}}^*|}$  denote the point-to-point error for part  $s$  relative to the distance of the eyes, where the  $*$  denotes the ground truth location. Images in fig. 4 are ranked by the maximal error of a single part  $m_{\max} = \max_{s \in V} m_s$  in descending order, so ranking is from worst=1 to best=1521 (BioID), 234 (Caltech). To compare our results in table 1 to [25] on the BioID dataset, we also included the measure  $m_{e4} = \frac{1}{4} \sum_{s \in V'} m_s$ , where  $V' = \{\text{l-eye, r-eye, l-mouth, r-mouth}\}$ , i.e., our original nodes without the nose. We assume a *hit* if the quality measure is below a given *tolerance*. Comparable hit-rates reported by [25] (estimated electronically from their plots) are  $\approx 0.94$  for  $m_{e4} < 10\%$  and  $\approx 0.97$  for  $m_{e4} < 15\%$ , where we achieved hit-rates of 0.9178 and 0.9908, respectively. We also give mean error and variances for each part over the whole image set.

For the Caltech face dataset, we used the same training images as [17], whereas for testing they excluded frames 328–336 (smaller scale) and 400, 402, 403 (paintings) which we kept in our test set. Note that we search faces at multiple scales and our method generalizes to the paintings, see e.g. fig. 3. [17] report a hit-rate of 0.92, but without mentioning a corresponding tolerance level or quality measure. In our tests we achieved a hit rate of 0.92 for  $m_{\max} < 16.7$ .

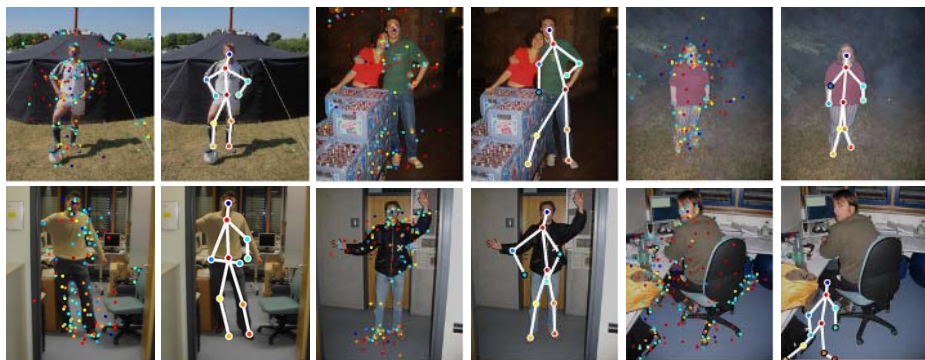
The BioID dataset was processed with exactly the same model learned from the 216 training images of the Caltech set. Typical examples for the two test sets are given in fig. 3. Some images with low rank are shown in fig. 4. For the Caltech set these are actually the ones with largest  $m_{\max}$ .

**Table 1.** Hit rates for the different face parts in the BioID and Caltech datasets. The increased errors for the nose in BioID as compared to Caltech, are due to a slightly different labeling scheme of our training set compared to the provided labels of BioID causing a systematic bias. Overall we can attest excellent performance of our *general* approach on these two unknown datasets.

BioID faces												
Tolerance	3%	6%	9%	12%	15%	18%	21%	24%	27%	30%	Mean Error	Var Error
Left eye	0.06	0.53	0.85	0.96	0.99	0.99	0.99	0.99	0.99	0.99	5.62%	0.52%
Right eye	0.26	0.56	0.80	0.90	0.93	0.96	0.98	0.99	0.99	0.99	6.41%	0.41%
Nose	0.07	0.29	0.52	0.72	0.82	0.89	0.93	0.96	0.98	0.99	10.06%	0.65%
Left mouth	0.17	0.49	0.72	0.87	0.95	0.98	0.98	0.99	0.99	0.99	7.11%	0.49%
Right mouth	0.22	0.54	0.74	0.87	0.95	0.98	0.99	0.99	0.99	0.99	6.75%	0.48%
$m_{e4}$	0.06	0.53	0.85	0.96	0.99	0.99	0.99	0.99	0.99	0.99	6.47%	0.33%
Caltech faces (test set only)												
Tolerance	3%	6%	9%	12%	15%	18%	21%	24%	27%	30%	Mean Error	Var Error
Left eye	0.57	0.86	0.91	0.95	0.98	0.98	0.98	0.98	0.98	0.98	4.52%	1.93%
Right eye	0.50	0.83	0.88	0.94	0.98	0.98	0.98	0.98	0.98	0.98	4.82%	1.80%
Nose	0.29	0.63	0.78	0.85	0.93	0.97	0.98	0.98	0.98	0.98	6.85%	1.99%
Left mouth	0.14	0.48	0.65	0.85	0.94	0.97	0.98	0.98	0.98	0.98	8.31%	2.31%
Right mouth	0.20	0.51	0.68	0.85	0.95	0.98	0.98	0.98	0.99	0.99	7.73%	2.29%
$m_{e4}$	0.13	0.66	0.94	0.98	0.99	0.99	0.99	0.99	0.99	0.99	6.34%	1.97%



**Fig. 4.** Bad recognitions. For each image pair, the left image is part candidates, right image is MAP-result. **Top row:** BioID frame#(rank): B486(1), B600(40), B1294(2). **Bottom row:** Caltech C320(2), C325(1), C417(3). For the ranking see text.



**Fig. 5.** Positive and negative recognition results for humans. **Left images:** part candidates. **Right images:** MAP-result with BP.

**Table 2.** Hit rates for the human data set. We have only used the images from the test set with a single human. Recognizing humans is much harder than faces. Especially hands are very hard to detect without color and the geometric prior cannot always resolve their position.

<i>Tolerance</i>	10%	20%	30%	40%	50%	Mean Error	Var Error
Head	0.47	0.69	0.83	0.87	0.91	20.68%	7.89%
Chest	0.63	0.79	0.88	0.91	0.91	18.07%	6.54%
Elbows	0.28	0.49	0.67	0.80	0.86	28.45%	7.72%
Hands	0.29	0.43	0.50	0.60	0.66	47.33%	24.82%
Hip	0.37	0.65	0.81	0.91	0.93	20.02%	4.05%
Knees	0.51	0.74	0.81	0.86	0.93	17.23%	3.33%
Feet	0.46	0.69	0.84	0.87	0.88	21.27%	7.11%
$m_{e11}$	0.15	0.49	0.76	0.80	0.89	26.20%	3.44%

Recognition of humans is much harder, because geometry is much less constrained, and because object parts are far less discriminative without the context. Locating an elbow or a hand in an image (without color information) turned out

to be quite challenging. The contextual information provided by our graphical model, however, helps a lot for resolving the ambiguities caused by false detections – compare the images with part candidates only and their corresponding MAP-result in fig. 5. However, a similar quality as for the face data sets cannot be expected; see table 2 for the hit-rates, where the tolerance levels for humans are relative to the distance between chest and hip. Failures are mainly due to the unknown scale, or due to a complete breakdown of part detectors. Fig. 5, bottom, shows some intricate examples.

## 6 Conclusion

We presented a general model for object class recognition. Our work demonstrates the feasibility of view-based object recognition even for articulated humans if a sufficiently rich data base is available for learning. The evaluation for different object classes showed a performance competitive to approaches that *only* work for a specific object class.

Our future work will focus on the real-time performance of all components, and on an approach to enlarge the learning data base with minimal supervision.

## References

1. Fergus, R., Perona, P., Zisserman, A.: A sparse object category model for efficient learning and exhaustive recognition. In: CVPR. (2005)
2. Weber, M., Welling, M., Perona, P.: Unsupervised learning of models for recognition. In: ECCV. (2000) 18–32
3. Kumar, M.P., Torr, P.H.S., Zisserman, A.: Extending pictorial structures for object recognition. In: BMVC. (2004) 789–798
4. Gavrila, D., Philomin, V.: Real-time object detection using distance transforms. In: Proc. Intelligent Vehicles Conf. (1998.)
5. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR. (2005) 886–893
6. Felzenszwalb, P., Huttenlocher, D.: Pictorial structures for object recognition. IJCV **61**(1) (2005) 55–79
7. Mikolajczyk, K., Schmid, C., Zisserman, A.: Human detection based on a probabilistic assembly of robust part detectors. In: ECCV, Springer (2004)
8. Ren, X., Berg, A., Malik, J.: Recovering human body configurations using pairwise constraints between parts. In: ICCV. (2005)
9. Sigal, L., Isard, M., Sigelman, B., Black, M.: Attractive people: Assembling loose-limbed models using non-parametric belief propagation. In: NIPS. (2003)
10. Ronfard, R., Schmid, C., Triggs, B.: Learning to parse pictures of people. In: ECCV. (2002) 700–714
11. Ramanan, D., Forsyth, D.A., Zisserman, A.: Strike a pose: Tracking people by finding stylized poses. In: CVPR. Volume 1. (2005) 271–278
12. Lowe, D.: Distinctive image features from scale-invariant keypoints. IJCV **60**(2) (2004) 91–110
13. Mikolajczyk, K., Schmid, C.: Scale & affine invariant interest point detectors. IJCV **60**(1) (2004) 63–86

14. Sivic, J., Russell, B., Efros, A., Zisserman, A., Freeman, W.: Discovering objects and their locations in images. In: ICCV. IEEE (2005)
15. Frey, B., Jojic, N.: A comparison of algorithms for inference and learning in probabilistic graphical models. *IEEE PAMI* **27**(9) (2005) 1392–1416
16. Szeliski, R., Zabih, R., Scharstein, D., Veksler, O., Kolmogorov, V., Agarwala, A., Tappen, M., Rother, C.: A comparative study of energy minimization methods for markov random fields. In: ECCV. (2006)
17. Pham, T., Smeulders, A.: Object recognition with uncertain geometry and uncertain part detection. *CVIU* **99**(2) (2005) 241–258
18. Kolmogorov, V., Zabih, R.: What energy functions can be minimized via graph cuts? *IEEE PAMI* **26**(2) (2004) 147–159
19. Platt, J.: Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In: *Advances in Large Margin Classifiers*. MIT Press (2000) 61–74
20. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. (2001)
21. Yedidia, J.S., Freeman, W.T., Weiss, Y.: Constructing free-energy approximations and generalized belief propagation algorithms. *IEEE Trans. Information Theory* **51**(7) (2005) 2282–2312
22. Hart, P., Nilsson, N., Raphael, B.: A formal basis for the heuristic determination of minimum cost paths. *IEEE Tr. Syst. Sci. Cybernetics* **4** (1968) 100–107
23. Bergtholdt, M., Kappes, J., Schnörr: Graphical knowledge representation for human detection. In: *International Workshop on The Representation and Use of Prior Knowledge in Vision*. (2006)
24. Jesorsky, O., Kirchberg, K., Frischholz, R.: Robust face detection using the hausdorff distance. In Bigun, J., Smeraldi, F., eds.: *Audio and Video based Person Authentication*, Springer (2001) 90–95
25. Cristinacce, D., Cootes, T.F., Scott, I.: A multi-stage approach to facial feature detection. In: *BMVC*. (2004)

# Properties of Patch Based Approaches for the Recognition of Visual Object Classes

Alexandra Teynor<sup>1</sup>, Esa Rahtu<sup>2</sup>, Lokesh Setia<sup>1</sup>, and Hans Burkhardt<sup>1</sup>

<sup>1</sup> University of Freiburg, Department of Computer Science  
Georges Köhler Allee 052, 79110 Freiburg, Germany

{teynor, setia, Hans.Burkhardt}@informatik.uni-freiburg.de

<sup>2</sup> University of Oulu, Department of Electrical and Information Engineering,  
PO Box 4500, 90014 Oulu, Finland  
erahtu@ee.oulu.fi

**Abstract.** Patch based approaches have recently shown promising results for the recognition of visual object classes. This paper investigates the role of different properties of patches. In particular, we explore how size, location and nature of interest points influence recognition performance. Also, different feature types are evaluated. For our experiments we use three common databases at different levels of difficulty to make our statements more general. The insights given in the conclusion can serve as guidelines for developers of algorithms using image patches.

## 1 Introduction

The amount of digital documents increases daily, and with it the need to organize this torrent of data in order to retrieve something again. Especially for digital images no ideal solution has been found yet. The manual annotation of images is very labor intensive, so the vast majority of images will remain unannotated. Techniques for content based image retrieval (CBIR) are able to find similar images based on pixel content only, however, usually the definition of similarity is on a color and texture level, not on a semantic level. Most users do not want to find things with just the same texture and color, but want to find semantic entities, images with particular objects like cows, sheep or cars. This is why the main focus of research is now drawn to the recognition of visual object classes rather than the already widely researched area of traditional CBIR, as surveyed e.g. in [1].

### 1.1 Basic Principles

Currently, the most promising approaches for the recognition of visual object classes are based on the use of image patches. The advantages are easy to see: local representations can deal with variability in object shape and partial occlusions. The majority of these approaches follow an easy basic pattern: first, points or areas of high information content become identified in images. For this, so

called interest point detectors or covariant region detectors are used. A survey about them can be found, e.g. in [2]. In the next step, features get extracted from these locations. Now models can be built for each class to be recognized or the feature vectors can be used directly. Depending on the model, different classifiers (e.g. SVMs [3], Winnows [4], Bayes [5]) can be used.

## 1.2 Related Work

A fair amount of work has already been done using image patches for classification and retrieval. One of the early approaches was by C. Schmid et al. [6]. She proposed calculating local gray value invariants at interest points for image retrieval. Weber et al. [7] and Fergus et al. [5] introduced a so called “constellation model”, i.e. image patches in a probabilistic spatial arrangement, to decide whether a certain object is present in a scene or not. Agarwal et al. [4] classified and localized objects in an image using binary vectors coding the occurrences and spatial relations of patches. Leibe et al. showed in [8] a method to simultaneously categorize and segment objects using an implicit shape model. D. Lowe [9] proposed highly distinctive SIFT features in order to detect objects reliably in a scene. More recent work on this topic was conducted, e.g. by Deselaers et al. [10], who used histograms of patch cluster memberships in order to compare different classification methods. Opelt et al. [11] used a great variety of features and classifiers in a boosting framework to distinguish the best choice for each class.

The authors of the previously mentioned works had to decide at some point where to take patches, how many and at which size. Most of these decisions were done empirically in the course of the work, or were predetermined by the models chosen. E.g. the joint probability model used in the constellation model prohibits the use of much more than 7 parts. Only few works that we are aware of deal explicitly with these questions, e.g., Deselaers et al. [10] conducted some experiments for different patch sizes. The choice of the local descriptor type was investigated in [12] for matching and in [13] for object categorization. However, there the size of the patches is selected by automatic detectors, which does not necessarily mean that the size is optimal for object categorization.

In this work, we want to investigate how the factors number and size of image patches, descriptor type and nature of interest points influence retrieval quality.

## 1.3 Outline of the Paper

After the introductory section we briefly describe the types of local descriptors and the interest point detectors used. In section 3 we explain the test setting and experiments we performed. In section 4 we describe the results of our experiments and discuss them. In the last section we set out our conclusions and give recommendations for developers of patch based approaches for object categorization.

## 2 Methods and Algorithms

### 2.1 Interest Point and Affine Covariant Area Detection

In most of our experiments we use an interest point detector to find prominent places for the extracted patches. Some guidelines on the choice of the detector can be found from evaluation papers like [2,14,12], but these concentrate mainly on repeatability and information content of such points, which is not necessarily the key issue in patch based object categorization. For the majority of the experiments we selected the wavelet based interest point detector presented by Loupas and Sebe in [15]. This choice was motivated by good results in evaluation papers [14], image retrieval [16] and object categorization [10]. Of course other detectors, especially scale invariant ones could have been used, however we wanted to see the direct influence of patch size. An extensive region detector evaluation was out of scope for this paper.

In addition to the location, the ideal shape of the patch is also a question. Simple approaches use round or square patches centered at interest locations, more sophisticated solutions use affine region detectors. To test how they perform compared to each other, we selected two affine region detectors as examples: the Harris-Affine detector [17] and the maximally stable extremal regions (MSER) detector [18].

### 2.2 Feature Extraction

Once interest points or covariant areas have been found, features can be extracted. In the following, we briefly describe the features used in this evaluation. Some of them are subject to a PCA (principal component analysis) in order to get a more compact representation, details about this can be found in section 4.1

**Gray values:** The simplest way to get a description of the area around the interest point is to directly use the gray values in a window with side length  $2d+1$  ( $d$  being the patch radius) centered around the interest point.

**Multi-Scale Autoconvolution:** The Multiscale Autoconvolution (MSA) is an  $\mathbf{R}^2 \rightarrow \mathbf{R}^2$  mapping which is invariant with respect to affine transformations of the input function. This makes it possible to use MSA transform values as features for affine invariant classification. The basic idea behind MSA is to apply probabilistic approaches to the affine coordinate system. For an image function  $f(x, y)$  the MSA transform is

$$If(\alpha, \beta) = E[f(\alpha(x_1 - x_0) + \beta(x_2 - x_0) + x_0)], \quad (1)$$

where  $\alpha, \beta \in \mathbf{R}$ ,  $E$  is the expected value and  $x_0, x_1, x_2$  are random points with probability density given by  $f(x, y)/\|f(x, y)\|_{L^1}$ . A comprehensive introduction to MSA can be found in [19].

**Haar integral based invariants:** Schulz-Mirbach [20] introduced image features based on Haar integrals invariant to transformation groups. These are



constructed as follows: Let  $\mathbf{M} = \mathbf{M}(i, j), 0 \leq i < N, 0 \leq j < M$  be an image, with  $\mathbf{M}(i, j)$  representing the gray-value at the pixel coordinate  $(i, j)$ . Let  $G$  be the transformation group of translations and rotations with elements  $g \in G$  acting on the images, such that the transformed image is  $g\mathbf{M}$ . An invariant feature must satisfy  $F(g\mathbf{M}) = F(\mathbf{M}), \forall g \in G$ . Such invariant features can be constructed by integrating  $f(g\mathbf{M})$  over the transformation group  $G$

$$I(\mathbf{M}) = \frac{1}{|G|} \int_G f(g\mathbf{M}) dg$$

which for a discrete image is approximated using summations. By using  $k$  different kernel functions  $f$  we get a  $k$ -dimensional feature vector for each location.

**Scale Invariant Feature Transform (SIFT):** Scale invariant feature transform (SIFT) introduced by Lowe in [21] is based on histograms of Gaussian weighted gradient orientations around scale invariant interest points. To be more comparable, we did not use the SIFT built-in interest point detector, but the same locations and scales as for the other features.

### 3 Databases and Test Setting

For our evaluation we used 3 image databases at different levels of difficulty. We only used gray value information. The most simple database is the ETH80 database introduced in [22]. Here 10 different objects from 8 different object classes are photographed in front of a uniform background. For each object, 41 views are taken at different angles. For this database, the classifier had to decide which of the 8 object classes is present. Tests were performed in a leave-one-object-out approach.

The second image sets are from the Caltech dataset <sup>1</sup>. We chose to take the most commonly used collections “airplanes\_side” (1074 images), “faces” (450 images) and “motorbikes\_side” (826 images). For this database an object present/absent task has to be solved. As a counter class, a set of mixed “background” (900 images) images is used. The individual objects differ in appearance and location, but are about the same size and orientation. The background is cluttered. We divided each collection randomly into two halves, from which one was used for training and the other one for testing.

A clearly more difficult categorization task is present in the Graz02 database<sup>2</sup>. This database has four object categories: “cars” (420), “persons” (311 images), “bikes” (365 images) and a so-called “none” category (380 images) which was used as a counter class. In all the categories, objects suffer from severe occlusions and have a highly variable appearance and pose, reflecting real world scenes more accurately. Experiments performed with this database used the same setting introduced with the Caltech database. Some example images from the three databases can be seen in Figure 1.

<sup>1</sup> <http://www.robots.ox.ac.uk/~vgg/data3.html>

<sup>2</sup> [http://www.emt.tugraz.at/~pinz/data/GRAZ\\_02](http://www.emt.tugraz.at/~pinz/data/GRAZ_02)



**Fig. 1.** Sample images: ETH80 (left), Caltech (middle), Graz02 (right)

### 3.1 Classification Procedure

In this paper our goal was to examine properties of patches and features extracted from them, so we wanted to keep the classification procedure simple. To do this, we apply a nearest neighbor classifier with a suitable distance measure. We first fit a multivariate Gaussian distribution to all feature vectors of each image, obtaining a mean vector  $\mu$  and the full covariance matrix  $\Sigma$ . To determine the distance, we use the symmetric form of the Kullback-Leibler Divergence, for which a closed form expression can be derived:

$$KL[p_1(x)||p_2(x)] = \int p_1(x) \log \frac{p_1(x)}{p_2(x)} dx + \int p_2(x) \log \frac{p_2(x)}{p_1(x)} dx \quad (2)$$

## 4 Experimental Results and Discussion

In the following, we show the outcome of our experiments. Due to space constraints, in the result tables for the ETH80 database, the apple and horse class are not shown, since these objects are similar in appearance to tomato and pear or horse and dog respectively. All categories are contained in the “all” column.

### 4.1 Feature Types

One of the first questions is which type of features to select. Different feature types have different properties for different tasks. We tested various features already described in section 2 for their suitability for object class recognition. For this experiment, the radius of the patches was 20, we used the Louprias interest point detector and 100 points were selected per image. Interest points closer to the border than the radius were omitted. The gray value features were reduced to 20, the SIFT and MSA features to 10 dimensions via PCA, since an estimation of a multivariate Gaussian distribution with full dimension would be too imprecise. The dimensions were chosen by the amount of variance covered by the corresponding eigenvectors. The different dimensionality of the features is due to the difference in initial feature size. For the Haar integral features, we used 20 kernel functions. The results are summarized in table 1.

For the features tested, the SIFT features performed best for the ETH80 database and the Caltech database. The results are especially good if we think about the simplicity of our classifier. The gray value based features performed in the upper range for all three databases, making them suitable for systems in need of simple feature extraction methods. The MSA and Haar invariants did

**Table 1.** Classification rate of different feature types (in %)

ETH80	car	cow	cup	dog	pea	tom	all	Caltech database			Graz02 database			
								airp.	faces	mot.	bike	cars	pers.	
Gray	86.6	49.5	75.9	70.2	93.2	90.7	73.9	Gray	91.1	93.0	90.7	72.7	68.8	79.5
MSA	71.0	70.0	65.6	52.7	86.1	90.7	68.4	MSA	85.8	91.1	91.9	72.7	67.3	73.1
Haar	81.7	66.3	64.3	62.4	89.0	67.6	67.0	Haar	90.0	91.0	90.6	71.3	63.0	70.8
SIFT	85.9	56.1	78.5	55.9	98.3	88.1	74.6	SIFT	97.3	95.9	91.4	71.9	58.8	67.1

not perform as well for this task. We can also notice that the results for the ETH80 database are worse than the global approaches introduced in [22]. For nearly segmented, unoccluded objects, global methods work better.

## 4.2 Patch Size

Another important question is the patch size. If we select it too small, we are in danger of getting unspecific parts, if we select it too big, we might end up with patches that no longer have generalization capabilities. In this experiment, we use gray values as baseline features (PCA, 20D unless otherwise stated). When judging the results, we have to keep in mind that smaller patches do not lose as much information with PCA as larger patches, since the initial data size is much smaller. In the Caltech database, the motorbikes have the same size relative to the image, but the image sizes vary, so we scaled them to the same height of 250. The objects in Graz02 database are of very different size, so we omit this database for this experiment, since no single patch size makes sense here.

**Table 2.** Classification rate of different patch size (in %)

ETH80	car	cow	cup	dog	pear	tom	all	Caltech	airp.	faces	mot.
2 (10D)	93.9	40.7	78.3	46.1	85.9	63.2	64.2	2 (10D)	96.9	94.4	97.6
5	91.5	40.7	75.4	63.9	80.2	44.4	63.7	5	95.9	92.6	97.2
10	86.1	49.5	78.3	77.8	88.8	72.9	71.5	10	94.9	86.4	95.6
20	86.6	49.5	75.9	70.2	93.2	90.7	73.9	20	93.3	86.5	94.6
30	85.9	51.7	74.4	63.9	97.3	94.6	74.1	25	93.7	87.6	93.7

For the segmented objects in the ETH80 database, on average bigger patches perform better. Looking at the classification results for different objects reveals more details: for rather uniform objects with a smooth outline like pears or tomatoes, bigger patches clearly perform better. This is likely because a bigger part of the silhouette carries more information, the smaller the parts we have the more similar they are. For more detailed objects, small parts usually work better. For the Caltech database, smaller patches seem to work best in all cases, since we do not have smooth objects there. Figure 2 gives us an impression how a patch looks at the same interest point in different sizes.



**Fig. 2.** Example patch for a motorbike wheel with radius 5, 10 and 20

### 4.3 Number of Interest Points

The next question we address is the number of interest points. In this experiment, we take the  $N$  most salient points given by the Loupias detector. At these points the gray values are taken in a window with radius 20, the feature dimension is reduced to 20 via PCA.

**Table 3.** Classification rate for different numbers of interest points (in %)

ETH80	car	cow	cup	dog	pear	tom	all	Caltech database			Graz02 database			
								airp.	faces	mot.	bike	cars	pers.	
20(10D)	63.7	40.7	69.3	51.0	87.1	92.2	63.1	20(10D)	87.7	88.3	85.8	61.1	61.3	71.1
50	83.4	40.0	72.0	68.5	88.8	92.9	71.4	50	92.7	93.0	89.3	74.0	67.0	74.0
100	86.6	49.5	75.9	70.2	93.2	90.7	73.9	100	91.0	93.0	90.9	72.7	68.8	79.5
200	88.3	52.4	73.2	65.6	94.6	89.8	74.4	200	90.6	91.1	92.0	73.7	66.3	76.9
500	88.8	52.4	74.9	60.7	95.9	91.7	75.0	500	89.8	90.1	91.9	74.3	67.8	74.9

When dealing with objects in front of a uniform background as in the ETH80 database, taking more interest points converges to an optimum for high numbers, since most of the patches convey object information. For databases with a (highly) cluttered background this is no longer the case. An intermediate range of about 100 interest points has shown to be sufficient, given our classification method and these databases. Taking too many Loupias interest points usually means taking more background clutter. Results are listed in table 3, in figure 3 we illustrate the area that is covered by 20, 50 and 200 interest points for a sample image.



**Fig. 3.** Area covered by  $N$  most salient points,  $N=20, 50$  and  $200$

### 4.4 Interest Points vs. Random Points

What is the role of the interest point detector in the selection of the patches? Does it give a clear advantage over taking random points? The following experiment should clarify this. We calculate the feature vectors (again for simplicity PCA reduced gray values, window radius 20, 20 dimensions) at a varying number of random points.

**Table 4.** Classification rate for different numbers of random points (in %)

ETH80	car	cow	cup	dog	pear	tom	all	Caltech database			Graz02 database			
								airp.	faces	mot.	bike	cars	pers.	
50(10D)	24.6	29.5	52.9	26.3	52.0	85.4	45.2	50(10D)	88.5	87.9	82.0	65.7	58.0	69.7
100	24.1	27.1	58.3	34.4	25.1	74.9	43.8	100	87.8	89.6	86.2	67.6	57.5	73.7
200	51.0	34.6	72.4	42.7	62.4	79.5	57.4	200	91.0	91.9	87.8	68.9	57.5	68.8
500	70.7	49.5	75.6	45.9	80.0	91.0	67.3	500	92.4	90.5	85.1	70.2	66.8	74.3
1000	78.3	48.8	77.3	53.2	88.8	92.0	71.2	1000	92.4	91.4	84.9	73.5	65.3	73.1

For our experiments, computing features at interest points is superior to random points, as can be seen in table 4. This is especially visible at the ETH80 and the Graz02 databases. Even for 1000 random points, the classification accuracy obtained with fewer interest points cannot be achieved. The exception to the rule is the airplanes category in the Caltech database. For this dataset, a uniform background (=sky), where no interest points are found, is a discriminative property. This confirms that context information can be beneficial for categorization. For the faces class, starting from 200 points, it does not make a difference whether to take interest or random points.

#### 4.5 Shape of Interest Points - Fix vs. Affine Invariant

In our last experiment, we wanted to see whether it is beneficial to use features calculated from covariant regions instead of using windows of fixed geometry (squares or circles). A problem with fixed patches is that their content might change considerably when the viewing angle or the scale of an object changes, however, the automatically detected orientations and scales do not need to be ideal for categorization. We tested the affine harris detector and the MSER detector, together with two feature extraction methods, SIFT and MSA. As MSA is affine invariant, it can be directly applied to the patches. For SIFT features, the elliptical regions have to be normalized to circles. For the calculation, we used the binaries provided by C. Schmid and K. Mikolajczyk<sup>3</sup>. The number of interest points detected by these detectors varied a lot depending on the image. We used parameters so that around 100-400 patches were found. This number is slightly higher than in the case with fixed geometry, since many of the affine covariant areas were too small to cover the object adequately.

The final classification results for the Caltech and the Graz02 databases are shown in table 5, together with corresponding results for a fixed geometry. We had to omit the ETH80 database, because the region detectors were not able to find reasonable regions from all of the images. Some objects, like pears or tomatoes, seem to be too smooth for covariant detectors to converge. Especially for the SIFT features, the combination with the MSER detector seems to have a clear advantage over fixed patches. However, the classification performance did not improve in all cases. Especially using the harris affine detector degraded

<sup>3</sup> <http://www.robots.ox.ac.uk/~vgg/research/affine/>

**Table 5.** Classification rates for different affine covariant patch detectors in % (ha = harris affine, mr = MSER)

	Caltech			Graz02		
	airp.	faces	mot.	bike	cars	pers
MSA	85.8	91.1	91.9	72.7	67.3	73.1
MSA ha	79.0	84.6	89.9	69.4	54.3	67.3
MSA mr	75.9	75.1	87.1	69.2	62.8	59.8

	Caltech			Graz02		
	airp.	faces	mot.	bike	cars	pers
SIFT	97.3	95.9	91.4	71.9	58.8	67.1
SIFT ha	83.5	75.7	78.8	73.7	57.5	63.9
SIFT mr	95.0	97.0	96.9	74.5	60.8	62.7

the results. We assume that the stable invariant areas found are not necessarily optimal in a categorization sense.

## 5 Conclusions

In this paper we addressed some fundamental questions about the use of patches in the categorization of visual object classes. We could show that feature type, size, number, shape and location of patches does influence the retrieval performance, in some cases significantly. The selection of the feature type depends on the image class to be recognized. This confirms that an automatic selection procedure for features as introduced by Opelt et al. in [11] is beneficial in order to get optimal results.

For detailed objects, smaller patches usually work better, for smooth and uniform objects, bigger patches are necessary to cover object information. Interest point detectors are preferable over random selection to determine the location for patches, as good retrieval results can be achieved with relatively few patches, at least for our simple classifier. This is especially true for images with prominent objects or segmented images, and holds less for images with much background clutter. Only in extreme cases, random selection is superior, especially when homogeneous areas, where no interest points are found, are discriminative. An intermediate number of interest points (usually a few hundred) should be extracted from moderately cluttered images, taking too many or too few points spoils recognition performance here. For segmented images, taking more patches converges to some optimum, since no corruptive background patches spoil recognition accuracy.

Affine covariant methods provide an elegant way to choose the shape of a patch, increasing the performance on some occasions. An interesting research issue is to further investigate to what extent the automatically chosen areas are advantageous for object categorization.

**Acknowledgment.** This work has been funded by the German Federal Ministry of Education and Research, project I-Search, grant No. 01IRB02B and the Muscle NoE, contract No. 507752.

## References

1. Santini, S., Gupta, A., Smeulders, A., Worring, M., Jain, R.: Content based image retrieval at the end of the early years. **22** (2000) 1349–1380
2. Schmid, C., Mohr, R., Bauckhage, C.: Evaluation of interest point detectors. *IJCV* **37** (2000) 151–172
3. Wallraven, C., Caputo, B., Graf, A.: Recognition with local features: the kernel recipe. In: Proc. ICCV. (2003)
4. Agarwal, S., Awan, A., Roth, D.: Learning to detect objects in images via a sparse, part-based representation. *IEEE TPAMI* **26** (2004) 1475–1490
5. Fergus, R., Perona, P., Zisserman, A.: Object class recognition by unsupervised scale-invariant learning. In: Proc. CVPR. Volume 2., Madison, WI (2003) 264–27
6. Schmid, C., Mohr, R.: Local greyvalue invariants for image retrieval. *IEEE TPAMI* **19** (1997) 530–535
7. Weber, M., Welling, M., Perona, P.: Unsupervised learning of models for recognition. In: Proc. ECCV, Dublin, Ireland (2000)
8. Leibe, B., Schiele, B.: Interleaved object categorization and segmentation. In: Proc. BMVC, Norwich, UK (2003)
9. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *IJCV* **60** (2004) 91–110
10. Deselaers, T., Keysers, D., Ney, H.: Discriminative training for object recognition using image patches. In: Proc. CVPR. Volume 2., San Diego, CA (2005) 157–162
11. Opelt, A., Pinz, A., Fussenegger, M., Auer, P.: Generic object recognition with boosting. *IEEE TPAMI* **28** (2006) 416–431
12. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. *IEEE TPAMI* **27** (2005) 1615–1630
13. Mikolajczyk, K., Leibe, B., Schiele, B.: Local features for object class recognition. In: Proc. ICCV. Volume 2. (2005) 1792–1799
14. Sebe, N., Lew, M.S.: Comparing salient point detectors. *PR Letters* **24** (2003) 89–96
15. Loupias, E., Sebe, N.: Wavelet based salient points for image retrieval. Technical report, Laboratoire Reconnaissance de Formes et Vision, INSA Lyon (1999)
16. Halawani, A., Burkhardt, H.: Image retrieval by local evaluation of nonlinear kernel functions around salient points. In: Proc. ICPR. (2004)
17. Mikolajczyk, K., Schmid, C.: Scale and affine invariant interest point detectors. *IJCV* **60** (2004) 63–86
18. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. In: Proc. BMVC, Cardiff, UK (2002)
19. Rahtu, E., Salo, M., Heikkilä, J.: Affine invariant pattern recognition using multi-scale autoconvolution. *IEEE TPAMI* **27** (2005) 908–918
20. Schulz-Mirbach, H.: Anwendung von Invarianzprinzipien zur Merkmalgewinnung. PhD thesis, TU Hamburg-Harburg (1995) Reihe 10, Nr. 372, VDI-Verlag.
21. Lowe, D.G.: Object recognition from local scale-invariant features. In: Proc. ICCV, Corfu, Greece (1999) 1150–1157
22. Leibe, B., Schiele, B.: Analyzing contour and appearance based methods for object categorization. In: Proc. CVPR, Madison, WI (2003)

# Feature Selection for Automatic Image Annotation

Lokesh Setia and Hans Burkhardt

Albert-Ludwigs-University Freiburg  
79110 Freiburg im Breisgau, Germany  
{setia, burkhardt}@informatik.uni-freiburg.de

**Abstract.** Automatic image annotation empowers the user to search an image database using keywords, which is often a more practical option than a query-by-example approach. In this work, we present a novel image annotation scheme which is fast and effective and scales well to a large number of keywords. We first provide a feature weighting scheme suitable for image annotation, and then an annotation model based on the one-class support vector machine. We show that the system works well even with a small number of visual features. We perform experiments using the Corel Image Collection and compare the results with a well-established image annotation system.

## 1 Introduction

The amount of available multimedia data is continuously on the rise. With this arises the need to be able to locate existing data effectively. Data which cannot easily be found is as good as lost. Multimedia search differs from text search in that the results are much more subjective, and exact matches are normally not possible. For digital images, a lot of research has been done in the field of “Content-Based Image Retrieval” (CBIR) in the past decade. A user typically searches a CBIR database using the **query-by-example** paradigm, and the CBIR system bases its search on visual features extracted from the image. A big obstacle for CBIR to gain mainstream acceptance has been the so-called *semantic-gap* problem [1,5], though it can be somewhat reduced using *relevance-feedback* techniques [2,3]. Another practical problem in CBIR is that the user may not have a query image available.

A metadata search system on the other hand bases its search on image metadata, such as date and place of creation, image size, other image acquisition parameters, and on image keyword-annotation. Here, the database images are typically manually annotated with keywords, a task which is very time-consuming and also subjective. For large databases, this is simply prohibitively expensive. Automatic Image Annotation tries to bridge these two approaches, in that it works on the content of the images, but gives the user a possibility to perform a metadata search. Of course, semantic-gap remains a problem here too.

We describe briefly some prior work in the field of automatic annotation. Barnard et al. [12] presented a scheme to link segmented image regions with



words, the results depending heavily on the quality of the segmentation. Julia Vogel [6] assigned semantically meaningful labels to local image regions formed by dividing the image into a rectangular grid. Li and Wang [11] gave a statistical modeling approach using a 2-D Multiresolution Hidden Markov Model for each keyword and choosing the keywords with higher likelihood values. Cusano et al. [13] use a multi-class SVM for annotation though their scheme can hardly be judged due to their very small vocabulary consisting of seven keywords.

In this paper we describe our annotation methodology which consists of a feature extraction, feature weighting, model evaluation and a keyword assignment routine. Note that we sometimes use the terms feature weighting and feature selection interchangeably, as once the system has given a weight to each feature, they can always be ranked to select the ones with the higher weights.

We describe briefly the outline of this paper. We first give a description of the visual features used, then present our feature weighting algorithm. Later we give a description of our model based on the one-class SVM, and present the results of the experiments. We conclude with a discussion and an outlook for possible improvements and future work.

## 2 Features

To demonstrate the effectiveness of the feature weighting and model evaluation modules, we use a small set of simple visual features comprising of the following:

**Colour Features:** Colour features are widely used for image representation because of their simplicity and effectiveness. We use color moments calculated on HSV images. For each of the three layers we compute the layer mean, layer variance and layer skewness respectively. This yields a 9-dimensional vector. Since this does not incorporate any interlayer information, we calculate three new layers SV, VH and HS non-linearly by point-wise multiplication of pairs from original layers and calculate the same 3 moments also for the new layers. The final 18-dimensional vector outperformed a 512-bin 3D Joint Colour Histogram in CBIR tests that we performed.

**Texture Features:** Texture features can describe many visual properties that are perceived by human beings, such as coarseness, contrast etc. [4]. We use the Discrete Wavelet Transformation (DWT) for calculating texture features. The original image is recursively subjected to the DWT using the Haar (db1) wavelet. Each decomposition yields 4 subimages which are the low-pass filtered image and the wavelets in three orientations: horizontal, vertical and diagonal. We perform 4 level of decompositions and for the orientation subimages we use the entropy  $(-\sum_{i=1}^L H(i) \cdot \log(H(i)))$ , with  $\mathbf{H} \in \mathbb{R}^L$  being the normalized intensity histogram of the subimage) as the feature, thus resulting in a 12-dimensional vector.

**Edge Features:** Shape features are particularly effective when image background is uncluttered and the object contour dominates. We use the edge-orientation histogram [8] which we compute directly on gray-scale images by first calculating the gradient at each point. For all points where the gradient magnitude

exceeds a certain threshold, the gradient direction is correspondingly binned in the histogram. We use an 18-bin histogram which yields bins of size 20 degrees each.

The final feature vector is a concatenation of the above three vectors and has a dimensionality of 48.

### 3 Feature Weighting

A large number of feature selection or feature weighting methods have been proposed in the machine learning literature. The interested reader can refer to [7] for an overview of some of the popular alternatives. The main distinction is between the so called *Filter* methods, which compute a ranking for the features without taking the inducer (classifier) into account, and the *Wrapper* methods, which search in the set of subsets of features for the optimum subset for the specific inducer.

We propose a feature weighting method suitable for the image annotation problem. Image annotation with keywords can be interpreted as a classification problem but with two distinct characteristics: a) The number of classes (keywords) can be very large, and b) An image object can belong to multiple classes simultaneously (in other words, an image is usually annotated with multiple keywords). Thus, traditional feature weighting methods for multi-class classification are not only overloaded with the high number of classes, but would also give incorrect weights because of the overlap between the classes.

Our final aim is to learn a model for each class (keyword) based on a few training images. If we consider the training data for all the classes collectively, the properties of the ensemble become evident: the classes overlap, data belonging to the positive class (the class in question) is limited, but the data belonging to the negative classes is huge and spread around the feature space. Thus a multi-class classifier or a feature selection method based on it would not easily find decision boundaries or relevant features. We show that it is indeed possible to weight the features effectively for each class, taking into account the general distribution of the features. Let us start with a short data terminology. Let the training samples belonging to the positive class be given through

$$\mathbf{x}_1, \dots, \mathbf{x}_l \in \mathbb{R}^n$$

and the training examples in all the negative classes through

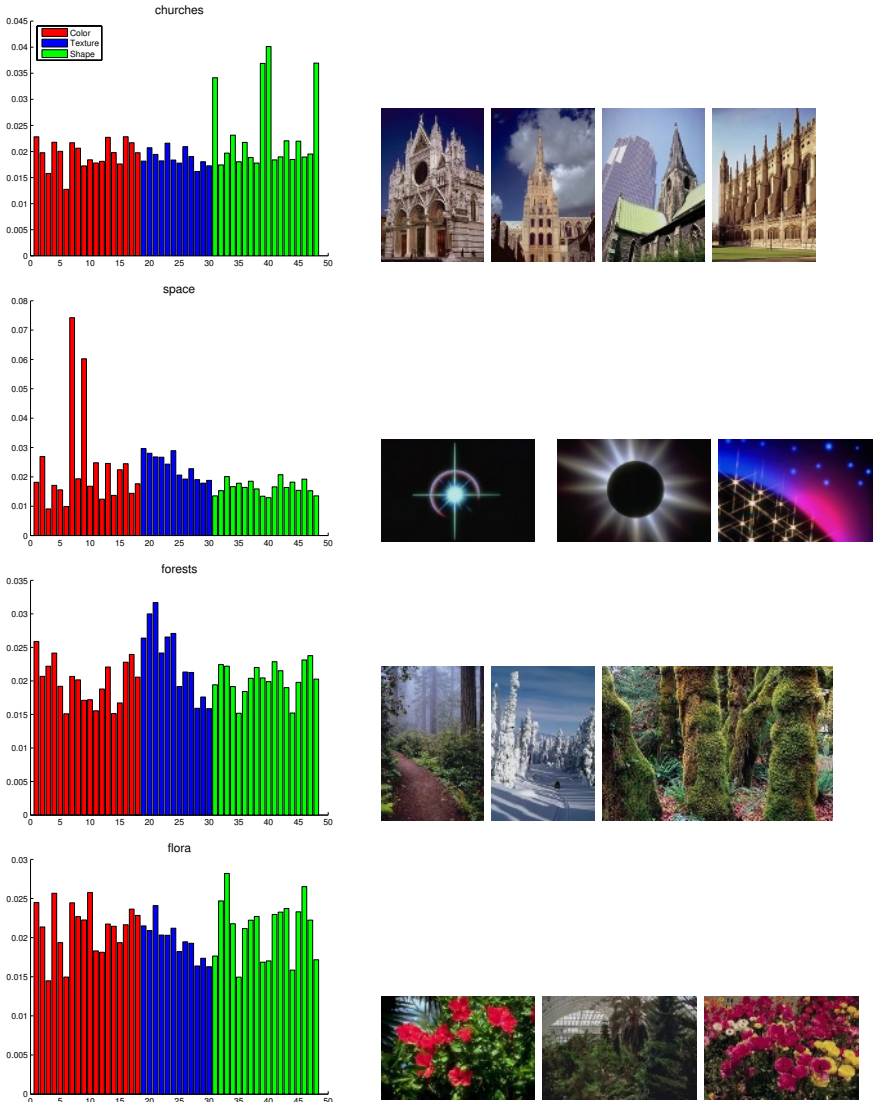
$$\mathbf{x}_{l+1}, \dots, \mathbf{x}_{l+m} \in \mathbb{R}^n$$

with  $m \gg l$ . Furthermore, we represent the  $i$ -th feature vector through the notation

$$\mathbf{x}_i = [x_i^{(1)} x_i^{(2)} \dots x_i^{(n)}]$$

All features are first normalised to zero mean and unit variance. Then, we estimate the distribution for each feature independently using the complete training data. We use a gaussian mixture model with three components to estimate the density,

$$p(x^{(k)}) = \sum_{j=1}^J \pi_j N(x^{(k)} | \Theta_j)$$



**Fig. 1.** Feature Weights for four sample Corel categories. Each row plots the feature weights and a few sample images from the category. The order of features in each graph is as follows 1) 18 Colour features: Mean, Variance and Skewness of Hue layer, followed by that of S, V, HS, SV and VH layers. 2) 12 Texture features: Entropy of H, V and D first level decomposition, followed by  $2^{nd}$  and  $3^{rd}$  levels 3) 18 Edge Features: Bins starting from  $0^\circ$  degrees in anti-clockwise direction with each bin having a span of  $20^\circ$ .

where  $N$  is the normal distribution with parameters  $\Theta_j = (\mu_j, \sigma_j)$ , and  $\pi_j$  is the weight of the  $j$ -th component, with  $\sum_{j=1}^J \pi_j = 1$ . The density is estimated using the expectation-maximization method.

We define the average likelihood for feature  $k$ , averaged only over the images of the positive class as

$$\text{avg}_k = \frac{\sum_{i=1}^l p(x^{(k)} = x_i^{(k)})}{l}$$

The higher the average likelihood is, the more similar this feature is between the positive and the negative classes and therefore less discriminative. Thus, we define the weight for the  $k$ -th feature as

$$w_k = 1/\text{avg}_k$$

The weights are normalized so that  $\sum_{k=1}^n w_k = 1$ . This has the effect that all models deliver optimum performance (tested through crossvalidation) for about the same model parameters. The features are then weighted with  $w_k$  before fed to the model computation routine (each model gets its own sets of weights). We show now that the weighting scheme is effective and the weights can in fact even be directly interpreted for our features. To do this, we plot in Fig. 1 the calculated weights for the 48 features for 4 corel categories: **churches**, **space1**, **forests** and **flora**. The training data consisted of 40 images each in the positive class and the complete Corel collection of 60,000 images as the negative (Note that it is immaterial here if the positive images are considered for determining the gaussian mixture distribution or not, as we have a very large number of samples available from the stochastic process). The sequence of the 48 features is explained in the figure caption.

For the **churches** category, the maximum weight went to the edge features corresponding to the directions  $0^\circ$  and  $180^\circ$ , i.e., the discriminative vertical edges present in churches and other buildings (most images in the category were taken upright). For the **space1** category, the most discriminative feature the system found was the 7<sup>th</sup> feature, which is the mean of the brightness ( $V$ ) component of the image (the images in the category are mostly dark). For the **forests** category, texture features get more weight, as does the hue component of the colour features. We however did find some categories where the weights were somewhat counter-intuitive or difficult to interpret manually. An example is the category **flora** in part d).

## 4 Model Computation

We assume that the presence or absence of a keyword in an image can be tested independently of other keywords. Though it is not necessarily true, it is a reasonable assumption to keep the complexity of the overall system in check. Otherwise, the system would need access to the conditional probabilities of keywords given the presence of other keywords.

We propose a slightly modified one-class Support Vector Machine (SVM) as our model. One-Class SVM were introduced by Schölkopf et al. [9]. One-Class

SVMs are binary functions which capture regions in the input space where the probability density lies (i.e. its support). We train a one-class SVM for every keyword with the aim to determine subspaces in the feature space where most of the data for that keyword is present.

One-Class SVMs are the solution to the following optimization problem: Find a hypersphere in  $\mathbb{R}^n$  which contains most of the training data and is at the same time as small as possible. This can be written in primal form as:

$$\min_{R \in \mathcal{R}, \zeta \in \mathcal{R}^l, \mathbf{c} \in \mathcal{F}} R^2 + \frac{1}{\nu l} \sum_i \zeta_i$$

subject to

$$\|\phi(\mathbf{x}_i) - \mathbf{c}\|^2 \leq R^2 + \zeta_i, \quad \zeta_i \geq 0, \quad i = 1, \dots, l$$

$\phi(\mathbf{x}_i)$  is the  $i$ -th vector transformed to another (possibly higher-dimensional) space using the mapping  $\phi$ .  $\mathbf{c}$  is the center and  $R$  the radius of the hypersphere in the transformed space. With the kernel trick [10] it is possible to work in the transformed space without ever calculating the map  $\phi(\mathbf{x}_i)$  explicitly. This can be achieved by defining a kernel function  $k(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$  as the algorithm needs access only to scalar products between vectors, and not to the actual vectors themselves.

The tradeoff between the radius of the hypersphere and the number of outliers can be controlled by the single parameter  $\nu \in (0, 1)$ . Using Lagrange multipliers, the above can be written in the dual form as:

$$\min_{\alpha} \sum_{i,j} \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) - \sum_i \alpha_i k(\mathbf{x}_i, \mathbf{x}_i)$$

subject to

$$0 \leq \alpha_i \leq \frac{1}{\nu l}, \quad \sum_i \alpha_i = 1$$

The optimal  $\alpha$ 's can be computed with the help of QP optimization algorithms. The decision function then is of the form

$$f(\mathbf{x}) = \text{sign}(R^2 - \sum_{i,j} \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) + 2 \sum_i \alpha_i k(\mathbf{x}_i, \mathbf{x}) - k(\mathbf{x}, \mathbf{x}))$$

This function returns positive for points inside this hypersphere and negative outside (note that although we use the term hypersphere the actual decision boundary in the original space can be varied by choosing different kernel functions. We use a gaussian kernel  $k(\mathbf{x}, \mathbf{y}) = \exp(-\gamma \|\mathbf{x} - \mathbf{y}\|^2)$ , with  $\gamma$  and  $\nu$  determined empirically through cross-validation). Since we need a rank for each keyword in order to annotate the image, we leave out the **sign** function, so that the results can be sorted on the basis of their “positiveness”. Furthermore, it was found that the results are biased towards keywords whose training images are very dissimilar to each other, i.e., the models for which  $R^2$  term is high.

Compact models are penalised, and therefore we use the following function instead for model evaluation:

$$g(\mathbf{x}) = \frac{R^2 - \sum_{i,j} \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) + 2 \sum_i \alpha_i k(\mathbf{x}_i, \mathbf{x}) - k(\mathbf{x}, \mathbf{x})}{R^2}$$

which can be interpreted as the normalized distance from the model boundary in the transformed space.

## 5 Experiments and Discussion

We perform our experiments similar to the ALIP system [11] to facilitate an objective comparison. The Corel Database with 600 categories is used. Each category is manually labelled<sup>1</sup> with few descriptive keywords (typically 3 to 5). Each category consists of 100 colour images of size  $384 \times 256$ , out of which we select 40 images randomly as training images. Normally for image annotation we would be training a model for every annotation keyword, and would annotate a query image with the keywords whose models evaluate the query image most favourably. For this experiment however, we learn a model for every Corel *category* instead of each *annotation keyword*. Then, for the best  $k$  category matches (we experiment with  $k = \{5, 8, 11, 14\}$ ), the category keywords are combined and the keywords least likely to have appeared by chance are taken for annotation, as in [11]. This scheme favours infrequent words like `waterfall` and `asian` over common ones like `landscape` and `people`.

To have an estimate of the discriminative performance of the system, we perform a classification task with the 600 categories. The system attains an accuracy of 11.3 % as compared to 11.88 % that of ALIP. However, as also pointed out in [11], many of the categories overlap (e.g. `Africa` and `Kenya`) and it is not clear how much can be read from this classification performance. Furthermore, we found that although the best category match was incorrect in the sense of the category ground truth, it was often meaningful with regard to the query image. We provide some annotation examples in Table 1.







For a more controlled test, we take 10 distinct Corel Categories, namely `Africa`, `beach`, `buildings`, `buses`, `dinosaurs`, `elephants`, `flowers`, `horses`, `mountains` and `food`. The confusion matrix for this task is shown in Table 2. Overall, the system attains classification accuracy of 67.8% as compared to 63.6% attained in ALIP.

**Computation Time:** All experiments were performed on an Intel Pentium IV 2.80 GHz single-CPU machine running Debian Linux. Calculation of image features takes about 1.2 seconds per image. Model computation with 40 training vectors per model takes only about 20 msec per model. A new query image needs about 4 seconds to be fully annotated (this includes computation time for feature extraction, evaluation of 600 models, and decision on unlikely keywords), as compared to 20 minutes for the HMM-based approach in ALIP. This makes

---

<sup>1</sup> We thank James Wang for making the category annotation available for comparison.

**Table 1.** Sample annotation results from the system. The query images are taken from the Corel Collection but did not belong to the training set. It can be observed that while the original category was sometimes not found, it was due to the fact that the categories often overlapped, as the top matches do indeed contain very similar categories, leading to robust annotation results.

Query Image	Original Category	Top 8 Matches	Final Annotation
	africa	wildlife_rare, architect, shells, dogs, mammals, newzealand, 197000, pastoral	grass, animal, dog, rareanimal, shell, mammal, NewZealand, pastoral
	wl_ocean	plants, green, foliage, can_park, US_garden, flora, texture13, flower2	plant, flower, green, foliage, leaf, flora
	189000	tribal, 239000, thailand, 189000, groups, perenial, indonesia, work	people, cloth, guard, face, life, tribal
	holland	rural_UK, forest, zionpark, flowerbeds, plants, forests, perenial, flower2	tree, forest, flower, ruralEngland, Zion, flowerbed, perenial
	lizard1	microimg, design1, textures, texture1, skins, texture7, texture9, food2	texture, natural, microimage
	yosemite	canyon_park, isles2, US_parks, alaska, 126000, rural_UK, gardens, cal_sea	Alaska, mountain, park, landscape, garden, house, California

**Table 2.** Confusion Matrix for the 10-category classification task

%	Africa	beach	buildings	buses	dnsrs	elephants	flowers	horses	mnts	food
Africa	<b>66</b>	6	12	0	0	2	2	2	6	4
beach	16	<b>32</b>	28	0	0	8	2	2	6	6
buildings	6	6	<b>76</b>	2	0	0	2	0	6	2
buses	0	0	30	<b>64</b>	0	0	0	6	0	0
dinosaurs	0	0	2	0	<b>94</b>	0	0	0	0	4
elephants	28	0	0	0	0	<b>50</b>	0	8	12	2
flowers	10	0	4	0	0	0	<b>78</b>	0	4	4
horses	6	2	6	0	0	2	2	<b>72</b>	10	0
mountains	4	4	10	0	0	0	6	0	<b>70</b>	6
food	0	2	14	0	2	0	2	0	4	<b>76</b>

our system faster by a factor of 300 (or 100 taking the clock speed of the ALIP system into account). The system scales linearly with the number of models.

## 6 Conclusion and Future Outlook

A feature weighting method and a modelling scheme based on the one-class SVM for automatic image annotation was presented in this paper. It is clear that the power of the overall system is heavily dependant on the discriminative power of the used features. Thus, complex features should in general be expected to lead to a performance improvement. Local features extracted around interest points, e.g. [14], have recently given excellent results in the field of object recognition and could be directly plugged into the system (at least the methods which can return a single consolidated feature vector per image, instead of a bag of vectors).

It was shown that the modelling scheme scales well to larger number of keywords, both in terms of annotation results quality as well as the speed of execution. The system ran orders of magnitude faster than a MHMM-based scheme while giving comparable or better results. The effectiveness of the feature weighting was also demonstrated as the small number of visual features used lent themselves to direct interpretation.

A simplified view of the linguistic component of the annotation system was taken, as it lies outside the scope of this work. Also, currently the system does not check for mutually exclusive keywords or other inconsistencies, and ends up annotating the same image with combinations like *sunrise* and *sunset*, or with *England* and *Finland*. This can however be taken care of automatically to an extent by extracting conditional probabilities of keywords given the presence or absence of other keywords, given sufficient training data.

**Acknowledgements.** This work was supported by the German Ministry for Education and Research (BMBF) through grant FKZ 01IRB02B and by the Muscle Network of Excellence, through contract no. 507752.



## References

1. Smeulders et. al.. Content-based image retrieval at the end of the early years, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, no. 12, pp. 1349-1380. Dec. 2000.
2. Y Rui, TS Huang, M Ortega, S Mehrotra. Relevance Feedback: A Power Tool for Interactive Content-Based Image Retrieval, *IEEE transactions on circuits and systems for video technology*, vol. 8, no. 5, september 1998.
3. C Meilhac, C Nastar. Relevance Feedback and Category Search in Image Databases, *ICMCS*, Vol. 1, 1999.
4. H. Tamura, S. Mori, T. Yamawaki. Texture features corresponding to visual perception, *IEEE Trans. Systems Man Cybernet SMC8 (6)* (1978) 00.
5. T Westerveld. Image retrieval: Content versus context, *Content-Based Multimedia Information Access RIAO* 2000.
6. Julia Vogel. Semantic Scene Modeling and Retrieval, PhD Thesis. Zurich, October 2004.
7. Ron Kohavi and George H. John. Wrappers for Feature Subset Selection, *Artificial Intelligence* 97, 1-2, Pages 273-324, 1997.
8. A. Vailaya, A. K. Jain and H.-J. Zhang. On Image Classification: City Images vs. Landscapes , *Pattern Recognition*, vol. 31, pp 1921-1936, December, 1998.
9. Schölkopf, B., J.C. Platt, J. Shawe-Taylor, A.J. Smola and R.C. Williamson. Estimating the support of a high-dimensional distribution. Technical report No.(87) Microsoft Research (1999)
10. V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, N.Y., 1995
11. Jia Li and James Z. Wang. Automatic Linguistic Indexing of Pictures by a Statistical Modeling Approach, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 9, pp. 1075-1088, 2003.
12. Kobus Barnard et al., Matching Words and Pictures, *Journal of Machine Learning Research*, Vol 3, pp 1107-1135. 2003
13. C. Cusano, G. Ciocca, R. Schettini. Image annotation using SVM, *Proceedings of Internet imaging V*, Vol. SPIE 5304, pp. 330-338, 2004.
14. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *IJCV* 60 (2004) 91-110

# Image Database Navigation on a Hierarchical MDS Grid

Gerald Schaefer<sup>1,2</sup> and Simon Ruszala<sup>2</sup>

<sup>1</sup> School of Engineering and Applied Science, Aston University, UK

<sup>2</sup> School of Computing and Informatics, Nottingham Trent University, UK

**Abstract.** Due to the ever growing size of image collections with commercial image databases in excess of 1,000,000 images, efficient and effective ways of browsing and navigation through them as well as locating and searching desired images are in high demand. An interesting approach of providing a navigational tool for image databases has been the application of multidimensional scaling (MDS) where thumbnails of images are projected onto a 2-dimensional space in such a way that the original similarities between images are best preserved. Unfortunately, MDS is both computationally expensive and is only of limited use for large image sets as images are occluding each other while at the same time certain parts of the projection space are not utilised. In this paper we provide an MDS based image database navigation approach that does not suffer from these disadvantages. Based on an initial MDS calculation images are placed on a regular grid which avoids any overlapping effects. Large image datasets are handled through a clustering technique which allows browsing in a hierarchical manner.

## 1 Introduction

Following the enormous increase in use and affordability of digital imaging equipment for both personal and professional use, currently there exists a large demand for ways of storing and exploring these image collections. With the size of image databases ranging from the average home user owning around 1,000 images to companies with databases in excess of 1,000,000 images, efficient and effective ways of browsing through them and locating and searching for desired images are highly sought after. Common tools display images in a 1-dimensional linear format where only a limited number of thumbnail images are visible on screen at any one time, thus requiring the user to search back and forth through thumbnail pages to view all images. Obviously, this constitutes a time consuming, impractical and exhaustive way of searching images, especially in larger catalogues. Furthermore, the order in which the pictures are displayed is based on attributes like file names and does not reflect the actual image contents and hence cannot be used to speed up the search.

In order to address these issues recently several approaches have been introduced which provide a more intuitive interface [10]. The basic idea behind most of these is to place images which are visually similar, as established through the

calculation of image similarity metrics based on features derived from image content, also close to each other on the visualisation screen, a principle that has been shown to decrease the time it takes to localise images [8]. One of the first - and still most popular - approaches was the application of multidimensional scaling (MDS) [3] used to project images being represented by high dimensional feature vectors to a 2-dimensional visualisation plane [9]. In the PicSOM system [4] tree-structured self organising maps are employed to provide both image browsing and retrieval capabilities. In [2] a hierarchical tree is employed to cluster images of similar concepts while image database navigation on a sphere was proposed in [12]. The application of virtual reality ideas and equipment to provide the user with an interactive browsing experience was introduced in [7].

While the application of MDS provides an intuitive and powerful tool for browsing image collections it has several drawbacks. As an iterative technique that proceeds on the complete dataset it is fairly computationally intensive, furthermore zooming into an area of interest usually initiates re-computation of the positions of the images that fall within the zoomed part. While MDS is useful for small image collection, for large image databases it provides a poor representation as many images are occluded, either fully or partially, by other images with similar feature vectors. In addition, empty spaces are common in areas where no images fall, creating an unbalanced representation on screen. In this paper we introduce an image database navigation method that addresses these issues. Based on MDS a visualisation of an image collection is created, yet in contrast to the previous approach this is being done in a hierarchical manner which can cope also with large image datasets and also has the advantage that all levels of the hierarchy can be pre-computed, thus allowing real-time browsing of the image database. In addition, images are laid out on a regular grid structure which avoids any unwanted overlapping effect between images. Furthermore, the visualisation space is better utilised by branching out images into otherwise unoccupied parts of the screen. The proposed method hence provides an effective, intuitive and efficient interface for image database navigation as is demonstrated on two medium sized image collections.

The rest of the paper is organised as follows: Section 2 describes in more detail multidimensional scaling and its application to image database navigation. Our proposed method is introduced in Section 3 and some experimental results are given in 4. Section 5 concludes the paper.

## 2 Multidimensional Scaling for Image Database Navigation

Rubner *et al.* were one of the first to suggest more intuitive interfaces for image database navigation [9]. They proposed the application of multidimensional scaling (MDS) [3] to calculate the locations of image thumbnails and displayed them in a global 2-dimensional view on a single screen. Using this method all images in a database are (initially) shown simultaneously; their locations are dependent on their visual similarity (based on features such as colour, texture

or shape descriptors) compared to all other images features in the database. The user is then able to zoom into an area of interest and hence to browse the image collection in an intuitive top-down manner.

MDS expresses the similarities between different objects in a small number of dimensions, allowing for a complex set of inter-relationships to be summarised in a single figure. MDS can be used to analyse any kind of distance or similarity/dissimilarity matrix created from a particular dataset. In general, there are two types of multidimensional scaling methods: metric and non-metric MDS. In metric MDS the distances between the data items are given and a configuration of points that would give rise to the distances is sought. This perfect reproduction of distances is not always possible or wanted, in which case non-metric MDS can be applied. In non-metric MDS the calculations between rank orders of similarity Euclidean distances and rank orders in the original space are computed to produce a set of metric co-ordinates which most closely approximate their non-metric distances.

First a distance matrix which contains all pairwise distances between the images in the databases is calculated. These distances  $d_{i,j}$  are typically based on similarity measures based on image features such as colour or texture distributions derived from the image content. The distances and Euclidean distances  $\hat{d}_{i,j}$  in the visualisation space are calculated and compared using Kruskal's stress formula [3]

$$\text{STRESS} = \frac{\sum_{i,j} (\hat{d}_{i,j} - d_{i,j})}{\sum_{i,j} d_{i,j}^2} \quad (1)$$

which expresses the difference between the distances  $d$  and the Euclidean values  $\hat{d}$  between all images. The aim of non-metric MDS is to assign locations to the input data so that the overall stress is minimal.

Typically an initial configuration is found through principal components analysis (PCA). While the degree of goodness-of-fit after this is in general fairly high it is not optimal. To move towards a better solution the locations of the points are updated in such a way as to reduce the overall stress. If for instance the distance between two specific samples has been overestimated it will be reduced to correct this deviation. It is clear that this modification will have implications for all other distances calculated. Therefore, the updating of the co-ordinates and the recalculation of the stress is being performed in an iterative way where during each iteration the positions are slightly changed until the whole configuration is stable and the algorithm has converged into a minimum where the distances between the projected samples correspond accurately to the original distances. Several termination conditions can be applied such as an acceptable degree of goodness-of-fit, a predefined maximal number of iterations or a threshold for the overall changes in the configuration. Once the calculation is terminated thumbnails of the images can then be plotted at the calculated co-ordinates on screen.

While the application of MDS provides an intuitive and powerful tool for browsing image collections and constitutes the best possible match for projecting images onto a visualisation plane it has several drawbacks. As it is based on an iterative convergence operation based on the complete image dataset it is fairly

computationally intensive with a computational complexity of  $O(kN^2)$  where  $k$  is the number of iterations required for convergence. While this can be at least partially avoided through off-line calculation (provided the database does not change too frequently), when zooming into a certain part during the interaction with the user, MDS is usually being re-applied to the images that fall within the zoomed area hence adding to the computation overhead. Alternatively faster methods such as PCA or Fastmap [1] can be applied; however they suffer from a lack of accuracy, in particular when the underlying metric is not Euclidean (or indeed, is not a metric at all) as is often the case with similarity measures employed in image retrieval tasks [11].



**Fig. 1.** MDS display of about 1400 images

Figure 1 shows an MDS plot of about 1400 images. As can be seen, MDS is not very well suited for large or even medium-sized image collections as images are being either totally or partially occluded by other images with similar feature vectors; obviously the more images in the dataset the higher the probability of occlusion/overlapping. On the other hand, areas of the visualisation space in which no images fall remain empty and hence create an unbalanced representation.

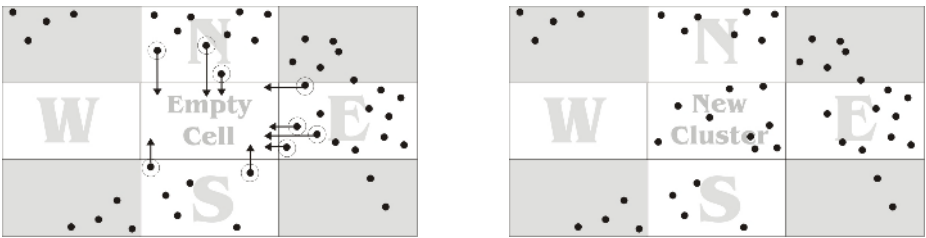
### 3 Image Database Navigation on a Hierarchical MDS Grid

In this paper we propose an image database visualisation and navigation approach based on multidimensional scaling yet without the disadvantages that have been highlighted above. That is, we address the problems associated with MDS browsing regarding occlusion and overlapping of images, unused visualisation space and

computational overheads when zooming into an area of interest. In addition, our system provides an intuitive interface also for large image collections.

### 3.1 Grid Layout

In the standard MDS visualisation the image thumbnails are placed so as to minimise the distortions in terms of distances in the projected space. From there it follows that out of necessity images will partially overlap with each other. Yet this overlapping will have a negative impact on the browsing experience [8]. An approach to minimise these effects has been proposed in [5] where images are slightly moved as a result of a local optimisation problem, yet this method provides only a partial solution to the problem. In [8] a user study was carried out which compared image visualisation models where images overlap with each other as is the case in a typical MDS layout with those where images are placed on a regular lattice without any overlapping. The results demonstrated that users largely prefer the latter as overlap adds to confusion between images and that hence a visualisation that avoids overlapping will lead to faster retrieval times. In our approach we therefore adopt these findings and constrain images to be placed on a regular grid structure where images do not overlap each other. That is, we carry out the initial MDS calculation as detailed in Section 2 but then map each images to its closest grid cell. The size of the grid structure is typically set between 10x10 and 20x20 depending monitor size and resolution. Clearly, and in particular for larger image sets, this will mean that more than one image can be mapped to a particular cell; in Section 3.3 we will describe how we are handling this case in an efficient and intuitive way through the employment of a hierarchical structure.



**Fig. 2.** Images from neighbouring cells being allocated to previously empty cell

### 3.2 Filling Empty Cells

While snapping images to a grid lattice prevents any overlapping effects, in essence it provides a "quantised" form of an MDS display. Thus, it still suffers from the relatively unbalanced view that is usually generated where certain areas of the visualisation space are not filled, which is in particular often the case for smaller image collections. To address this problem and to provide a more

uniformly inhabited browsing screen, local search strategies are employed which move images across grid boundaries to previously unoccupied cells. First the positions of all empty cells are retrieved. For each of these cells the 4-neighbourhood is then inspected. If 3 or 4 of the neighbours are occupied a relative percentage of those images closest to the borders is moved across the border to fill the previously empty cell as illustrated in Figure 2.

Performing this operation will usually fill some but not all of the empty cells. However, repeating the process based on the newly generated layout will in turn fill more cells. Hence the process is repeated a few (usually 3 or 4) times. Not all empty cells will have been assigned after that but then this is not desired as it would mean images being positioned too far from their original co-ordinates which in turn would distort the overall premise that images that are visually close should remain close on the browsing display.

### 3.3 Hierarchical Browsing

As mentioned above classical MDS displays will provide only limited usability when being applied to large but also to even medium-sized databases of a few thousand images. The reason for this is that due to the limited space on the visualisation plane images not only overlap each other partially but many images do not appear at all due to occlusion and hence only a partial view of the database is provided as can be clearly observed from Figure 1. Zooming in provides only a partial solution, in particular if there are many images with similar image features. Furthermore, a zooming operation usually re-applies MDS on the selected images which, although it tends to spread the images more evenly, also constitutes a serious computational overhead. As the zoomed-in area is specified by the user interactively and is hence not known a priori this computation cannot be performed off-line.

We employ a hierarchical tree structure to address both the navigation through large image collections and to eliminate the need for further computations. Hierarchical browsing environments such as the one described in [2] have been shown to provide an effective and efficient way of moving through large image datasets. In our approach we make direct use of the grid mapping introduced above to build a hierarchical tree based on clustering images. The resolution of the grid layout (e.g. 10x10 cells) directly determines the maximal number of clusters present at a given level (which will only be met if all cells are filled). The grid cells (after applying the filling strategy explained in Section 3.2) also determine which images fall into which clusters. What remains to be performed is the selection of a representative image to be displayed on the visualisation grid. To do this we simply select the centroid image  $I_c$ , that is the image for which the cumulative distance to all other images in the cluster

$$D_i = \sum_{j=1}^N d(I_i, I_j) \quad (2)$$

where  $I_i$  is the  $i$ -th of  $N$  images in the cluster and  $d(.,.)$  denotes the distance between two images, is minimal, i.e.

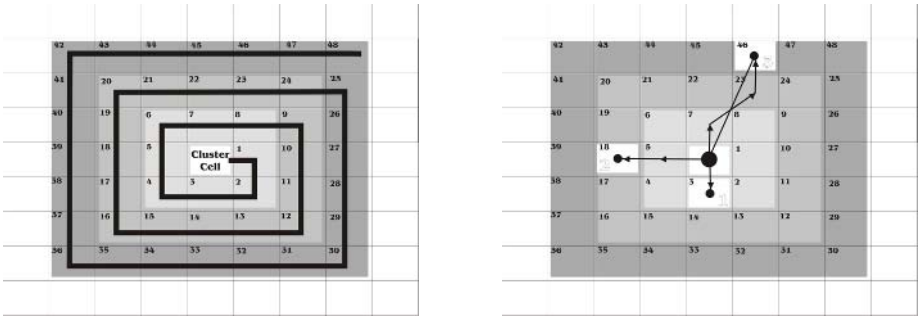


Fig. 3. Spread strategy for filled cells at tree nodes

$$D_c < D_i, \forall i \neq c \tag{3}$$

(Note that we always keep both the original MDS and the derived co-ordinates for each image.)

This procedure is adopted at the each level of the tree hierarchy, i.e. first at the root node (the initial global display) and then for each non-empty cell again in a recursive manner, where the images of each child node are again mapped to a grid structure, until the complete tree is derived.

The resulting interface provides an intuitive way of browsing to the user who can, starting from the initial display, successively select a representative image to refine the search. That image cluster (plus, if wanted, the 8 neighbouring clusters) is then expanded in the next level of the display where the user can again select an image group to navigate further into the image collection. Even based on a small grid of 10x10 cells and a fairly conservative estimate of 50% of cells being filled on average this approach requires - on average - only 4 levels of the hierarchy to provide access to each of more than 6 million images (i.e.  $50^4 = 6250000$ ) which will suffice for even the largest images databases these days.

The grid-tree structure also provides another advantage. As the structure is fixed it can be pre-computed in completeness off-line, together with all possible grid views configuration the user can encounter<sup>1</sup> which in turn provides the user with the possibility of real-time browsing large image collection.

### 3.4 Image Spreading in Tree Cells

In the tree nodes of the cells it will commonly occur that only a few images occur most of which will be visually fairly similar. To avoid them from being mapped to the same cell and hence to trigger another tree level, a spreading algorithm is applied which displays them on the same screen once only a certain percentage of cells are filled for a cluster (we currently set this threshold to 25%).

The algorithm is based on the "place", "bump" and "double-bump" principle and is similar to the one employed in [8]. When a cluster is encountered a spiral

<sup>1</sup> These structures only contain pointers to the images and can hence be maintained in memory. Image thumbnails are then loaded upon request.





**Fig. 4.** Global grid MDS view of the UCID dataset

scan is initiated that searches for and fills empty cells close by until all images are distributed. If an empty cell is encountered on the first ring around the cell, the next image of the cluster is assigned to that cell ("place"). When an empty cell in the second ring is found it is first established which of the cells of the first ring is closest to the direct path from to the identified empty cell. The image from the thus identified cell is then moved to the empty cell whereas the next image from the cluster is placed in the cell from the first ring ("bump"). The same principle is applied to empty cells identified in the third ring with images from the first and second ring being moved ("double bump"). Both the spiral scan and the three placement strategies are illustrated in Figure 3.

## 4 Experimental Results

We tested our novel approach to image database navigation on two medium-size databases: the UCID dataset [13] which contains about 1400 images and the MPEG-7 common colour dataset [6] of about 4500 images. Unfortunately, due to space restrictions, we can only provide a "tip of the iceberg" view of the capabilities in the following figures.

A standard MDS global display of the UCID images was already shown in Figure 1. The corresponding global grid view, i.e. the initial view of our browsing approach, based on a 15x15 grid, is given in Figure 4<sup>2</sup>. As can be seen, in contrast to the standard MDS layout where many images overlap each other, here the grid structure greatly contributes to the clarity of the visualisation.

<sup>2</sup> We note that the axes here are defined differently as in Figure 1, being roughly mirrored along the vertical axis and slightly rotated. However, as the axes in MDS do not carry any meaning the two representations are equivalent.

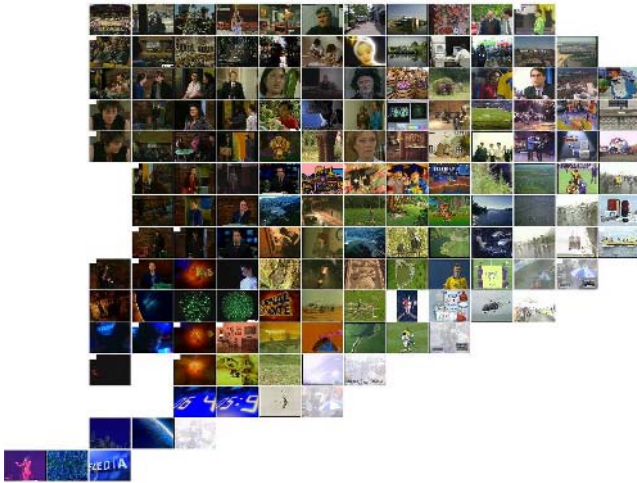


Fig. 5. Global grid MDS view of the MPEG-7 dataset

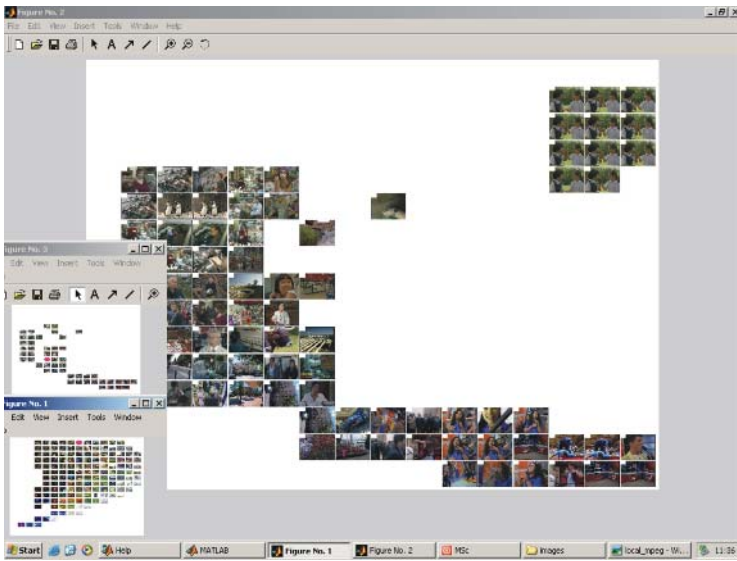


Fig. 6. Screenshot of the browsing application. User navigated to level 3 of the tree structure.

Figures 5 and 6 show the global view of the MPEG-7 database and an actual screenshot of the browsing application where the user navigated through to level 3 of the tree. To aid navigation the previous hierarchy levels are also displayed and the current position within those grid marked with the red dot. At the shown level the image spreading algorithm described in Section 3.4 has been automatically applied as originally only 5 of the cells were filled.

## 5 Conclusions

An intuitive and efficient technique to image database visualisation and navigation was presented. Based on an MDS approach, a hierarchical tree structure is generated which can be pre-computed completely and is hence able to provide image browsing functionality in real time. Overlapping and occlusion of images is achieved through the adoption of a regular grid layout paired with a hierarchical browsing functionality.

For future research we are planning to evaluate the proposed navigation tool on a real user group and compare it with other methods in the literature. We are also in the process of integrating the system with a professional image provider with a dataset in excess of 3 million images.

## References

1. C. Faloutsos and K-I. Lin. FastMap: a fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. In *ACM SIGMOD Int. Conference on Management of Data*, pages 163–174, 1995.
2. S. Krishnamachari and M. Abdel-Mottaleb. Image browsing using hierarchical clustering. In *4th IEEE Symposium on Computers and Communications*, 1999.
3. J.B. Kruskal and M. Wish. *Multidimensional scaling*. Sage Publications, 1978.
4. J. Laaksonen, M. Koskela, P. Laakso, and E. Oja. PicSOM - content-based image retrieval with self organising maps. *Pattern Recognition Letters*, 21:1197–1207, 2000.
5. B. Moghaddam, Q. Tian, N. Lesh, C. Shen, and T.S. Huang. Visualization and user-modeling for browsing personal photo libraries. *Int. Journal of Computer Vision*, 56(1-2):109–130, January 2004.
6. Moving Picture Experts Group. Description of core experiments for MPEG-7 color/texture descriptors. Technical Report ISO/IEC JTC1/SC29/WG11/ N2929, 1999.
7. M Nakazato and T.S. Huang. 3D MARS: Immersive virtual reality for content-based image retrieval. In *IEEE Int. Conference on Multimedia and Expo*, 2001.
8. K. Rodden, D. Basalaj, W. ans Sinclair, and K. Wood. Evaluating a visualisation of image similarity as a tool for image browsing. In *IEEE Symposium on Information Visualization*, pages 36–43, 1999.
9. Y. Rubner, L. Guibas, and C. Tomasi. The earth mover’s distance, multi-dimensional scaling, and color-based image retrieval. In *Image Understanding Workshop*, pages 661–668, 1997.
10. S.D. Ruszala and G. Schaefer. Visualisation models for image databases: A comparison of six approaches. In *Irish Machine Vision and Image Processing Conference*, pages 186–191, 2004.
11. Simone Santini and Ramesh Jain. Similarity measures. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 21(9):871–883, 1999.
12. G. Schaefer and S. Ruszala. Image database navigation: A globe-al approach. In *Int. Symposium on Visual Computing*, volume 3804 of *Springer Lecture Notes on Computer Science*, pages 279–286, 2005.
13. G. Schaefer and M. Stich. UCID - An Uncompressed Colour Image Database. In *Storage and Retrieval Methods and Applications for Multimedia 2004*, volume 5307 of *Proceedings of SPIE*, pages 472–480, 2004.

# Linear vs. Nonlinear Feature Combination for Saliency Computation: A Comparison with Human Vision

Nabil Ouerhani, Alexandre Bur, and Heinz Hügli

Institute of Microtechnology, University of Neuchâtel  
Rue A.-L. Breguet 2, CH-2000 Neuchâtel, Switzerland  
{nabil.ouerhani, alexandre.bur, heinz.hugli}@unine.ch

**Abstract.** In the heart of the computer model of visual attention, an interest or saliency map is derived from an input image in a process that encompasses several data combination steps. While several combination strategies are possible and the choice of a method influences the final saliency substantially, there is a real need for a performance comparison for the purpose of model improvement. This paper presents contributing work in which model performances are measured by comparing saliency maps with human eye fixations. Four combination methods are compared in experiments involving the viewing of 40 images by 20 observers. Similarity is evaluated qualitatively by visual tests and quantitatively by use of a similarity score. With similarity scores lying 100% higher, non-linear combinations outperform linear methods. The comparison with human vision thus shows the superiority of non-linear over linear combination schemes and speaks for their preferred use in computer models.

## 1 Introduction

It is generally admitted today that the human vision system makes extensive use of visual attention mechanisms in order to select a reduced set of relevant information among the huge amount of visual input gathered by the retina. By reducing the amount of data to be transferred to cortical areas responsible for higher level tasks, visual attention speeds up the vision process and contributes to its efficiency. Like in human vision, visual attention represents a fundamental mechanism for computer vision where similar speed up of the processing can be envisaged. Thus, the paradigm of computational visual attention has been widely investigated during the last two decades. Numerous computational models have been therefore reported [1], [3]. Most of them rely on the feature integration theory [4]. The saliency-based model of Koch and Ullman was first presented in [5] and gave rise to numerous software and hardware implementations [6], [7]. Further, it has been used to solve numerous issues in various fields including mobile robotics [8], [9], color image segmentation [10] and object recognition [11].

The saliency-based model of visual attention generates, for each visual cue (color, intensity, orientation, etc), a conspicuity map, i.e. a map that highlights the scene locations that differ from their surroundings according to the specific visual cue. Then, the computed maps are integrated into a unique map, the saliency map which encodes the saliency of each scene location. Depending on the scene, visual cues may

contribute differently to the final saliency and of course, some scene locations may have higher saliency values than others. Therefore, the integration process of the conspicuity maps into the saliency map should account optimally for these two aspects.

Note that the map integration process, described here for the purpose of fusing cues, is also available at earlier steps of the computational model, namely for the integration of multi-scale maps or integration of different features. Omnipresent in the model, the competitive map integration process plays an important role and deserves careful design. The question whether the map integration process is linear or non-linear, or more precisely which of the linear or non-linear model performs better in comparison to human eye movements motivated this research.

In [12] four methods are considered for performing the competitive map integration and the methods were evaluated with respect to the capability to detect reference locations, but no comparison with eye movements is performed. Specifically, the authors propose an interesting weighting method which will be considered here. Also a so-called iterative method is proposed which performs a non-linear transform of a map. Another feature integration scheme which comprises several masking mechanisms was also proposed in [18]. Leaving by side for the moment these two advanced non-linear approaches as well as other scaling like the long-term normalization proposed in [13], the present paper compares two simple linear and two simple exponential models.

The comparison of saliency maps with human eye fixations for the purpose of model evaluation has been performed previously. In [15] the authors propose the notion of chance-adjusted saliency for measuring the similarity of eye fixations and saliency. This requires the sampling of the saliency map at the points of fixations. In [17] the authors propose the reconstruction of a human saliency map or fixation map from the fixations and perform the comparison by evaluating the correlation coefficient between fixation and saliency maps. This method was also used in [18]. In the present work, the chance adjusted saliency method is used to define a similarity score.

The remainder of this paper is organized as follows. Section 2 gives a brief description of the saliency-based model of visual attention. Section 3 defines the tools used for comparing saliency and fixations. Section 4 is devoted to the selection and definition of the four map integration methods that are then evaluated by experiments described in section 5. Finally, section 6 concludes the paper.

## 2 The Saliency-Based Model of Visual Attention

The saliency-based model of visual attention was proposed by Koch and Ullman in [5]. It is based on three major principles: visual attention acts on a multi-featured input; saliency of locations is influenced by the surrounding context; the saliency of locations is represented on a scalar saliency map. Several works have dealt with the realization of this model [2], [6]. Although any number of features and cues can be considered, this paper describes the model used during in order to simplify the notation. In fact, the model generates a saliency map from 3 cues namely contrast, orientation and chromaticity and the cues stem from 7 features. The different steps of the model are detailed below.

## 2.1 Feature Maps

First, 7 features ( $j=1\dots 7$ ) are extracted from the scene by computing the so-called feature maps from an RGB color image. The features are:

- Intensity feature:  $F_1 = I = 0.3 R + 0.59 G + 0.11 B$
- Two chromatic features based on the two color opponency filters red-green and blue-yellow:  $F_2 = (R-G)/I$  and  $F_3 = (B-Y)/I$ . Note that the normalization of the opponency signals by  $I$  decouples chromaticity from intensity.
- Four local orientation features  $F_4\dots F_7$  according to the angles  $\theta \in \{0^\circ; 45^\circ; 90^\circ; 135^\circ\}$ .

## 2.2 Conspicuity Maps

In a second step, each feature map is transformed into its conspicuity map. The computation of the conspicuity maps relies on three main components:

- The multiscale approach is aimed at detecting conspicuous features of different sizes and consists in the representation of each feature  $F_j$  at multiple resolution levels ( $k=1\dots 6$ ), producing a set of images  $F_{j,k}$
- The center-surround mechanism is used to extract local activities and consists in a difference-of-Gaussians-filter DoG which applies at each resolution level and produces the multiscale maps:

$$M_{j,k} = \left| F_{j,k} * DoG \right|. \quad (1)$$

- The map integration scheme. At this level, the multiscale maps are combined, in a competitive way, into a single *feature conspicuity map*  $C_j$  in accordance with:

$$C_j = \sum_{k=1}^K N(M_{j,k}), \quad (2)$$

where  $N(\cdot)$  is a normalization function that simulates both intra-map competition and inter-map competition among the different scale maps.

In the third step, using the same competitive map integration scheme as above, the seven ( $j=1\dots 7$ ) features are then grouped, according to their nature, into the three cues intensity, color and orientation. Formally, the *cue conspicuity maps* are thus:

$$C_{\text{int}} = C_1; \quad C_{\text{orient}} = \sum_{j \in \{2,3,4,5\}} N(C_j); \quad C_{\text{chrom}} = \sum_{j \in \{6,7\}} N(C_j). \quad (3)$$

## 2.3 Saliency Map

In the final step of the attention model, the cue conspicuity maps are integrated, by using the scheme as above, into a saliency map  $S$ , which formally is:

$$S = \sum_{\text{cue} \in \{\text{int}, \text{orient}, \text{chrom}\}} N(C_{\text{cue}}). \quad (4)$$

### 3 Comparing Fixations and a Saliency Map

The idea is to design a computer model which is close to human visual attention and, here, our basic assumption is that human visual attention is tightly linked to eye movements. Thus, eye movement recording is a suitable means for studying the spatial deployment of human visual attention. More specifically, while the observer watches at the given image, the  $K$  successive fixation locations of his eyes

$$\mathbf{X}^i = (\mathbf{x}_1^i, \mathbf{x}_2^i, \mathbf{x}_3^i, \dots, \mathbf{x}_k^i, \dots, \mathbf{x}_K^i) \tag{5}$$

are recorded and then compared to the computer generated saliency map.

The degree of similarity of a set of successive fixations with the saliency map is evaluated qualitatively and quantitatively. For the qualitative comparison, the fixations are transformed in a so-called fixation map which resembles the saliency map and the similarity is evaluated by comparing them visually. For the quantitative comparison, a similarity score is used.

#### 3.1 Fixation Map

The fixation map is computed under the assumption that it is an integral of weighted point spread functions  $h(\mathbf{x})$  located at the positions of the successive fixations. It is assumed that each fixation  $\mathbf{x}_k$  gives rise to a gaussian distributed activity. The width  $\sigma$  of the gaussian was chosen to approximate the size of the fovea. A weighting of  $h(\mathbf{x})$  as a function of the fixation duration or position  $k$  in the eye trajectory was not considered. Formally, the human *fixation map* is:

$$H(x) = \frac{1}{K} \sum_{k=1}^K h(\mathbf{x}_k) \tag{6}$$

#### 3.2 Score

In order to compare a computational saliency map and human fixation patterns quantitatively, we compute a score  $s$ , similar to the chance-adjusted saliency used in [15]. The idea is to define the score as the difference of average saliency  $\bar{s}_{fix}$  obtained when sampling the saliency map  $S$  at the fixations points with respect to the average  $\bar{s}$  obtained by a random sampling of  $S$ . In addition, the score used here is normalized and thus independent of the scale of the saliency map, as argued in [16]. Formally, the score  $s$  is thus defined as:

$$s = \frac{\bar{s}_{fix} - \bar{s}}{\bar{s}} \quad , \quad \text{with} \quad \bar{s}_{fix} = \frac{1}{K} \sum_{k=1}^K S(\mathbf{x}_k) . \tag{7}$$

### 4 Four Map Integration Methods

The summation in eq. 2, 3 and 4, which is supposed to perform the competitive map integration, uses the normalization function  $N(.)$  which will now be defined.

To perform intra-map competition, and for the purpose of linear and non-linear scaling, we choose a straightforward peak to peak linear normalization and the corresponding exponential normalization as follows:

$$C' = \frac{C - C_{\min}}{C_{\max} - C_{\min}} ; \quad C'' = \left( \frac{C - C_{\min}}{C_{\max} - C_{\min}} \right)^{\gamma} . \quad (8)$$

The exponential form of this transformation promotes the higher conspicuity values and demotes the lower values; it therefore tends to suppress the lesser important values forming the background.

For the purpose of inter-map competition, most of the previous works dealing with saliency-based visual attention use a competition-based scheme for map combination [6]. We adopt the same scheme in this work and attribute a weight  $w$  to each conspicuity maps for expressing its contribution. The weight is computed from the conspicuity map itself and tends to catch the global interest of that map. We consider following weight definitions:

$$w_1 = (M - \bar{m})^2 \text{ and } w_2 = \frac{C_{\max}}{\bar{C}} . \quad (9)$$

In the first weight expression  $w_1$  which stems from [6],  $M$  is the maximum value of the normalized conspicuity map and  $\bar{m}$  is the mean value of its local maxima. This weight tends to promote maps with few dissimilar peaks and to demote maps with a lot of same peaks. In the second weight expression  $w_2$ ,  $C_{\max}$  and  $\bar{C}$  are respectively the maximum and mean values of the conspicuity map. This weight tends to promote maps with few large peaks and demote maps with a lot of similar peaks.

Considering above alternatives, we come up with the following definition of  $N(\cdot)$

$$\begin{aligned} N_{lin w_1}(C) &= w_1 \cdot C' & N_{exp w_1}(C) &= w_1 \cdot (C')^{\gamma} \\ N_{lin w_2}(C) &= w_2 \cdot C' & N_{exp w_2}(C) &= w_2 \cdot (C')^{\gamma} \end{aligned} \quad (10)$$

where  $C'$  is the peak to peak normalized conspicuity  $C$  according to eq. 8. Four map integration methods are thus defined.

## 5 Comparison Results

This section presents comparisons between the four map integration methods. The basic idea consists in comparing, for a given set of color images, the saliency maps produced by the four methods with human eye movement patterns recorded while subjects are looking at the same color images [14].

Eye movements were recorded with an infrared video-based tracking system (EyeLink™, SensoMotoric Instruments GmbH, Teltow/Berlin). This system consists of a headset with a pair of infrared cameras tracking the eyes, and a third camera monitoring the screen position in order to compensate for any head movements. The images were presented in blocks of 10. The images were presented in a dimly lit room on a 1900 CRT display with a resolution of 800x600, 24 bit color depth, and a refresh rate of 85 Hz. Every image was shown for 5 seconds, preceded by a center fixation display

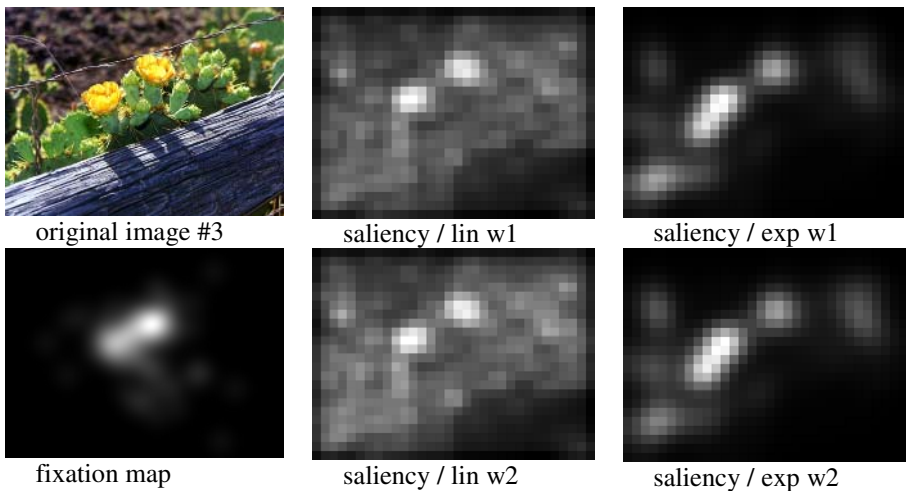


of 1.5 seconds. Image viewing was embedded in a recognition task. For every image and each subject, the measurements yielded a sequence of fixations according to eq. 5.

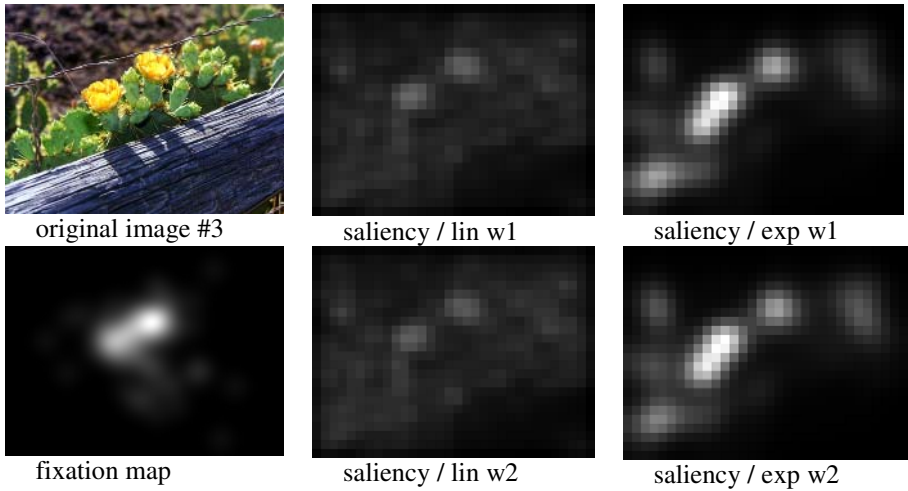
The experimental image data set consists in 40 color images of various types like natural scenes, fractals, and abstract art images. Most of the images (36) were shown to 20 subjects while the remaining were viewed by 7 subjects only. As stated above, these images were presented to the subjects for a duration of 5 seconds per image, resulting in an average of 290 fixations per image. Regarding the fixation maps, they were computed according to eq. 6 using, for a given image, all fixations from all subjects. The four map integration methods were used, the value of  $\gamma$  is set to 2.

Figure 1 provides for image #3 a visual comparison of fixation map and the saliency maps computed by the four different methods. We note only small differences between the  $w_1$  and  $w_2$  alternatives, but significant differences between the linear and non-linear methods. Comparing later methods with the fixation map, we observe good similarity at the higher intensity values, but at the lower intensity values, the linear methods provide a lot of energy where there is none in the fixation map. This illustrates the advantage of the non-linear methods, which tend to keep only the highest peaks at each map integration step and accumulate thus less background signal in comparison to linear methods.

Figure 2 provides another illustration of the same comparison. Unlike previous figure where each saliency map is individually scaled to the full intensity range for best viewing purposes, here all saliency maps are scaled to the same average intensity, as this is the way a universal comparison can be performed with the fixation map. The motivation for this is the fact that all fixation maps have a constant average by construction and that they should also be compared with saliency maps with the same constant averages. This figure illustrates even better the higher similarity of the fixation map with saliencies for non-linear methods. Note that the score definition in eq. 7 reflects quantitatively the comparison illustrated here. Another example is provided in figure 3 with image #7.

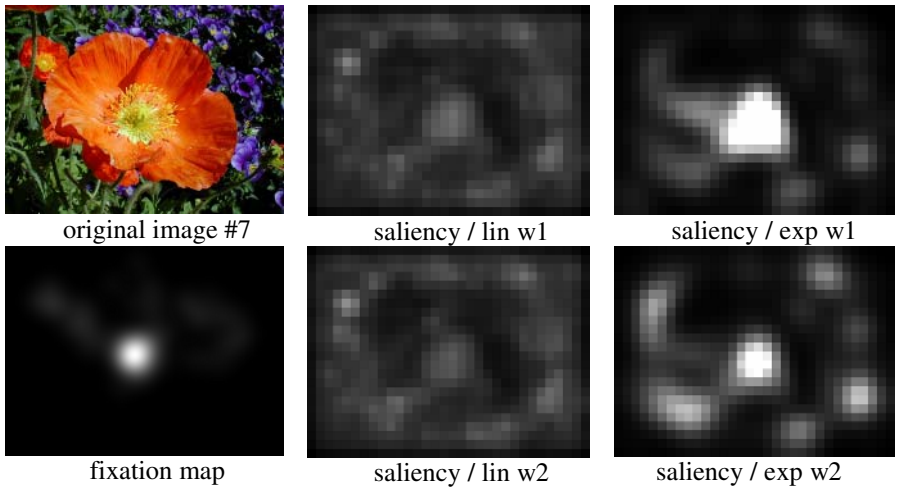


**Fig. 1.** For image #3, comparison of fixation map with saliency maps obtained by 4 different methods. Each saliency map is represented on the full scale image intensity.



**Fig. 2.** For image #3, comparison of fixation map with saliency maps obtained by 4 different methods. All saliency maps are represented with the same average intensity.

The result of the quantitative comparison is given in figures 4 and 5. Figure 4 shows the average score over all subjects obtained by each method and each individual image. More precisely, the presented values reflect the average score over the first 5 fixations, but other numbers of fixations look similar. The plot illustrates the relatively large individual variations; detailed analysis shows that the non-linear methods outperform linear methods in more than 80% of the images.



**Fig. 3.** For image #7, comparison of fixation map with saliency maps obtained by 4 different methods. All saliency maps are represented with the same average intensity.

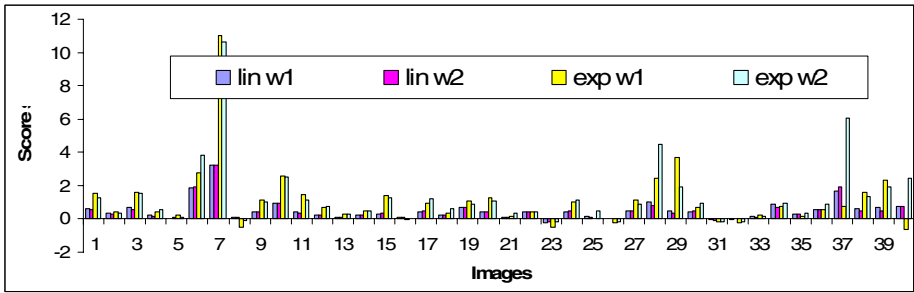


Fig. 4. Scores of the four methods for the 40 individual images

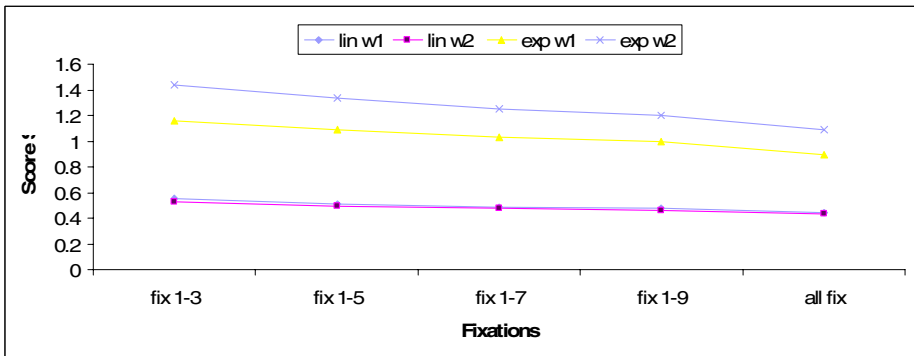


Fig. 5. Scores of the four methods for different viewing durations

Figure 5 shows the results of the comparisons of the four methods. The represented score is the average score over all subjects and all images considering a varying number of fixations. The successive values represent the first 3, 5, 7, etc fixations recorded during the viewing of a single image and illustrate the influence of viewing duration. It is noteworthy that for all cases, the model of visual attention using non-linear methods fares better in predicting where human observers foveate than the model using linear competition method. Quantitatively, the non-linear methods model yields an average score over 100% higher than the linear model. Regarding the weighting methods,  $w_2$  performs better than  $w_1$  with the non-linear method but both perform similarly with the linear methods. Here, differences are not very significant for a general preference of a method.

## 6 Conclusions

This paper presents a contribution to the design of models for visual attention computation by measuring the performance of selected methods. Performance is evaluated under the assumption that human visual attention is tightly linked to eye movements and that best similarity between the eye fixations and the saliency maps reflects also best performance. Motivated by visual comparisons of a large number of

fixation maps and corresponding saliency maps, we selected four different map integration methods and conducted a number of experiments to assess their performance. The four methods differ in their intra-map normalization scheme and inter-map weighting scheme. The normalization is either linear or exponential and there are two weighting schemes. The experiments refer to the evaluation of the collective and individual scores obtained with 40 images and from measurements of the eye movement by 20 subjects. For each image, the fixation map was visually compared to the saliency maps generated according to the different methods, and also, the relative score was computed in order to assess the performance quantitatively. The alternate weighting schemes do not differ very much in performance. The normalization methods however do, and the exponential method exhibits a score value more than twice as large as the linear method score, clearly showing the advantage of the non-linear approach. The advantage of the non-linear approach seems to be bound to the reduction of the background noise which tends to accumulate with the linear scheme. Further work is planned that will analyze this question and also consider integration schemes for evaluation.

**Acknowledgments.** The presented work was supported by the Swiss National Science Foundation under project number FN-108060.

## References

1. S. Ahmed. VISIT: An Efficient Computational Model of Human Visual Attention. PhD thesis, University of Illinois at Urbana-Champaign, 1991.
2. R. Milanese: Detecting Salient Regions in an Image: from Biological Evidence to Computer implementation. PhD thesis, Dept. Computer Science, Univ. of Geneva, Switzerland, 1993.
3. J.K. Tsotsos: Toward a computational model of visual attention. In T. V. Papathomas, C. Chubb, A. Gorea & E. Kowler, Early vision and beyond, MIT Press, pp. 207-226, 1995.
4. A.M. Treisman and G. Gelade: A feature-integration theory of attention. *Cognitive Psychology*, pp. 97-136, 1980.
5. Ch. Koch and S. Ullman: Shifts in selective visual attention: Towards the underlying neural circuitry. *Human Neurobiology*, Vol. 4, pp. 219-227, 1985.
6. L. Itti, Ch. Koch, and E. Niebur: A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, Vol. 20, No. 11, pp. 1254-1259, 1998.
7. N. Ouerhani and H. Hügli: Real-time visual attention on a massively parallel SIMD architecture. *International Journal of Real Time Imaging*, Vol. 9, No. 3, pp. 189-196, 2003.
8. J.J. Clark and N.J. Ferrier: Control of visual attention in mobile robots. *IEEE Conference on Robotics and Automation*, pp. 826-831, 1989.
9. N. Ouerhani, A. Bur, and H. Hügli: Visual attention-based robot self-localization. *European Conference on Mobile Robotics (ECMR 2005)*, September 7-10, 2005, Ancona, Italy, pp. 8-13, 2005.
10. N. Ouerhani and H. Hügli: MAPS: Multiscale attention-based presegmentation of color images. *4th International Conference on Scale-Space theories in Computer Vision*, Springer Verlag, *Lecture Notes in Computer Science (LNCS)*, Vol. 2695, pp. 537-549, 2003.

11. D. Walther, U. Rutishauser, Ch. Koch, and P. Perona: Selective visual attention enables learning and recognition of multiple objects in cluttered scenes. *Computer Vision and Image Understanding*, Vol. 100 (1-2), pp. 41-63, 2005.
12. L. Itti and Ch. Koch: A comparison of feature combination strategies for saliency- based visual attention systems. *SPIE Human Vision and Electronic Imaging IV (HVEI'99)*, Vol. 3644, pp. 373-382, 1999.
13. N. Ouerhani, T. Jost, A. Bur, and H. Hügli: Cue normalization schemes in saliency- based visual attention models. *Proc. International Cognitive Vision Workshop, Graz, Austria, 2006*.
14. T. Jost, N. Ouerhani, R. von Wartburg, R. Mueri, and H. Hügli: Assessing the contribution of color in visual attention. *International Journal of Computer Vision and Image Understanding (CVIU)*, Vol. 100, pp. 107-123, 2005.
15. D. Parkhurst, K. Law, and E. Niebur: Modeling the role of salience in the allocation of overt visual attention. *Vision Research*, Vol. 42, No. 1, pp. 107-123, 2002.
16. L. Itti. Quantitative modeling of perceptual salience at human eye position. *Visual Cognition*, in press, 2005.
17. N. Ouerhani, R. von Wartburg, H. Hügli, R.M. Müri: Empirical validation of Saliency-based model of visual attention, *Electronic Letters on Computer Vision and Image Analysis* 3(1): 13-24, 2003.
18. O. Le Meur, P. Le Callet, D. Barba, D. Thoreau: A coherent computational approach to model bottom-up visual attention. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, Vol.28, No. 5, May 2006.

# Facial Expression Modelling from Still Images Using a Single Generic 3D Head Model

Michael Hähnel, Andreas Wiratanaya, and Karl-Friedrich Kraiss

Institute of Man-Machine-Interaction, RWTH Aachen University, Germany  
{haehnel, wiratanaya, kraiss}@mmi.rwth-aachen.de

**Abstract.** We propose two approaches to facial expression modelling from single still images using a generic 3D head model without the need of large image databases (like e.g. Active Appearance Models). The first approach estimates the parameters of linear muscle models to obtain a biologically inspired model of the facial expression which may be changed intuitively afterwards. The second approach uses RBF-based interpolation to deform the head model according to the given expression. As a preprocessing stage for face recognition, this approach could achieve significantly higher recognition rates than in the un-normalized case based on the Eigenface approach, local binary patterns and a grey-scale correlation measure.

## 1 Introduction

For facial expression modelling, synthesis, and recognition many methods have been proposed [1]. One might also consider facial expression modelling as a preprocessing step in face recognition by first modelling the facial expression and then normalizing the face to a standard expression to achieve some sort of facial expression invariance which besides pose, illumination and occlusion is one of the most significant factors influencing the results of automatic face recognition [2].

In this contribution, we introduce two approaches to facial expression modelling based on a single generic 3D head model. The first approach is based on a biomechanical 3D head model which originally was proposed by Waters et al. for the synthesis of facial expressions [3,4]. The second approach uses RBF-interpolated deformation to adapt the head model mesh to the given face image. It is mainly motivated by a recent publication that suggests the use of radial basis functions (RBF) for volume oriented deformation [5]. Hence, a "pseudo-muscle" model is built using RBF interpolators. This approach is not biologically motivated but has advantages like continuity and improved processing performance.

It must be noted that both approaches presented here are not driven by a database, e.g. images are not required that contain all possible expressions. Especially, no images of the faces to be processed are required. Furthermore, all processing steps are performed on single still images.

## 2 Adapting a Generic Face Model for Facial Expression Modelling

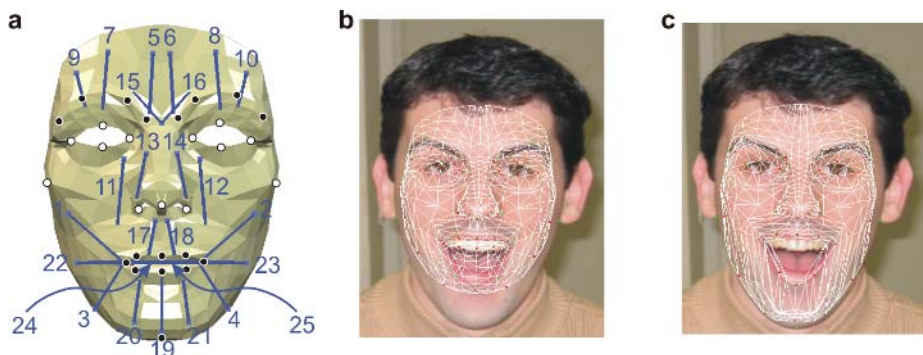
Both approaches for facial expression modelling presented here, consist of two processing steps:

1. A *coarse registration* step (pose adaption) which transforms the face model using translation, rotation and scaling so that it is optimally (in the sense of least-squares error minimization) aligned to the given feature point locations.
2. A *precise registration* step, that determines the displacements for each mesh vertex according to a few expression-variant correspondences (3D mesh vertex  $\Leftrightarrow$  2D facial feature points) which need to be defined beforehand.

First, Sect. 2.1 gives some comments on the corresponding feature points used for the registration of the head models. Sect. 2.2 then describes the (integrated coarse and precise) registration of the biomechanical model. The pseudo-muscle model is discussed in Sect. 2.3.

### 2.1 Corresponding Feature Points

Two sets of corresponding feature points are used (Fig. 1a). The mesh vertices marked with white circles are used for the coarse registration of the head model as these points are not strongly influenced by facial expression changes. An example result of the coarse registration using these feature points is shown in Fig. 1b.



**Fig. 1.** (a) feature points (black and white circular markers) for model registration and muscle model naming. (b) coarse registration by scaling, rotating and translating the generic head model. (c) precise registration using the muscle models and jaw rotation.

The feature points marked with black circles allow a vast description of the facial expression shown in the image as most changes in facial appearance can be found around the mouth and in the eye region. These points are additionally used to either estimate the muscle parameters of the biomechanical model or the RBF

interpolation functions of the pseudo-muscle model. More corresponding feature points can be defined for higher precision of the expression modelling, however, we identified the feature points mentioned here to be sufficient for "natural" expressions and making a good trade-off between precision and performance. A result of the precise registration step can be found in Fig. 1c.

## 2.2 Registration of a Biomechanical Head Model

The following sections will describe the muscle-based facial model introduced by Waters [3] and some changes and extensions we applied to it.

We removed the neck vertices from Waters original head model and used 25 muscle models which we identified to be a good set to describe most facial expressions. As already proposed by Waters, the jaw part of the head model can be rotated independently to simulate a (more extreme) opening of the mouth.

Waters et al. additionally model movements of the lips by adding a sphincter which is defined by a center point and an elliptical influence region which pulls the mouth border points towards the center. Breton et al. show however, that this limits the number of possible mouth shapes and suggest modelling the mouth by using linear muscle models only [6]. We follow this approach.

**Linear model of the face muscles.** Waters linear muscle models are defined by the tuple  $\mathcal{M} = (\mathbf{H}, \mathbf{T}, R_1, R_2, \Omega)$ , where  $\mathbf{H}$  and  $\mathbf{T}$  define the head and tail fix point of the muscle, respectively, i.e. the position of the muscle. The region in which mesh points are influenced, is defined by the angle  $\Omega$  and the two radii  $R_1$  and  $R_2$  which further divide the region into two sections  $Z_1$  and  $Z_2$ . In  $Z_1$  a maximum of displacement is modelled while in  $Z_2$  the displacement is damped towards the tail point.

The displacement  $\delta_{x_{head,i}}$  of a mesh point  $x_{head,i} \in \mathbf{R}^3$  is given by:

$$\delta_{x_{head,i}} = \cos(\alpha) \cdot k \cdot r \cdot \frac{\mathbf{x}_{head,i} - \mathbf{H}}{\|\mathbf{x}_{head,i} - \mathbf{H}\|} \quad (1)$$

where  $\alpha$  denotes the angle between the linear muscle and the connection between mesh point  $x_{head,i}$  and head  $\mathbf{H}$  of the muscle model. The muscle parameter  $k$  controls the contraction of the muscle. Its range is unique for each muscle model and was empirically determined by Waters. The damping is modelled by the function  $r$ :

$$r = \begin{cases} \cos\left(1 - \frac{\|\mathbf{H} - \mathbf{x}_{head,i}\|}{R_1}\right) & \text{if } \mathbf{x}_{head,i} \in Z_1 \\ \cos\left(\frac{\|\mathbf{H} - \mathbf{x}_{head,i}\| - R_1}{R_2 - R_1}\right) & \text{if } \mathbf{x}_{head,i} \in Z_2 \end{cases} \quad (2)$$

**Registration of the head model mesh by estimating the muscle model parameters.** The biomechanical model limits the displacement of the mesh vertices due to the inherited muscle models. Therefore, we can use the (non-linear but more precise) perspective projection that can be integrated in the registration steps as follows.

Given a set of mesh points  $x_{head,i}$  and their corresponding 2D feature points  $x_{image,i}$  in the face image, we perform the registration of the mesh and the



estimation of the muscle model parameters  $K$  by step-wise iterative minimization of a cost function  $C$  using the Levenberg-Marquardt algorithm [7]:

$$C = \sum_i \|\mathbf{P} \cdot \mathbf{M} \cdot x_{head,i}(K) - x_{image,i}\|^2 \rightarrow \min. \quad (3)$$

where  $\mathbf{M} = \mathbf{T}(t_x, t_y, t_z) \cdot \mathbf{R}_z(\alpha) \cdot \mathbf{R}_y(\beta) \cdot \mathbf{R}_x(\gamma) \cdot \mathbf{S}(s_x, s_y, s_z)$  describes the rigid transformations (translation, rotation, scaling) applied to the 3D head model and  $\mathbf{P}$  denotes the perspective projection matrix. The position of each mesh point  $x_{head,i}$  depends on the opening of the jaw and on the muscle parameters combined in the vector  $K = [k_1, \dots, k_{l+1}]^T$  where  $l$  denotes the number of used muscle models.

During *coarse registration* the parameters of  $\mathbf{M}$  are varied to align the mesh to the expression-invariant feature points (Fig. 1a). The muscle parameters  $k_i$  are kept constant. A example of the coarse registration can be seen in Fig. 1b). Then, the *precise registration* starts with an estimation of the mouth opening by rotating the mesh points of the jaw, i.e. by only varying parameter  $k_{l+1}$ . This adaptation process is only controlled by the point at the lower peak of the chin. All other muscle parameters ( $k_i, i \in \{1, \dots, l\}$ ) are kept constant. Finally, the expression-variant feature points (Fig. 1a) are used to estimate the muscle parameters  $k_i, i \in \{1, \dots, l\}$  by iteratively optimizing the muscle contraction and thus minimizing the cost function  $C$ . See Fig. 1c for an example of the precise registration using the biomechanical model.

### 2.3 Registration of the Pseudo-muscle Model

In comparison to the biomechanical model, the pseudo-muscle model does not impose any limitations on the displacements of the mesh vertices. Hence, an infinite number of solutions are possible when using perspective projection, though most of them produce "unnatural" looking results. Therefore, we first discuss an alternative method for coarse registration and afterwards outline the precise registration using RBF interpolation.

**Coarse registration of the mesh.** By using the weak-perspective projection<sup>1</sup> the coarse registration can be performed using a linear least-squares approach. Problematic are the rotation matrices because they contain non-linear trigonometric functions. For small angles  $\theta$ , i.e. close to frontal views, however, we can estimate e.g. the rotation matrix around the  $x$ -axis:

$$\mathbf{R}_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & \sin \theta \\ 0 & -\sin \theta & \cos \theta \end{bmatrix} \approx \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & \theta \\ 0 & -\theta & 1 \end{bmatrix} \approx \mathbf{I} + \theta \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} = \mathbf{I} + \theta \mathbf{R}_1$$

<sup>1</sup> As we are processing frontal faces, we substitute the perspective projection by a weak-perspective projection that assumes that the depth variation of the head is small in comparison to the distance to the camera and the projected coordinates are estimated by a suitable scaling operation [8].

The result of the registration should minimize the distance of the corresponding feature points:

$$\|\mathbf{m}_{xy}^* - \mathbf{f}\|^2 \longrightarrow MIN \quad (4)$$

where  $\mathbf{f} = (x_{image,1}, \dots, x_{image,n})$  denote the detected 2D feature points in the image,  $\mathbf{m} = (x_{head,1}, \dots, x_{head,n})$  the corresponding 3D vertices of the mesh and  $\mathbf{m}^*$  the deformed model that is obtained from the original model by considering rigid transformations with:  $\mathbf{m}^* = \mathbf{R}\mathbf{S}\mathbf{m} + \mathbf{t}$ . Here,  $\mathbf{R}$  and  $\mathbf{S}$  denote the rotation and the scaling matrices, respectively, and  $\mathbf{t}$  is the translation vector.

Using the linear approximations  $R_1, R_2, R_3$  of the rotation matrices around the  $x, y$  and  $z$  axis respectively, we get:

$$\mathbf{R}\mathbf{m} \approx (\theta_1\mathbf{R}_1 + \theta_2\mathbf{R}_2 + \theta_3\mathbf{R}_3 + \mathbf{I})\mathbf{m}$$

This leads to the deformed model  $\mathbf{m}^*$  using the overall transformation:

$$\mathbf{m}^* \approx \hat{\mathbf{m}}^* = \left(\sum_i \sigma_i \mathbf{S}_i\right)\mathbf{m} + \left(\sum_i \theta_i \mathbf{R}_i\right)\mathbf{m} + \sum_i \tau_i \mathbf{t}_i \quad (5)$$

Equation 5 can be written as  $\hat{\mathbf{m}}^* = \mathbf{A}\mathbf{s}$ , where  $\mathbf{A} \in \mathbb{R}^{3n \times 9}$  defines all transformations

$$\mathbf{A} = [\mathbf{S}_1\mathbf{m}, \mathbf{S}_2\mathbf{m}, \mathbf{S}_3\mathbf{m}, \mathbf{R}_1\mathbf{m}, \mathbf{R}_2\mathbf{m}, \mathbf{R}_3\mathbf{m}, \mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3] \quad (6)$$

and  $\mathbf{s} \in \mathbb{R}^{9 \times 1}$  the transformation parameters

$$\mathbf{s} = [\sigma_1, \sigma_2, \sigma_3, \theta_1, \theta_2, \theta_3, \tau_1, \tau_2, \tau_3]^T \quad (7)$$

Because we are using the weak-perspective projection the  $z$ -coordinate has no effect and hence scaling ( $\sigma_3$ ) and translation ( $\tau_3$ ) along the  $z$ -axis can be ignored. This leads to:

$$\min \|\mathbf{m}_{xy}^* - \mathbf{f}\|^2 \approx \min \|\hat{\mathbf{A}}\hat{\mathbf{s}} - \mathbf{f}\|^2 \quad (8)$$

which can be solved by e.g. SVD or using the pseudo-inverse:  $\hat{\mathbf{s}} = (\hat{\mathbf{A}}^T \hat{\mathbf{A}})^{-1} \hat{\mathbf{A}}^T \mathbf{f}$

**Precise registration of a 3D head model using RBF interpolation.** To precisely register the head model mesh, we apply the following global interpolation function to all mesh vertices  $\mathbf{p}$ :

$$f(\mathbf{p}) = \sum_i \mathbf{c}_i \phi_i(\mathbf{p}) + \mathbf{M}\mathbf{p} + \mathbf{t} \quad (9)$$

with the radial basis functions  $\phi_i(\mathbf{p}) = \phi(\|\mathbf{p} - \mathbf{p}_i\|)$ .

To determine the coefficients  $\mathbf{c}_i$  and the affine components  $\mathbf{M}$  and  $\mathbf{t}$ , a linear equation system needs to be solved under the interpolation constraint  $\mathbf{d}_i = f(\mathbf{p}_i)$ , where  $\mathbf{p}_i$  are the interpolation centers and  $\mathbf{d}_i$  the displacement vectors associated with each  $\mathbf{p}_i$ . Additionally, the constraints  $\sum_i \mathbf{c}_i = 0$  and  $\sum_i \mathbf{c}_i \mathbf{p}_i^T = 0$  must be considered to remove the affine components from the radial basis functions.

The displacement vectors for each correspondence (facial feature points in Fig. 1a) can be simply determined by:  $\mathbf{d}_i = \mathbf{m}_{xy,i}^* - \mathbf{x}_{image,i}$ . Hence, we get a

set of interpolation centers and displacement vectors which allow to determine the  $\mathbf{c}_i$  by solving the above mentioned equation system. By applying the RBF interpolation (Eq. 9) to the head model, a displacement vector for each mesh vertex can be estimated and hence the whole head model can be appropriately adapted to the given face image.

It further must be decided which basis function can be used. We achieved the best results using the Gaussian basis function:  $\phi(r) = e^{-cr^2}$ . The value of the parameter  $c$  depends on the size of the head model and can be determined experimentally. By using the Gaussian basis function the overall interpolation of the head model by a single basis function has some kind of local character. This property is necessary as the head model must be more strongly distorted in some parts (e.g. around the mouth) than in others (e.g. nose). This can be controlled by appropriately choosing suitable corresponding feature points.

By choosing the Gaussian kernel function, self-penetration of the mesh is unlikely because the global deformation function becomes  $C^\infty$ -continuous. This fact is important to mention, because discontinuous artefacts can appear when using the biomechanical model. In this case, the linear muscle models perform a patch-wise deformation of the mesh because they take affect in local regions that might overlap. Additionally, it might happen that several vertices are moved to the position of the muscle's head vertex when the muscle needs to be strongly "contracted". These two properties can lead to discontinuous artefacts, hence, requiring a different modelling approach, like e.g. the RBF interpolation.

### 3 Experimental Results

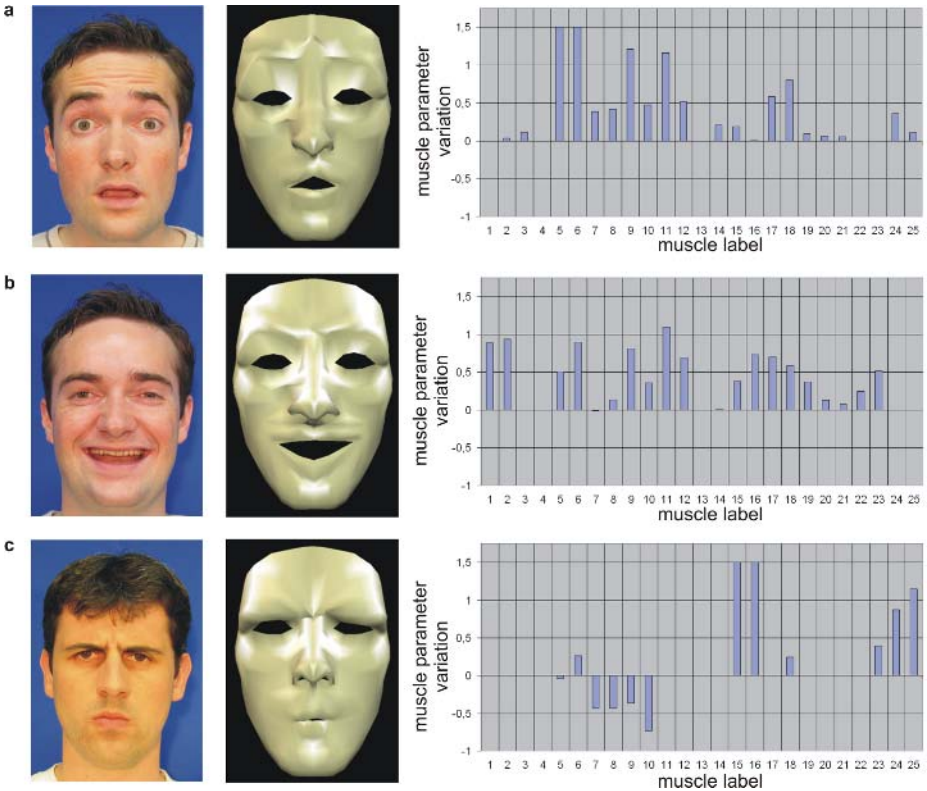
In this section, we present some results which we achieved using the two approaches proposed here. Sect. 3.2 will outline the face expression modelling results which we achieved using the biomechanical model in a qualitative manner. In Sect. 3.3 we like to show some results of our application of the RBF-interpolation based expression modelling approach, that is robust face recognition by image normalization.

#### 3.1 Face Image Databases

For our experiments we used a face image database of 17 persons each showing 6 expressions (neutral, smiling, surprise, anger, (pursed lips for a) kiss, grinning). The images were taken in two sessions with a resolution of  $800 \times 600$  pixels, i.e. the face's width was about 160 pixels in average.

#### 3.2 Facial Expression Modelling Using Linear Muscle Models

Fig. 2 shows some results which were obtained on facial images in frontal pose. On the left side, the face region of the processed images are depicted. The center column shows the head model with contracted muscles after the whole adaptation process. The diagrams on the right side show which muscles (see Fig. 1a)



**Fig. 2.** Three example results showing the ability to model facial expression variations: (a) frightened, (b) smile, (c) furious. Original image (left column), adapted head model (center column) and muscle contraction parameters (right column).

were contracted and how the contraction parameters were varied from the neutral position (zero line).

The first example shows a "frightened" expression. The typical raising of the eyebrows in this facial expression is well estimated by the adaptation especially of the muscles located on the forehead (muscles 5 – 10). Additionally, a slight adaption of the mouth shape is performed (muscles 17 – 25).

The smiling facial expression is modelled precisely as well. Mainly adaptation of the mouth shape (muscles 1 + 2 and 17 – 23) but also adaptation of the eye region (muscles 9 – 12 and 15 + 16) is performed.

A distinctive characteristic for the furious expression in the third example (Fig. 2c) is the contraction of eye brows and mouth. Both are modelled by the contraction of the muscles 7 – 10 and 15 – 16 (eye brows) and 23 – 25 (mouth), respectively.

The muscle parameter plots show that the muscle parameter vector  $K = [k_1, \dots, k_l]$  (Eq. 1) could be used to build a facial expression recognition system. This, however, was not part of this study and must be examined in future work.

### 3.3 Face Recognition Using Expression-Normalized Views

The pseudo-muscle model is obviously better suited for normalizing images for face recognition in cases when the muscle models must be extremely contracted. In this case, the patch-wise deformation property of the muscles as well as the possible contraction of multiple vertices into the head of a muscle result in a discontinuous folding of the image texture thus producing unnatural results, e.g. note the eye brows in Fig. 3a. Therefore, we only examined the pseudo-muscle model by performing face recognition experiments.

To demonstrate the impact of the proposed approach using different methods of face recognition, we compare three approaches here: the eigenface approach [9], local binary patterns (LBP) [10] and a simple grey-value comparison. In the latter case, the grey-values of the face images were simply concatenated into a feature vector and the Euclidean distance was used as distance measure.

In each test session, the images of a facial expression were trained and tested against all other facial expressions of the 17 persons in the database, i.e. only one training sample per subject and 85 test images in total were used. For comparison, the tests were performed with the original images that were only normalized by scale and rotation considering the eye locations. The results are shown in figure 3 and table 1.

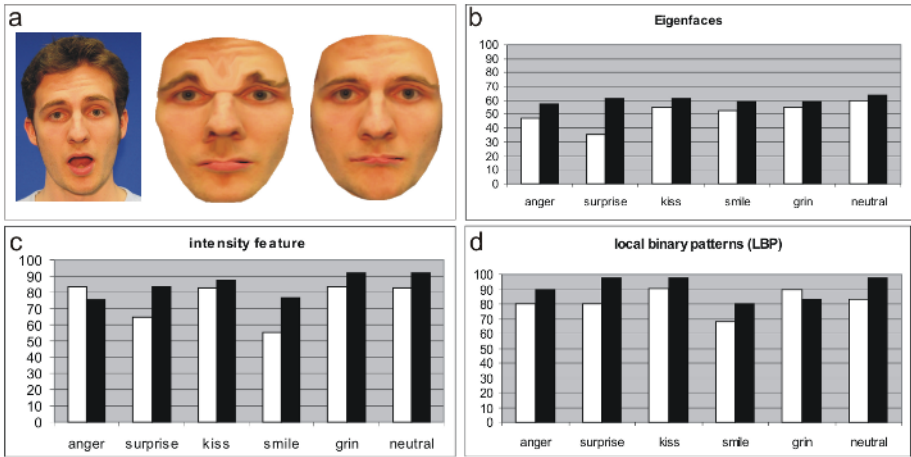
**Table 1.** Overall recognition results of the examined recognition methods

	without normalization	with normalization
Eigenfaces	51,0%	60,2%
intensity	75,3%	84,3%
LBP	82,0%	91,0%

For the eigenface approach a performance improvement of up to 26% could be achieved ("surprise"). In average the recognition performance of the eigenface approach could be increased by 9,2% (Fig. 3b).

The intensity feature based classification (Fig. 3c) shows that our approach can deal with facial expressions involving strong muscle tensions ("smile", "surprise") as well as with minor changes in expression ("grin", "neutral"). Only the "angry" facial expression leads to a decrease in performance. This can be explained by the fact that we can not reconstruct textures that are covered, e.g. the texture of the region above the eyes in the "angry" expression). The eigenface approach is not affected due to the data reduction inherited in the PCA.

The results of the LBP-based face recognition experiment (Fig. 3d) show that incorrect edge information can be contained in the normalized images. Here, the grinning expression led to a decreased recognition rate. However, for the other expressions an increase of 7% ("kiss") – 17,6% ("surprise") could be achieved, in average an increase of 9%. The overall recognition rate for the LBP-based recognition increased to 91%. These results motivate a combination of the recognition algorithms as different information is considered by the different features.



**Fig. 3.** a) Example of a normalization result of the two approaches (left: original; center: biomechanical model; right: RBF model); b)-d) Recognition rates of the examined methods for each expression using the pseudo-muscle model based normalization

## 4 Conclusion

We presented two approaches that can be used to model facial expressions found in a single still image by a generic 3D head model without using large image databases. The adaptation of the mesh is performed in a two-step process by first coarsely aligning the mesh until the minimal distance between corresponding feature points in the image and in the mesh is obtained. Especially, feature points that are largely invariant against facial changes are involved in this registration step. In comparison to that, feature points that are significantly moved under different facial expressions control the precise adaptation of the 3D mesh to the face image.

In this stage, a biologically inspired head model allows to simulate facial expressions using linear muscle models, i.e. the contraction parameters of the muscle models are estimated using a non-linear optimization process by further minimizing the distance between corresponding feature points. This approach should be chosen if further processing or changing of the facial expression in a realistic manner is intended.

A deformation based on RBF-interpolation using a Gaussian kernel function is not biologically motivated, but is more reliable due to its mathematical properties. If only a "distorted" mesh is needed, e.g. for a normalization step when performing automatic face recognition, this approach is more suitable than the biomechanical approach. The potential of this method was shown in face recognition experiments. By normalizing face images to a neutral expression, recognition rates could be improved significantly by using three recognition algorithms.

## Acknowledgments

This work was supported by the Federal Ministry of Economics and Labor of Germany under grant no. 20K0302Q. The authors are responsible for the contents of this publication.

## References

1. Pantic, M., Rothkrantz, L.: Automatic analysis of facial expressions: The state of the art. *IEEE PAMI* **22** (2002) 1424–1445
2. Hähnel, M., Fillbrandt, H.: Person recognition and tracking. In Kraiss, K., ed.: *Advanced Man-Machine Interaction*. Springer (2006) 191–261
3. Waters, K.: A muscle model for animating three-dimensional facial expressions. In: *Proceedings of SIGGRAPH 87*. Volume 21. (1987) 17–24
4. Lee, Y., Terzopoulos, D., Waters, K.: Realistic modeling for facial animation. In: *Proceedings of SIGGRAPH 95*, Los Angeles (1995) 55–62
5. Botsch, M., Kobbelt, L.: Real-Time Shape Editing using Radial Basis Functions. *Proc. of Eurographics '05* (2005)
6. Breton, G., Bouville, C., Pelé, D.: FaceEngine - A 3D facial animation engine for real time applications. In: *Proceedings of the WEB3D'01 Symp.* (2001) 15–22
7. Nocedal, J., Wright, S.: *Numerical Optimization*. Springer (1999)
8. Aloimonos, Y.: Perspective approximations. *Image Vision Comp.* (1990) 177–192
9. Turk, M., Pentland, A.: Eigenfaces for recognition. *Journal of Cognitive Neuroscience* (1991) 71–86
10. Ahonen, T., Hadid, A., Pietikäinen, M.: Face recognition with local binary patterns. In: *Proc. of the 8th ECCV*, Prague, Czech Republic (2004) 469–481

# Extraction of Haar Integral Features on Omnidirectional Images: Application to Local and Global Localization

Ouiddad Labbani-Igbida, Cyril Charron, and El Mustapha Mouaddib

Centre de Robotique, Electrotechnique et Automatique,  
Universit de Picardie Jules Verne, 7 rue du Moulin Neuf,  
F-80039 Amiens Cedex, France  
{ouiddad.labbani, cyril.charron, mouaddib}@u-picardie.fr

**Abstract.** In this paper, we present a new method for producing omnidirectional image signatures that are purposed to localize a mobile robot in an office environment. To solve the problem of perceptual aliasing common to the image based recognition approaches, we choose to build signatures that greatly vary between rooms and slowly vary inside a given room. We suggest an averaging technique based on Haar integral invariance. It takes into account the movements the robot can do in a room and the omni image transformations thus produced.

The variability of the built signatures is adjusted (total or partial Haar invariance) according to defined subsets of the group transformation. The experimental results prove to get significantly interesting results for place recognition and robot localization with variable accuracy: From global rough localization to local precise one.

## 1 Introduction

Recently, an increased interest in omnidirectional vision for robotic applications could be noted, thanks to the very large field of view of these sensors. However, theoretical and methodological challenges should be taken to account for the specific properties of the particular omnidirectional imaging sensor. This work deals with the localization of robots in an indoor office environment using an omnidirectional camera.

Unlike conventional methods for robot localization using landmarks [1] or map matching [2], image- (or appearance-) based localization approaches do not call for explicit models of the environment. These approaches (as [3,4,5]) prevent you from having to use a map and give a rough estimation of the robot location by matching a set of views taken by the robot to reference views stored in previous experiments.

We are interested in localizing a mobile robot in unknown and explored environments, using only omnidirectional images. The robot position estimation problem consists in finding the best match for the current image among the reference images. This can be a tricky problem if the environment displays symmetrical structures like doors and corridors, so the current view will match not only the referred location image, but also all similar images giving *perceptual aliasing*.



The approach we propose in this paper deals with this problem, giving an elegant way to formally compute image signatures using Haar integral features. We investigate partial and global Haar invariance to set different localization accuracies depending on the kind of motion required by the robot. We also provide a suitable adaptation of the signature extraction method to the geometrical image distortions caused by the catadioptric omni sensor.

The paper is organized as follows: We start (Section 2) with a review of the related work and, in Section 3, present our method of signature extractions using Haar features. Sections 4 and 5 give experimental evidence of the proposed method, and show comparative results of partial and total Haar signatures, in several indoor environments. Finally, Section 6 concludes the work with a summary and an outline of our ongoing work.

## 2 Related Works

Image-based localization methods are almost inspired by matching techniques, developed in the image retrieval field. Actually, image retrieval systems aim to find images that are similar in appearance to an input query, from a large-size database. However, while a few bad matches are not a problem in image retrieval, a single bad match could lead the robot localization system to get lost and must therefore be strictly avoided for the localization task.

Image retrieval systems usually rely on histograms for the matching process. This is due to their compact representation of the images, their invariance to rotation (which is very interesting for omnidirectional images) and their very low sensitivity to small translations. Multi-dimensional histograms of omnidirectional images were applied in [4] for robot indoor localization, and in [5] for outdoor environments using local characteristics, evaluated on multiple rings sampled on the omni-image. But the histograms are not invariant when important movements are involved.

Authors in [3,6,7] have issued image compact representations using subspaces of Fourier harmonics, i.e., they calculate the Fourier coefficients to represent images in a lower-dimensional subspace. These representations lack robustness since the Fourier transform is inherently a non-robust transformation to occlusions. To overcome the lack of robustness in case of perceptual aliasing, [8] used a Monte-Carlo localization technique and their system was able to estimate and track the position of the robot while it was moving. To achieve rotational independence, [9] proposed a zero-phase representation of images by zeroing the phase of the first harmonic of the Fourier transform. The method gives similar reference orientations for images taken at nearby positions, but is very sensitive to variations in the scene and occlusions (since operating with one single frequency).

The main interest of the previous techniques is to find an invariant representation to the omnidirectional image rotations (taken at the same position under different orientations of the robot). A solution is provided by wrapping images to cylindrical panoramic representations in that way a rotation of the original image is equivalent to a shift of the image plane deployed from the cylindrical image.

This is widely investigated in eigenspace approaches [10,11,12,13], that build a database model by computing the eigenvectors or the principal components on the wrapped images.

Invariance theory helps to build image features that should exhibit invariance against different transformations on the scenes. Lowe [14,15] proposed SIFT<sup>1</sup> features as invariants to image translation, scaling and rotation; and partially invariants to illumination changes. Variants of SIFT (Modified SIFT [16], Iterative SIFT [17]) have been proposed to reduce the computational efforts of the feature extraction and matching process, and applied in almost real time robot localization.

We propose in this paper to *formally* define invariant signatures of images based on *Haar integrals*. Firstly introduced by Schulz-Mirbach in [18], this invariant has been used, in case of euclidian motions, for image retrieval in [19,20,21], and for mobile robot localization [22] (although the Haar integral was not explicitly used). Haar integral invariant features could be extracted directly from raw images, without need to preprocessing such as segmentation or edge extraction.

### 3 Haar Invariant Signatures Extraction

#### 3.1 The Process of the Signature-Based Localization

The image-based localization approach is a twofold procedure: In a setup stage, the robot explores the environment following a training strategy and acquires several omnidirectional snapshots which together form a good depiction of the environment and constitute the reference images. Figure 1 sketches an example of an indoor environment seen by the omnidirectional sensor of the robot. Partial or total Haar invariant signatures (detailed below) are computed off line for the reference images. This allows for efficient memory consumption, efficient matching, and localization. In the running stage, the robot acquires new images, computes online their Haar signatures and finds the best match for the current image among the reference images.

#### 3.2 The General Idea of Haar Integral Features

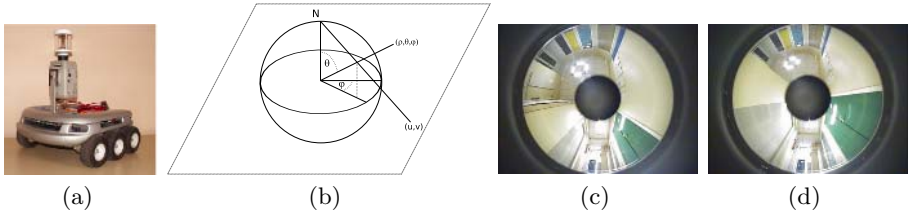
The Haar invariant integral could be viewed as a course through the space of the transformation group parameters. It is expressed as:

$$I_{Haar}(\mathbf{x}) = \frac{1}{|G|} \int_G f[g(\mathbf{x})] dg \text{ with } |G| = \int_G dg \quad (1)$$

where  $G$  is the transformation group, and  $g(x)$  the action of  $g$ , an element of  $G$ , on vector  $\mathbf{x}$ . In case of  $\mathbf{x} = \mathbf{M}$  being an image, eq.1 suggests that the integral invariant feature is computed by first 1) applying kernel function  $f$  to each pixel in transformed image  $g(\mathbf{M})$  then 2) summing up over all transformations of  $G$  and 3) normalizing the result to get a single representation of the invariant feature.

---

<sup>1</sup> Scale Invariant Feature Transform.



**Fig. 1.** The Koala robot (a) and its omnidirectional sensor, modelled by a projection on the equivalent sphere (b): To 3D point  $(\rho, \theta, \varphi)$  seen by the robot sensor, corresponds a point  $(u, v)$  in the raw image. (c) and (d) show images acquired by the robot sensor during a translation.

Our approach differs from existing Haar approaches [18,19] in the nature of the transformations considered. The authors deal with euclidian transformations of the images, given cyclic boundary conditions. These hypotheses on image transformations do not hold anymore when dealing with the geometry of omnidirectional sensors. We generalize Haar integral features to integrate the transformations induced by the geometry of the sensor and transformed under the robot movements (translations and rotations) in the scene. The variability of the built features can be adjusted (partial Haar invariance) according to defined subsets of the group transformation ( $G_j \subseteq G$ ). We finally define distributions based on a partition of the constructed (partial or global) Haar features and build up histograms of these distributions.

### 3.3 The Camera Transformation Model

Due to the complexity of omni-image transformations, they are often projected back onto a cylinder, and then mapped back by an isometry into a plane so as to have them looking more similar to classical perspective images. Here we use the equivalence sphere model given by Geyer and Daniilidis [23]. They prove that central catadioptric projection can be modeled with the projection of the sphere to a horizontal plane from a point on the vertical axis of the sphere. Once the sensor has been calibrated, the raw image is projected onto this sphere, equivalent to the actual mirror from the point of view of the image formation process (Fig. 1(b)).

Spherical image  $\mathbf{M}_S(\theta, \varphi)$  (eq.2) has a topology which looks more adapted to the sensor properties than the raw image.  $\mathbf{M}_S$  is formed by regularly meshing the sphere in  $(\theta, \varphi)$  and interpolating at the corresponding points in the original image,  $\mathbf{M}(u, v)$  using the projective equation (2):

$$\begin{cases} u = \cot(\theta/2) \cdot \cos(\varphi) \\ v = \cot(\theta/2) \cdot \sin(\varphi) \end{cases} \quad (2)$$

### 3.4 The Transformation Group

Without loss of generality, the reference frame of the robot and that of the mirror can be considered as aligned along the  $Oz$ -axis. Let's consider how the spherical

image,  $\mathbf{M}_S(\theta, \varphi)$  can be transformed when the robot moves. The rotation of the robot (at the same position) around its vertical axis by angle  $\varphi$  induces the rotation of the sphere around the  $Oz$  axis. Using an omnidirectional camera is an advantage to have a complete view so that the image content information does not change when changing the orientation.

When the robot motion involves a translation (robot changes in position), the transformation acting onto the spherical image is composed of translations in  $\theta$  and  $\varphi$ . We assume these transformations still verifying a group action, as almost scene information remain present in the omni-image. Due to the non uniform resolution and distorsion of omnidirectional images (see ex. fig.1), the image point transformations are not uniform and are weighted by the variation of Haar measure  $dg = \sin \theta d\theta d\varphi$ .

We exploit the symmetry of revolution of the considered kernel supports to remove the integration over the rotation group action. Finally, the group parameters acting on the images are translation parameters in  $\theta \in [\pi/2, \pi[$  and  $\varphi \in [0, 2\pi[$ . In the Haar integration over  $\theta$ , the transformation group support is varied to adjust the variability of features to local or global changes in the scene, corresponding to small or large movements of the robot.

### 3.5 The Kernel Function

The averaging technique to construct invariant features depends on kernel function  $f$ . The definition of kernel  $f$  appears to be important for the robustness and stability of the built invariant features. Work in [19,20] used a large set of monomials and relational kernel functions with local support to increase the completeness and non ambiguity of the invariant sets.

We use a kernel function of local characteristics based on a Difference of Gaussians ( $DoG$ ) (eq.3). The  $DoG$  is usually applied for keypoint detection, and is shown [15] to be a good approximation for the  $\sigma^2$ -Gaussian Laplacian which is invariant<sup>2</sup> to affine change of luminosity, rotation and locally invariant to perspective transform.

$$DoG(\theta, \varphi, \sigma) = \frac{1}{2\pi(k\sigma)^2} e^{-\frac{\theta^2 + \varphi^2}{2(k\sigma)^2}} - \frac{1}{2\pi\sigma^2} e^{-\frac{\theta^2 + \varphi^2}{2\sigma^2}} \quad (3)$$

Kernel function  $f : \mathbf{M}_S \mapsto \mathcal{C}_f$  defines a mapping from spherical image  $\mathbf{M}_S$  to feature space  $\mathcal{C}_f$ .  $f(\mathbf{M}_S)$  denotes the image of local features obtained by Haar integration and is produced by: 1) convoluting the (grey-scaled) spherical image points with the difference of two nearby scale gaussians; 2) averaging on the pixel neighbors belonging to DoG support  $\Delta_{DoG}$  (of size  $6k\sigma$ ); and 3) partitioning the DoG space into a fixed partition  $\{DoG_i\}_{i=1, \dots, k}$ . Similarly to [19], we build fuzzy partitions using continuous triangle functions to avoid discontinuities of feature assignments at the edges of DoG supports. We thus produce

---

<sup>2</sup> [24] referenced by Lowe [15].

$f(\mathbf{M}_S) = \{f_i(\mathbf{M}_S)\}_{i=1,\dots,k}$  that we normalize to make the sum at a given point of the  $f_i$  feature space equals one<sup>3</sup>.

$$f(\mathbf{M}_S(\theta_0, \varphi_0)) = \left\{ \sum_{(\theta, \varphi) \in \Delta_{DoG_i}} DoG_i(\theta - \theta_0, \varphi - \varphi_0, \sigma) * \mathbf{M}_S(\theta, \varphi) \right\}_{i \in [1, 2, \dots, k]} \tag{4}$$

Haar integration consists then in a course (path) between the *feature* image points belonging to every  $f_i$ , weighted by the Haar measure depending on their position in the image:  $I_{Haar} = \{I_{Haar_i}\}_{i=1,\dots,k}$  (eq.5). Written in the discrete case, we have for  $i = 1, \dots, k$ :

$$I_{Haar_i}(\mathbf{M}_S) = \frac{1}{\Delta_\theta} \frac{1}{2\pi} \sum_{\theta \in \Delta_\theta; \varphi \in [0, 2\pi[} f_i(\theta, \varphi) \sin \theta d\theta d\varphi \tag{5}$$

The size of  $\theta$ -support  $\Delta_\theta$  is varied from  $\pi/2$  (full integration) to  $\pi/2/l$  for  $l$  partial integrations over annulus regions of thickness  $l$ . The obtained distribution looks like a histogram,  $h(I_{Haar}(\mathbf{M}_S)) = \{I_{Haar_i}(\mathbf{M}_S); i = 1, \dots, k\}$  of invariant features and constitutes the image signature (parameterized by  $\sigma$ ).

Finally, to compare image signatures, we use the  $L_1$ -norm similarity measure between their Haar distributions. This similarity measure is averaged in case of  $l$  partial Haar integrations by:

$$D(\mathbf{M1}_S, \mathbf{M2}_S) = \frac{1}{l} \sum_{j=1}^l d(I_{Haar}(\mathbf{M1}_{S/j}), I_{Haar}(\mathbf{M2}_{S/j}))$$

where  $l$  is the thickness of the  $j^{th}$ -annulus region,  $\mathbf{M}_{S/j}$ , of spherical image  $\mathbf{M}_S$ .

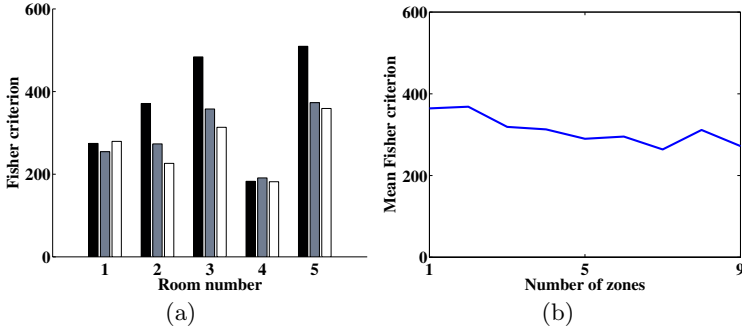
## 4 The Experimental Setup

The Koala robot, endowed with a catadioptric camera, was moved in our indoor lab environment, composed of several rooms, where the perceptual aliasing is particularly high (doors, "wall cupboards", ...). We have built an image database composed of approximately 250 images evenly distributed in the lab rooms. Images taken in a given room were manually clustered in an associated directory, thus allowing to calculate the *centroid* and the *variance* of the cluster with respect to the  $L_1$ -norm. This was repeated for every room the robot has surveyed.

## 5 The Experimental Results

This section reports experimental results we have obtained with the robot. We study the performance of the Haar invariant signatures in distinguishing different rooms as well as giving slightly shifted localizations in the same room.

<sup>3</sup>  $f_i$  could be seen as the probability for a characteristic to belong to a given feature bin.



**Fig. 2.** Comparison of the Fisher criterion using total and partial invariant Haar signatures ( $\sigma = 1$  and  $k = \sqrt{2}$ ). (a): Results for different lab rooms explored by the robot. *dark*: Total Haar integration, *grey* and *white*: Partial Haar integrations while sampling the omni images in 3 and 5 annulus regions respectively. (b): The mean values of Fisher criteria vs. the number of annulus regions used for Haar integration.

## 5.1 The Place Recognition Performance

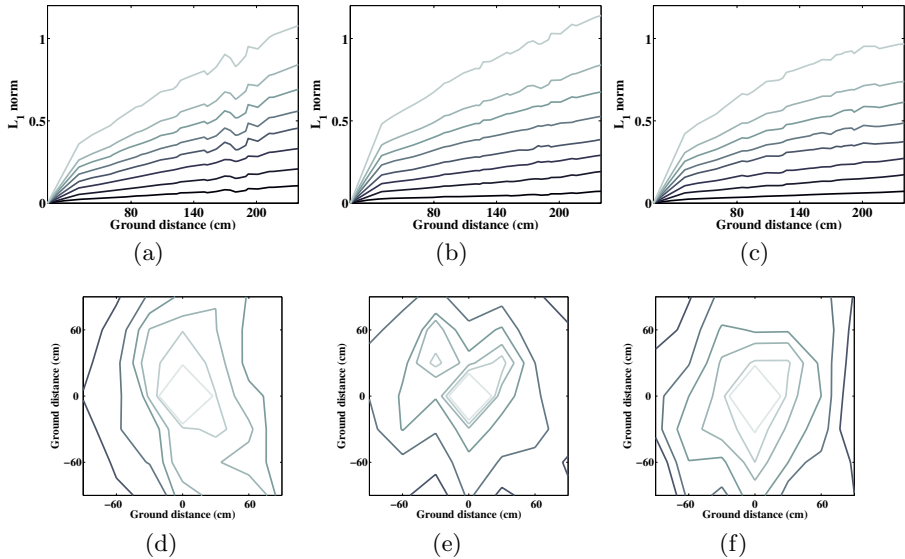
To differentiate between different places in the lab, in spite of the high perceptual aliasing, we need good clustering properties of the extracted features. To assess the clustering performance the proposed method, we use the *Fisher criterion* which measures the separation between two classes. It is defined as  $\mathcal{J} = (\eta_1 - \eta_2)^2 / (\sigma_1^2 + \sigma_2^2)$ , evaluating the ratio of the squared distance between centroids  $\eta_1$  and  $\eta_2$  of the classes over variances  $\sigma_1^2$  and  $\sigma_2^2$  of the representations belonging to them. The higher this ratio is, the better the separability between two classes is. Figure 2 shows the comparative results when total and several partial Haar integral techniques were used.

For all Haar invariant signatures, we obtain high Fisher criteria, which reveals a good separability and recognition ability of the lab rooms. In a previous work [25], we have compared total Haar integration to other signature extraction methods [9,7] for robot localization, and have shown an improvement of Fisher criteria of order 5 to 10, depending on the lab rooms.

As could be expected, by increasing the number of partial Haar integrations, we slightly weaken the Fisher criteria and so the separability of image equivalent classes. This is due to the reduction of the integral support size ( $G_j$ ).

## 5.2 The Robot Position Estimation Performance

We are now interested to evaluate the signature discrimination for the robot position estimation inside each room. When the robot moves in an area, small distortions are produced in the omnidirectional image space and we should expect small changes in the feature space. We study this through the evolution of the  $L_1$ -norm between signatures of the image database with respect to the physical distance between the positions they are associated to.



**Fig. 3.** (a)→(c): The evolution of the  $L_1$ -norm with respect to the distance (in cm) on the ground in different rooms. (d)→(f): Robot localization results, applying total and partial Haar integral signatures. *Dark line*: Total Haar integration. The *gradual lighting of grey levels* of lines corresponds to the increase of the number of group supports  $G_j$  in partial Haar integrations (resulting in increase of precision in robot localization).

The results are shown in fig. 3(a-c). Total Haar invariant signatures produce smaller variations than partial Haar invariant signatures for the same rooms. This is not surprising as we were looking to produce a global invariant. When increasing the number of partial Haar integrations, which in the same time, reduces the integral support size, we consider transformations on annulus regions that correspond to small movements in the nearby of the robot. This increases the precision of the robot localization. Figure 3(d-f) gives the position estimation of the robot in different rooms with variable accuracies corresponding to variable partial Haar integration supports. An interesting fact is that the variation of the (partial and total) Haar signatures is still *monotonic*. Thus, we can define a bijective relation between the  $L_1$ -norm and the physical distance, allowing a localization inside a given room using these signatures.

On an AMD 1800MHz, the whole process of the signature construction takes approximately 0.3s. The comparison process between a signature and all the signatures (250) of the image database takes around  $400\mu s$ .

## 6 Conclusion and Ongoing Work

We have proposed an efficient methodology to build invariant signatures for omni- image based localization applications. Haar integral formalism offers a solid theoretic foundation to the invariant signatures of images we have intro-

duced. The integration over the group transformations allows us to deal naturally with the geometric and projective transformations of omnidirectional sensors, as produced by the robot movements.

Our method benefits from the local invariance properties of the defined kernel function and the global invariance of Haar integration. The latter is tuned by varying the number the supports of the transformation group, corresponding to small movements in the nearby of the robot. Partial Haar integrations increase the accuracy of the robot localization.

Using the Fisher criterion, the built signatures have figured out a wide separation ability of room classes, contributing to reduce the perceptual aliasing. Moreover, the smooth variation and the continuity property of the built Haar signatures, inside each category, provides a good approximation to the robot position for localization.

Additional development is under way to build different non linear kernel functions as we believe that this will influence the stability and the precision of localization in a positive way.

## References

1. J. A. Castellanos and J. D. Tardos, *Mobile robot localization and map building: A multisensor fusion approach*. Kluwer Academic Publishers, Boston, Mass, 2000.
2. J. Borenstein, B. Everett, and L. Feng, *Navigating mobile robots: Systems and techniques*. A. K. Peters, Ltd., Wellesley, MA, February 1996.
3. H. Ishiguro and S. Tsuji, "Image-based memory of environment," in *IEEE/RSJ International Conference on Intelligent RObots and Systems*, vol. 2, 1996, pp. 634–639.
4. I. Ulrich and I. Nourbakhsh, "Appearance-based place recognition for topological localization," in *International Conference on Robotics and Automation*. San Francisco, CA: IEEE, April 2000, pp. 1023–1029.
5. J. Gonzalez and S. Lacroix, "Rover localization in natural environments by indexing panoramic images," in *IEEE International Conference on Robotics and Automation*, 2002, pp. 1365–1370.
6. M. Ishikawa, S. Kawashima, and N. Homma, "Memory-based location estimation and navigation using bayesian estimation," in *International Conference On Neural Information Processing*, vol. 1, October 1998, pp. 112–117.
7. E. Menegatti, T. Maeda, and H. Ishiguro, "Image-based memory for robot navigation using properties of the omnidirectional images," *Robotics and Autonomous Systems, Elsevier*, vol. 47, no. Issue 4, pp. 251–267, 2004.
8. E. Menegatti, M. Zoccarato, E. Pagello, and H. Ishiguro, "Image-based monte-carlo localisation with omnidirectional images," *Robotics and Autonomous Systems, Elsevier*, vol. 48, no. Issue 1, pp. 17–30, 2004.
9. T. Pajdla and V. Hlaváč, "Zero phase representation of panoramic images for image based localization," in *8-th International Conference on Computer Analysis of Images and Patterns*, F. Solina and A. Leonardis, Eds., no. 1689. Ljubljana, Slovenia: LNCS, Springer Verlag, September 1999, pp. 550–557.
10. M. Jogan and A. Leonardis, "Robust localization using an omnidirectional appearance-based subspace model of environment," *Robotics and Autonomous Systems*, vol. 45, no. 1, 2003.



11. S. Maeda, Y. Kuno, and Y. Shirai, "Active navigation vision based on eigenspace analysis," in *International Conference on Intelligent Robots and Systems*. IEEE/RSJ, 1997, pp. 1018–1023.
12. A. Leonardis and H. Bischof, "Robust recognition using eigenimages," *Computer Vision and Image Understanding Special Issue on Robust Statistical Techniques in Image Understanding*, vol. 78, no. 1, pp. 99–118, 2000.
13. J. Gaspar, N. Winters, and J. Santos-Victor, "Vision-based navigation and environmental representations with an omnidirectional camera," in *IEEE Transactions on Robotics and Automation*, vol. 16, no. 6, December 2000, pp. 890–898.
14. D. Lowe, "Object recognition from local scale-invariant features," in *International Conference on Computer Vision*, September 1999, pp. 1150–1157.
15. —, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
16. H. Andreasson, A. Treptow, and T. Duckett, "Localization for mobile robots using panoramic vision, local features and particle filter," in *IEEE International Conference on Robotics and Automation*, 2005.
17. H. Tamimi, H. Andreasson, A. Treptow, T. Duckett, and A. Zell, "Localization of mobile robots with omnidirectional vision using particle filter and iterative sift," in *European Conference on Mobile Robots*, Ancona, Italy, 2005.
18. H. Schulz-Mirbach, H. Burkhardt, and S. Sigglekow, "Using invariant features for content based data retrieval," in *Workshop on Nonlinear Methods in Model-Based Image Interpretation*, Lausanne, Switzerland, September 1996, pp. 1–5.
19. S. Sigglekow, "Feature histograms for content-based image retrieval," Ph.D. dissertation, Universitat Freiburg im Breusgau, 2002.
20. A. Halawani and H. Burkhardt, "Image retrieval by local evaluation of nonlinear kernel functions around salient points," in *International Conference on Pattern Recognition*, vol. 2, August 2004, pp. 955–960.
21. —, "On using histograms of local invariant features for image retrieval," in *IAPR Workshop on Machine Vision Applications*, May 2005, pp. 538–541.
22. H. B. J. Wolf, W. Burgard, "Using an image retrieval system for vision-based mobile robot localization," in *Proc. of the International Conference on Image and Video Retrieval (CIVR)*, M. S. Lew, N. Sebe, and J. P. Eakins, Eds. Springer-Verlag Berlin Heidelberg, 2002, pp. 108–119.
23. C. Geyer and K. Daniilidis, "Catadioptric projective geometry," *International Journal of Computer Vision*, vol. 45, no. 3, pp. 223–243, 2001.
24. T. Lindeberg, "Scale-space theory: A basic tool for analysing structures at different scales," *Journal of Applied Statistics*, vol. 21, no. 2, pp. 224–270, 1994.
25. C. Charron, O. Labbani-Igbida, and E. Mouaddib, "On building omnidirectional image signatures using haar invariant features: Application to the localization of robots," in *To appear in ACIVS*. Antwerp, Belgium: Lecture Notes in Computer Science, Springer Verlag, September 2006.

# Model Selection in Kernel Methods Based on a Spectral Analysis of Label Information

Mikio L. Braun<sup>1</sup>, Tilman Lange<sup>2</sup>, and Joachim M. Buhmann<sup>2</sup>

<sup>1</sup> Fraunhofer Institute FIRST, Intelligent Data Analysis Group,  
Kekuléstr. 7, 12489 Berlin, Germany  
mikio@first.fhg.de

<sup>2</sup> Institute of Computational Science, ETH Zurich, 8092 Zurich, Switzerland  
{langet, jbuhmann}@inf.ethz.ch

**Abstract.** We propose a novel method for addressing the model selection problem in the context of kernel methods. In contrast to existing methods which rely on hold-out testing or try to compensate for the optimism of the generalization error, our method is based on a structural analysis of the label information using the eigenstructure of the kernel matrix. In this setting, the label vector can be transformed into a representation in which the smooth information is easily discernible from the noise. This permits to estimate a cut-off dimension such that the leading coefficients in that representation contains the learnable information, discarding the noise. Based on this cut-off dimension, the regularization parameter is estimated for kernel ridge regression.

## 1 Introduction

Kernel methods represent a widely used family of learning algorithms for supervised learning. Irrespective of their theoretical motivation and background, kernel methods compute a predictor which can be expressed as

$$\hat{f}(x) = \sum_{i=1}^n k(x, X_i) \hat{\alpha}_i + \hat{\alpha}_0 \quad (1)$$

with  $X_i$  being the features of training examples  $(X_i, Y_i)$ ,  $k$  the kernel function and a parameter vector  $\hat{\alpha} = (\hat{\alpha}_0, \dots, \hat{\alpha}_n) \in \mathbb{R}^{n+1}$  which is determined by the learning algorithm based on the training examples. Typical examples for algorithms which generate this kind of fit include Support Vector Machines of various types, Kernel Ridge Regression, and Gaussian Processes.

Since all the algorithms have to solve basically the same problem of finding a parameter vector in eq. (1) such that the resulting  $\hat{f}$  leads to good predictions, the relationship between the space of all functions of the form (1) and the data-source generating the training examples provides an *a priori* condition of the learning task in the setting of kernel methods.

This leads to the question of model selection, either concerning the fitness of the kernel, or the choice of regularization parameters. This problem is commonly approached

by adopting a black-box approach, and estimating the generalization error by cross-validation. While this works well in practice (in particular when the cross-validation error can be computed efficiently, as is the case in the context of kernel ridge regression), the question arises, whether additional insight into the nature of the learning problem cannot lead to a less black-box method for model selection.

Now, recent approximation results on the eigenvalues ([1], [2], [3]) and eigenvectors ([4], [5]) of the kernel matrix, and in particular the improved bounds from [6], have lead to novel insights into the relationship of the label information  $Y_i$  and the eigenvectors of the kernel matrix which will allow us to address the question of model selection without resorting to hold-out-testing: Using the orthogonal basis of eigenvectors of the kernel matrix, one can estimate an effective dimensionality of the learning problem, based on which one can then select regularization constants.

This structural analysis of the label information is introduced in Section 2. In Section 3, we show how this analysis can be used to perform model selection in the context of kernel ridge regression, which we have picked as an example. In Section 4, we compare the resulting model selection method against state-of-the-art methods to show that competitive model selection without hold-out testing is possible.

## 2 Spectral Analysis of the Labels

In this section, we will discuss how recent approximation results imply that under certain conditions, a transformation of the vector of training labels using the eigenvectors of the kernel matrix leads to a new representation of the label vector where the interesting information is contained in the leading coefficients. By determining a cut-off dimension in this representation, one can effectively separate the relevant from the noise part in the training label information.

Fix a training set  $(X_1, Y_1), \dots, (X_n, Y_n)$  of size  $n$  and a kernel function  $k$ , which is assumed to be a Mercer kernel (see [7]). The *kernel matrix*  $\mathbf{K}$  is the  $n \times n$  matrix with entries  $[\mathbf{K}]_{ij} = k(X_i, X_j)$ .

For general data-sources, no easy answers can be expected, because the learning task can be arbitrarily ill-behaved. Therefore, we restrict the discussion to the case where the training examples are computed by subsampling a smooth function:

$$Y_i = f(X_i) + \varepsilon_i, \tag{2}$$

where  $\varepsilon_1, \dots, \varepsilon_n$  is independent zero mean noise. Smoothness of  $f$  is defined in the sense that  $f$  is a member of the reproducing kernel Hilbert space (RKHS)  $\mathcal{H}_k$  induced by  $k$ . More specifically, by Mercer's theorem, there exists a  $\ell^1$ -sequence  $(\gamma_i)_{i \in \mathbb{N}}$  and an orthogonal family of functions  $(\psi_i)_{i \in \mathbb{N}}$ , such that

$$k(x, y) = \sum_{i=1}^{\infty} \gamma_i \psi_i(x) \psi_i(y). \tag{3}$$

Then,  $f \in \mathcal{H}_k$ , iff  $f = \sum_{i=1}^{\infty} c_i \psi_i$ , with  $\|f\|_{\mathcal{H}_k}^2 := \sum_{i=1}^{\infty} c_i^2 / \gamma_i < \infty$ . Consequently, the coefficients  $c_i$  decay rather quickly.

It will be convenient to consider the vector of all labels  $Y = (Y_1, \dots, Y_n)$ . By our modelling assumption (2), with  $F = (f(X_1), \dots, f(X_n))$  and  $\varepsilon = (\varepsilon_1, \dots, \varepsilon_n)$ , we can write  $Y$  as the sum of a sample vector of a smooth function and noise:  $Y = F + \varepsilon$ . Obviously, in its original sample-wise representation, the two parts  $F$  and  $\varepsilon$  of  $Y$  are not easily distinguishable. We are looking for a change of representation which allows us to distinguish between  $F$  and  $\varepsilon$ . We will shortly see that the eigendecomposition of the kernel matrix can be used to this end.

Recall that the kernel matrix is symmetric and positive definite, since  $k$  is a Mercer kernel. Therefore, there exists a so-called eigendecomposition of  $\mathbf{K}$  as  $\mathbf{K} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$ , where  $\mathbf{U}$  is orthogonal (that is,  $\mathbf{U}\mathbf{U}^\top = \mathbf{U}^\top\mathbf{U} = \mathbf{I}$ ), and  $\mathbf{\Lambda} = \text{diag}(\lambda_1, \dots, \lambda_n)$ . We will assume throughout this paper that the columns of  $\mathbf{U}$  and  $\mathbf{\Lambda}$  have been ordered such that  $\lambda_1 \geq \dots \geq \lambda_n$ . It is easy to see that the  $i$ th column  $u_i$  of  $\mathbf{U}$  is the eigenvector of  $\mathbf{K}$  to the corresponding eigenvalue  $\lambda_i$ . Since  $\mathbf{U}$  is orthogonal, its columns (and therefore the eigenvectors of  $\mathbf{K}$ ) form an orthonormal basis of  $\mathbb{R}^n$ , the *eigenbasis* of  $\mathbf{K}$ .

Now since  $\mathbf{U}$  is orthogonal, we can easily compute the coefficients of  $Y$  with respect to the eigenbasis of  $\mathbf{K}$ ,  $u_1, \dots, u_n$ , simply by applying  $\mathbf{U}^\top$  to  $Y$ . We obtain,

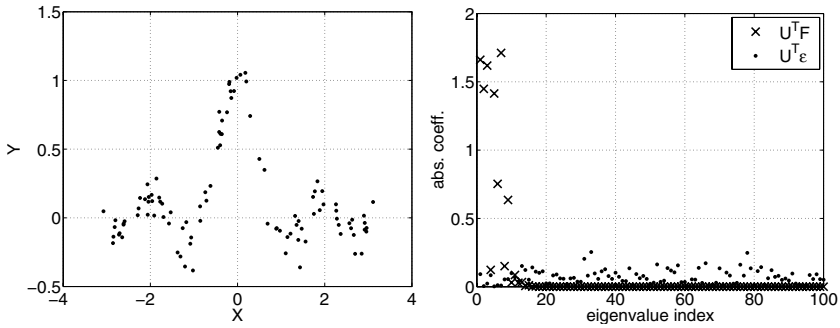
$$\mathbf{U}^\top Y = \mathbf{U}^\top (F + \varepsilon) = \mathbf{U}^\top F + \mathbf{U}^\top \varepsilon, \quad (4)$$

that is, the coefficients of  $Y$  are given by the superposition of the coefficients of  $F$  and those of the noise  $\varepsilon$ . The interesting observation is now that  $\mathbf{U}^\top F$  and  $\mathbf{U}^\top \varepsilon$  have radically different structural properties.

First, we have a look at  $\mathbf{U}^\top F$ . Recall that in (3), we have introduced an orthogonal family of functions  $(\psi_i)$ . These are also the eigenfunctions of the integral operator  $T_k$  associated with  $k$ . One can show that the scalar products  $\langle \psi_i, f \rangle$  are approximated by the scalar products  $u_i^\top F$ , due to the fact that  $\mathbf{K}/n$  approximates  $T_k$  in an appropriate sense as  $n \rightarrow \infty$  (the actual details are rather involved, see [4], [5] for a reference.) Now since the  $\psi_i$  are orthogonal,  $\langle \psi_i, f \rangle = c_i$ , and as  $f \in \mathcal{H}_k$ ,  $c_i$  decays to zero quickly. Therefore, since  $u_i^\top F$  approximates  $c_i$ , we can expect that  $u_i^\top F$  decays to zero as  $i \rightarrow n$  as well (recent results [6] show that even in the finite sample setting, the coefficients are approximated with high relative accuracy). The actual decay rate depends on the complexity (or non-smoothness) of  $f$ . In summary,  $\mathbf{U}^\top F$  will only have a finite number of large entries in the beginning (recall that we have sorted  $\mathbf{U}$  such that the associated eigenvalues are in non-increasing order.) In addition, this number is independent of the number of training examples, such that it is a true characterization of  $f$ .

Now let us turn to  $\mathbf{U}^\top \varepsilon$ . First of all, assume that  $\varepsilon$  is normally distributed with mean 0 and covariance matrix  $\sigma_\varepsilon^2 \mathbf{I}_n$ . In that case,  $\mathbf{U}^\top \varepsilon$  has the same distribution as  $\varepsilon$ , because  $\mathbf{U}^\top \varepsilon$  is just a (random) rotation of  $\varepsilon$  and, since  $\varepsilon$  is spherically distributed, so is  $\mathbf{U}^\top \varepsilon$ . Therefore, a single realization of  $\mathbf{U}^\top \varepsilon$  will typically be uniformly spread out, meaning that the individual coefficients  $[\mathbf{U}^\top \varepsilon]_i$  will all be on the same level. This behavior will still hold to a lesser extent if  $\varepsilon$  is not normally distributed as long as the variances for the different  $\varepsilon_i$  are similar. Thus, a typical realization of  $\varepsilon$  will be more or less uniformly spread out, and the same applies to  $\mathbf{U}^\top \varepsilon$ .

In summary, starting with the label vector  $Y$ , through an appropriate change of representation, we obtain an alternative representation of  $Y$  in which the two parts  $F$  and  $\varepsilon$  have significantly different structures:  $\mathbf{U}^\top F$  decays quickly, while  $\mathbf{U}^\top \varepsilon$  is uniformly



**Fig. 1.** The noisy sinc function. Left: The input data. Right: Absolute values of the coefficients with respect to the eigenbasis of the kernel matrix for a radial-basis kernel with width 0.3 of the subsampled function  $F$  and the noise  $\varepsilon$ , respectively. The coefficients of  $F$  decay quickly while those of  $\varepsilon$  are uniformly spread out.

spread out. Figure 1 illustrates these observations for the example of  $f(x) = \text{sinc}(4x)$ , and normally distributed  $\varepsilon$ .

### 2.1 Estimating the Cut-Off Dimension

The observations so far are interesting in their own right, but what we need is a method for automatically estimating the relevant, non-noise content  $F$  in  $Y$ . As explained in the last section,  $\mathbf{U}^\top Y = \mathbf{U}^\top F + \mathbf{U}^\top \varepsilon$ , and we can expect that there exists some *cut-off dimension*  $d$  such that for  $i > d$ ,  $[\mathbf{U}^\top Y]_i$  will only contain noise. The problem is that neither the exact shape of  $\mathbf{U}^\top F$ , nor the noise variance is in general known.

We thus propose the following heuristic for estimating  $d$ . Let  $s = \mathbf{U}^\top Y$  where  $s$  is assumed to be made up of two components:

$$s_i \sim \begin{cases} \mathcal{N}(0, \sigma_1^2) & 1 \leq i \leq j, \\ \mathcal{N}(0, \sigma_2^2) & j + 1 \leq i \leq n. \end{cases} \quad (5)$$

For the second part corresponding to the noise, the assumption of Gaussianity is actually justified if  $\varepsilon$  is Gaussian. For the first part, since prior knowledge is not available, the Gaussian distribution has been chosen as a baseline approximation. We will later see that this choice works very well despite its special form.

We perform a maximum likelihood fit for each  $j \in \{1, \dots, n - 1\}$ . The negative log-likelihood is then proportional to

$$l_j = \frac{j}{n} \log \sigma_1^2 + \frac{n-j}{n} \log \sigma_2^2, \quad \text{with} \quad \sigma_1^2 = \frac{1}{j} \sum_{i=1}^j s_i^2, \quad \sigma_2^2 = \frac{1}{n-j} \sum_{i=j+1}^n s_i^2. \quad (6)$$

We select the  $j$  which minimizes the negative log-likelihood, giving the cut-off point  $d$ , such that the first  $d$  eigenspaces contain the signal. The algorithm is summarized in Figure 2. The computational requirements are dominated by the computation of the eigendecomposition of  $\mathbf{K}$ , which requires about  $O(n^3)$ , and the computation of  $s$ . The log-likelihoods can then be computed in  $O(n)$ .

Input: kernel matrix  $\mathbf{K} \in \mathbb{R}^{n \times n}$ ,  
 labels  $Y = (Y_1, \dots, Y_n) \in \mathbb{R}^n$ .  
 Output: cut-off dimension  $d \in \{1, \dots, n - 1\}$

```

1 | compute eigendecomposition  $\mathbf{K} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$  with
   |  $\mathbf{\Lambda} = \text{diag}(\lambda_1, \dots, \lambda_n)$ ,  $\lambda_1 \geq \dots \geq \lambda_n$ .
2 |  $s = \mathbf{U}^\top Y$ .
3 | for  $j = 1, \dots, n - 1$ ,
3a |  $\sigma_1^2 = \frac{1}{j} \sum_{i=1}^j s_i^2$ ,  $\sigma_2^2 = \frac{1}{n-j} \sum_{i=j+1}^n s_i^2$ ,
3b |  $l_j = \frac{j}{n} \log \sigma_1^2 + \frac{n-j}{n} \log \sigma_2^2$ .
4 | return  $d = \text{argmin}_{j=1, \dots, n-1} l_j$ 
    
```

**Fig. 2.** Estimating the cut-off dimension given a kernel matrix and a label vector

### 3 Model Selection for Kernel Ridge Regression

We will next turn to the problem of estimating the regularization constant in Kernel Ridge Regression (KRR). It is typically used with a family of kernel functions, for example rbf-kernels. The method itself has a regularization parameter  $\tau$  which controls the complexity of the fit as well. These two parameters have to be supplied by the user or be automatically inferred in some way.

Let us briefly review Kernel Ridge Regression. The fit is computed as follows:

$$\hat{f}(x) = \sum_{i=1}^n k(x, X_i) \hat{\alpha}_i, \text{ with } \hat{\alpha} = (\mathbf{K} + \tau \mathbf{I})^{-1} Y. \tag{7}$$

One can show (see for example [8]) that this amounts to computing a least-squares fit with penalty  $\tau \alpha^\top \mathbf{K} \alpha$ . There is also a close connection to Gaussian Processes [9], in that  $\hat{f}$  is equivalent to the maximum a posteriori estimate using Gaussian processes in a Bayesian framework. The complexity of the fit depends on the kernel function and the regularization parameter with larger  $\tau$  leading to solutions which are more regularized. The model selection task consists in determining a  $\tau$  which reconstructs the function  $f$  best while suppressing the noise.

#### 3.1 The Spectrum Method for Estimating the Regularization Parameter $\tau$

We will now discuss how the cut-off dimension from Section 2 could be used to determine the regularization constant given a fixed kernel. The idea is to adjust  $\tau$  such that the resulting fit reconstructs the signal up to the cut-off dimension, discarding the noise.

In order to understand how this could be accomplished, we first re-write the in-sample fit computed by kernel ridge regression using the eigendecomposition of the kernel matrix:

$$\hat{Y} = \mathbf{K}(\mathbf{K} + \tau \mathbf{I})^{-1} Y = \mathbf{U}\mathbf{\Lambda}(\mathbf{\Lambda} + \tau \mathbf{I})^{-1} \mathbf{U}^\top Y = \sum_{i=1}^n u_i \frac{\lambda_i}{\lambda_i + \tau} u_i^\top Y. \tag{8}$$

As before, the scalar products  $u_i^\top Y$  compute the coefficients of  $Y$  expressed in the basis  $u_1, \dots, u_n$ . KRR then computes the fit by shrinking these coefficients by the factor  $\lambda_i/(\lambda_i + \tau)$ , and reconstructing the resulting fit in the original basis. These factors  $w_i = \lambda_i/(\lambda_i + \tau)$  depend on the eigenvalues and the regularization parameter, and will be close to 1 if the eigenvalues are much larger than  $\tau$ , and close to 0 otherwise. Now, for kernel usually employed in the context of kernel methods (like the rbf-kernel), the eigenvalues typically decay very quickly, such that the factors  $w_i$  approximate a step function. Therefore, KRR approximately projects  $Y$  to the eigenspaces belonging to the first few eigenvectors, and the number of eigenvectors depends on the regularization parameter  $\tau$ .

We wish to set  $\tau$  such that the factor  $w_d$  is close to 1 at the cut-off point  $d$  and starts to decay for larger indices. Therefore, we adjust  $\tau$  such that require that  $w_d > \rho$ , for some threshold  $\rho$  close to 1. This leads to the choice

$$\tau = w_d = \frac{\lambda_d}{\lambda_d + \tau} \quad \Rightarrow \quad \tau = \frac{1 - \rho}{\rho} \lambda_d. \quad (9)$$

The choice of  $\rho$  is rather arbitrary, but the method itself is not very sensitive to this choice. We have found that  $\rho = 10/11$  works quite well in practice. We will call this method of first estimating the cut-off dimension and then setting the regularization parameter according to (9) the *spectrum method*.

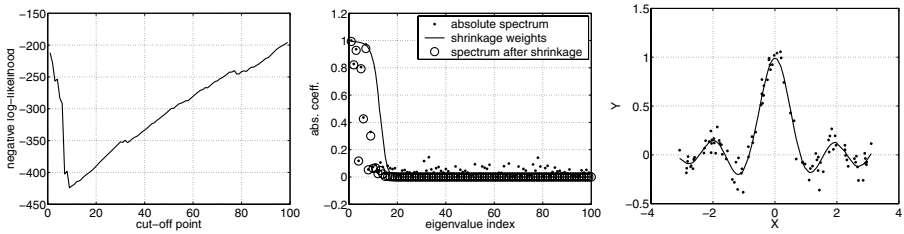
The proposed procedure is admittedly rather ad-hoc, however, note that the underlying mechanisms are theoretically verified. Also, in the choice of  $\tau$ , we make sure that no relevant information in the labels is discarded. Depending on the rate of decay of the eigenvalues, further dimensions will potentially be included in the reconstruction. However, this effect is in principle less harmful than estimating a too low dimension, because additional data points can correct this choice, but not the error introduced by estimating a too low dimension.

## 4 Experimental Evaluation

In this final section, we will compare the spectrum method to a number of state-of-the-art methods. This experimental evaluation should study whether it is possible to achieve competitive model selection based on our structural analysis. Unless otherwise noted, the other methods will be used as follows: For estimating regularization constants, the respective criterion (test-error or likelihood) is evaluated for the same possible values as available for the spectrum method, and the best performing value is taken. If the kernel widths is also determined, again all possible values are tested and the best performing candidate is taken. For the spectrum method, the regularization parameter is first determined by the spectrum method, and then, the kernel with the best leave-one-out error is selected. All data sets were iterated over 100 realizations.

### 4.1 Regression Data Sets

For regression, we will compare the spectrum method (SM) with leave-one-out cross-validation (CV) and evidence-maximization for Gaussian processes (GPML). For kernel ridge regression, it is not necessary to recompute the solution for all  $n - 1$  instances



**Fig. 3.** The noisy sinc function. Left: The negative log-likelihood for different cut-off points. Middle: The coefficients of signal and noise, and the shrinkage factors for the  $\tau$  selected by the spectrum method. One can see that the noise is nicely filtered out. Right: The resulting fit.

with one point removed, but the leave-one-out cross-validation error can be calculated in closed-form (see for example [10]).

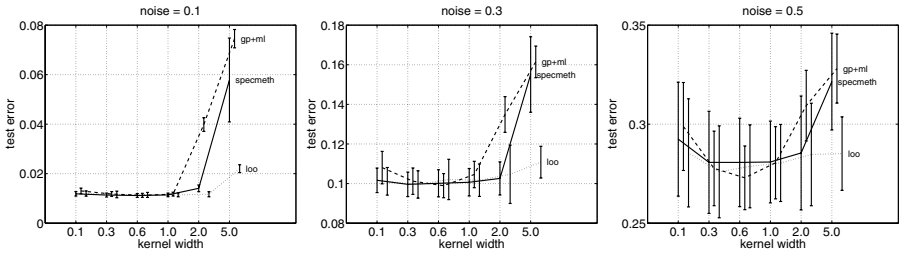
Evidence-maximization for Gaussian processes works by choosing the parameters which maximize the marginal log-likelihood of the labels, which is derived, for example, in [9, eq. (4)]. Note that this approach is fairly general and can be extended to more kernel parameters which are then determined by gradient descent. For our application, we will restrict ourselves to a single kernel width for all directions and performing an exhaustive search.

**The Noisy Sinc Function.** We begin with a small illustrative example: The *noisy sinc function* example is defined as follows: The  $X_i$  are drawn uniformly from  $[-\pi, \pi]$ , and  $Y_i = \text{sinc}(4X_i) + 0.1\varepsilon_i$ , where  $\varepsilon_i$  is  $\mathcal{N}(0, 1)$ -distributed. A typical example data set for  $n = 100$  is shown in Figure 1. The kernel width is  $c = 0.3$ . In the left panel of Figure 3, the negative log-likelihood is plotted. The minimum is at  $d = 9$ , which results in  $\tau = 0.145$ . In the right panel, the spectra of the data are plotted before and after shrinkage, together with the shrinkage coefficients. One can see that the noise is nicely suppressed. In the lower panel, the resulting fit is plotted.

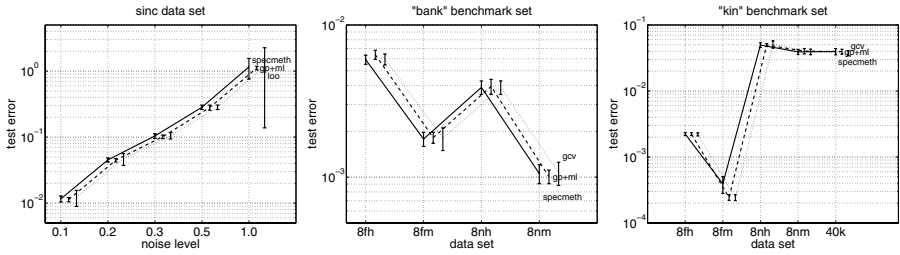
Next we want to study the robustness of the algorithms. We vary the kernel width and the noise levels. The resulting test errors for the CV and GPML and its standard deviation are plotted in Figure 4. We see that the spectrum method performs competitively to CV and GPML, except at large kernel widths, but we also see that GPML is much more sensitive to the choice of the kernel. It seems that evidence maximization tries to compensate for a mismatch between the kernel width and the actual data. For the optimal kernel width (around  $c = 0.6$ ), evidence maximization yields very good results, but for too small or too large kernels, the performance deteriorates. To be fair, we should add that evidence maximization is normally not used in this way. Usually, the kernel width is included in the adaptation process.

**Benchmark Data Sets.** Next, we compare the methods by also estimating the kernel width. We have compared the three procedures on the sinc data set as introduced above, and also for the bank and kin(etic) data sets from the DELVE repository (<http://www.cs.toronto.edu/~delve>), and a variant of the kin data set, called kin40k, prepared by A. Schwaighofer (<http://www.cis.tugraz.at/igi/aschwaig/data.html>).





**Fig. 4.** The noisy sinc functions for different noise levels and kernel widths. Widths  $c$  were chosen from  $\{0.1, 0.3, 0.6, 1.0, 2.0, 5.0\}$ , and noise variances from  $\{0.1, 0.3, 0.5\}$ . Training set size was 100, test set size 1000.



**Fig. 5.** Benchmark data sets. Both parameters, the kernel width  $c$  and the regularization constant  $\tau$  were estimated. Training set size was 100, test set size was 100 for sinc, 39000 for kin40k, and 8092 else.

Figure 5 shows the resulting test errors for the three methods. We see that all three methods show the same performance. The only exception is the kin-8fm data set, where the spectrum method results in a slightly larger error. We conclude that the spectrum method performs competitively to the state-of-the-art procedures CV and GPML. On the positive side, the spectrum method gives more insight into the structure of the data set than cross-validation and it requires weaker modelling assumptions than GPML.

Table 1 shows the cut-off dimensions for the sinc, bank, and kin data set. For the sinc data set, the cut-off dimension decreases with increasing noise. This behavior can be interpreted as the noise masking the fine structures of the data. The same effect is visible for the “h” (high noise) data sets versus the “m” (moderate noise) data sets. We also see that the data sets are moderately complex, having at most 17 significant coefficients in the spectral analysis.

### 4.2 Classification Data Sets

Next, we would like to evaluate the spectrum method for classification. Since the estimation of the cut-off dimension did not depend on the loss function with which the label differences are measured, the procedure should in principle also work for classification.

As usual, in order to apply Kernel Ridge Regression to classification, we use labels  $+1$  and  $-1$ . With that, the target function  $f$  is given as  $f(x) = E(Y|X = x)$ . The noise

**Table 1.** Cut-off dimensions for different data sets

sinc	$\sigma_\varepsilon = 0.1$	0.2	0.3	0.5	1.0
$d$	$9 \pm 1$	$9 \pm 1$	$8 \pm 2$	$8 \pm 2$	$5 \pm 4$
bank	8fh	8fm	8nh	8nm	
$d$	$9 \pm 1$	$11 \pm 4$	$10 \pm 3$	$17 \pm 8$	
kin	8fh	8fm	8nh	8nm	40k
$d$	$7 \pm 2$	$9 \pm 2$	$6 \pm 3$	$7 \pm 3$	$8 \pm 4$

**Table 2.** Test errors and standard deviations on the benchmark datasets from [11] (also available online from <http://www.first.fhg.de/~raetsch>) Each data set has already been split into 100 realizations of training and test data. The best achieved test errors (having the smallest variance in the case of equality) have been highlighted. The last column shows the kernel widths used for all three algorithms.

Dataset	SVM	SM	GCV	$c$
banana	$11.5 \pm 0.7$	<b><math>10.6 \pm 0.5</math></b>	$10.8 \pm 0.7$	1
breast-cancer	<b><math>26.0 \pm 4.7</math></b>	$27.0 \pm 4.7$	$26.3 \pm 4.6$	50
diabetes	$23.5 \pm 1.7$	<b><math>23.2 \pm 1.6</math></b>	$23.2 \pm 1.8$	20
flare-solar	<b><math>32.4 \pm 1.8</math></b>	$33.8 \pm 1.6$	$33.7 \pm 1.6$	30
german	$23.6 \pm 2.1$	<b><math>23.5 \pm 2.1</math></b>	<b><math>23.5 \pm 2.1</math></b>	55
heart	$16.0 \pm 3.3$	<b><math>15.9 \pm 3.1</math></b>	$18.7 \pm 6.7$	120
image	<b><math>3.0 \pm 0.6</math></b>	$3.1 \pm 0.4$	$6.3 \pm 4.1$	30
ringnorm	<b><math>1.7 \pm 0.1</math></b>	$4.9 \pm 0.7$	$6.6 \pm 2.0$	10
splice	<b><math>10.9 \pm 0.6</math></b>	$11.3 \pm 0.6$	$11.9 \pm 0.5$	70
titanic	<b><math>22.4 \pm 1.0</math></b>	$22.8 \pm 0.9$	$22.6 \pm 0.9$	2
thyroid	$4.8 \pm 2.2$	<b><math>4.4 \pm 2.2</math></b>	$12.6 \pm 4.1$	3
twonorm	$3.0 \pm 0.2$	<b><math>2.4 \pm 0.1</math></b>	$2.7 \pm 0.3$	40
waveform	$9.9 \pm 0.4$	$10.0 \pm 0.5$	<b><math>9.7 \pm 0.4</math></b>	20

is then  $Y - E(Y|X = x)$ , which has mean zero, but has a discrete distribution, and a non-uniform variance.

We use the benchmark data set from [11], which consists of thirteen artificial and real world data sets. We compare the spectrum method to a tentative gold-standard achieved by a support vector machine (SVM) whose hyperparameters have been fine-tuned by exhaustive search and  $k$ -fold cross validation. Furthermore, we compare the spectrum method to generalized cross validation (GCV) [12].

Table 2 plots the results. Over all, the spectrum method performs very well and achieves roughly the same classification rates as the support vector machine. GCV performs worse on a number of data sets. Note that GCV has the same possible values for  $\tau$  at its disposal including those values leading to a better performance. For those data sets we have performed GCV again, letting  $\tau$  vary from  $10^{-6}$  to 10, but this improves the results only on the *image* data set to  $4.6 \pm 2.1$ . Finally, we repeated the experiments for a subset of the data sets, this time choosing the kernel widths by the spectrum method and  $k$ -fold cross validation as in the SVM case. While this produced different kernel widths, the results were not significantly different, which underlines the robustness of the spectrum method.

In summary, we can conclude that the spectrum method performs very well on real-world classification data sets, and even outperforms generalized cross validation on a number of data sets.

## 5 Conclusion

We have proposed a novel method for model selection for kernel ridge regression which is not based on correcting for the optimism of the training error, or on some form of hold-out testing, but which employs a structural analysis of the learning problem at hand. By estimating the number of relevant leading coefficients of the label vector represented in the basis of eigenvectors of the kernel matrix, we obtain a parameter which can be used to pick a regularization constant leading to good performance. In addition, one obtains a structural insight into the learning problem in the form of the estimated dimensionality.

**Acknowledgements.** This work has been supported by the DFG grant #Buh 914/5, the PASCAL Network of Excellence (EU #506778), and BMBF grant 16SV2231 (FaSor).

## References

1. Koltchinskii, V., Giné, E.: Random matrix approximation of spectra of integral operators. *Bernoulli* **6**(1) (2000) 113–167
2. Taylor, J.S., Williams, C., Cristianini, N., Kandola, J.: On the eigenspectrum of the Gram matrix and the generalization error of kernel PCA. *IEEE Transactions on Information Theory* **51** (2005) 2510–2522
3. Blanchard, G.: Statistical properties of kernel principal component analysis. *Machine Learning* (2006)
4. Koltchinskii, V.I.: Asymptotics of spectral projections of some random matrices approximating integral operators. *Progress in Probability* **43** (1998) 191–227
5. Zwald, L., Blanchard, G.: On the convergence of eigenspaces in kernel principal component analysis. In: *NIPS 2005*. (2005)
6. Braun, M.L.: Spectral Properties of the Kernel Matrix and their Application to Kernel Methods in Machine Learning. PhD thesis, University of Bonn (2005) published electronically, available at [http://hss.ulb.uni-bonn.de/diss\\_online/math\\_nat\\_fak/2005/braun\\_mikio](http://hss.ulb.uni-bonn.de/diss_online/math_nat_fak/2005/braun_mikio).
7. Vapnik, V.: *Statistical Learning Theory*. J. Wiley (1998)
8. Cristianini, N., Shawe-Taylor, J.: *An Introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press (2000)
9. Williams, C.K.I., Rasmussen, C.E.: Gaussian processes for regression. In Touretzky, D.S., Mozer, M.C., Hasselmo, M.E., eds.: *Advances in Neural Information Processing Systems 8*, MIT Press (1996)
10. Wahba, G.: *Spline Models For Observational Data*. Society for Industrial and Applied Mathematics (1990)
11. Rätsch, G., Onoda, T., Müller, K.R.: Soft margins for AdaBoost. *Machine Learning* **42**(3) (2001) 287–320 also NeuroCOLT Technical Report NC-TR-1998-021.
12. Golub, G., Heath, M., Wahba, G.: Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics* **21** (1979) 215–224

# Importance-Weighted Cross-Validation for Covariate Shift

Masashi Sugiyama<sup>1</sup>, Benjamin Blankertz<sup>2</sup>, Matthias Krauledat<sup>2,3</sup>,  
Guido Dornhege<sup>2</sup>, and Klaus-Robert Müller<sup>2,3</sup>

<sup>1</sup> Department of Computer Science, Tokyo Institute of Technology, Tokyo, Japan

<sup>2</sup> Fraunhofer FIRST.IDA, Berlin, Germany

<sup>3</sup> Department of Computer Science, University of Potsdam, Potsdam, Germany

sugi@cs.titech.ac.jp

{benjamin.blankertz, matthias.krauledat, guido.dornhege,  
klaus}@first.fhg.de

**Abstract.** A common assumption in supervised learning is that the input points in the training set follow the *same* probability distribution as the input points used for testing. However, this assumption is not satisfied, for example, when the outside of training region is extrapolated. The situation where the training input points and test input points follow *different* distributions is called the covariate shift. Under the covariate shift, standard machine learning techniques such as empirical risk minimization or cross-validation do not work well since their unbiasedness is no longer maintained. In this paper, we propose a new method called importance-weighted cross-validation, which is still unbiased even under the covariate shift. The usefulness of our proposed method is successfully tested on toy data and furthermore demonstrated in the brain-computer interface, where strong non-stationarity effects can be seen between calibration and feedback sessions.

## 1 Introduction

The goal of supervised learning is to infer an unknown input-output dependency from training samples, by which output values for unseen test input points can be estimated. When developing a method of supervised learning, it is commonly assumed that the input points in the training set and the input points used for testing follow the *same* probability distribution (e.g., [9,3,5]). However, this common assumption is not fulfilled, for example, when the outside of training region is extrapolated and when training input points are designed by an active learning (experimental design) algorithm.

The situation where the training input points and test input points follow different probability distributions is called the *covariate shift* [6]. For data from many applications such as off-policy reinforcement learning, bioinformatics, or brain-computer interfacing, the covariate shift phenomenon is conceivable.

In an idealized situation where the model used for learning is *correctly specified* (i.e., the learning target is included in the model), empirical risk minimization

(ERM, cf. Eq.(4)) which is a typical parameter learning method still gives an asymptotically unbiased estimator of the true parameter even under the covariate shift. However, in practical situations where the model is *misspecified* (i.e., the learning target is not included in the model), the asymptotic unbiasedness<sup>1</sup> does not hold anymore; ERM yields a biased estimator even asymptotically.

To illustrate this phenomenon, let us employ a toy regression problem of fitting a linear function to the sinc function (see Figure 1). Here, we consider an extrapolation problem: training input points are distributed in the left-hand side of the input domain, while test input points are distributed in the right-hand side. The density functions of the training and test input points are depicted by the solid and dashed lines in Figure 1-(A). If ordinary least-squares (OLS) (which is an ERM method with squared-loss) is used for fitting the straight line, we have a good approximation of the left-hand side of the sinc function (see Figure 1-(B)). However, this is not an appropriate function for estimating the test output values ('×' in the figure). Thus, OLS results in a large test error.

Under the covariate shift with misspecified models, *importance-weighted ERM* (IWERM, cf. Eq.(6)) is shown to give an asymptotically unbiased estimator [6]. The key idea of IWERM is to weight the empirical risk according to the *importance*, which is the ratio of densities of the training and test input points. By this density ratio, the training input distribution is systematically adjusted to the test input distribution.

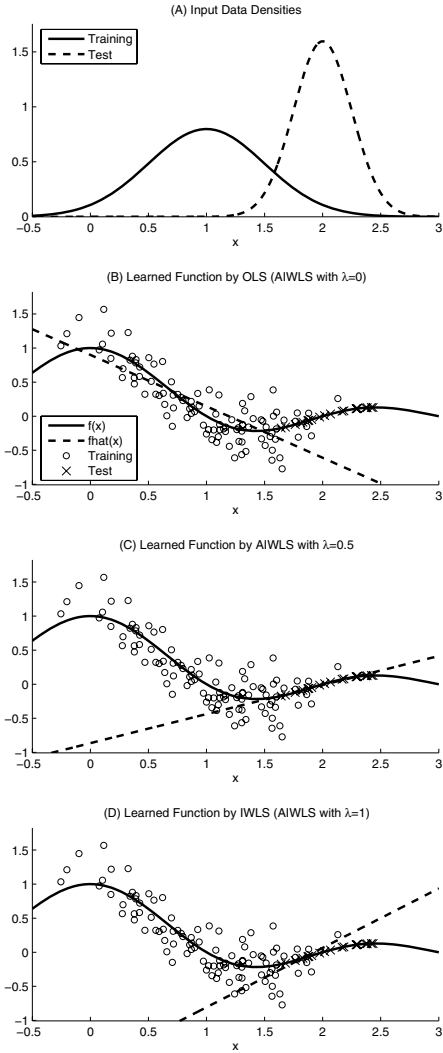
Figure 1-(D) depicts the learned function obtained by importance-weighted least-squares (IWLS). Compared with OLS, IWLS gives a better function for estimating the test output values; the learned function converges to the optimal function as the number of training samples goes to infinity.

The asymptotic unbiasedness can be achieved by IWERM, which may result in good estimation of the test output values, as illustrated above. However, IWERM generally yields an estimator with larger variance than ordinary ERM. This may be intuitively confirmed by the fact that OLS is the best linear unbiased estimator, i.e., having the smallest variance among all linear unbiased estimators. Therefore, IWERM may not be optimal; a slightly biased variant of IWERM with smaller variance could be better. The bias-variance trade-off may be controlled by slightly 'weakening' the importance in IWERM [6] or by adding a regularization term to IWERM. We refer to such a variance-reduced variant as *adaptive IWERM* (AIWERM, cf. Eq.(8)). AIWERM includes a tuning parameter  $\lambda$  ( $0 \leq \lambda \leq 1$ );  $\lambda = 0$  corresponds to ordinary ERM (uniform weight) and  $\lambda = 1$  corresponds to IWERM (weight equal to the importance).

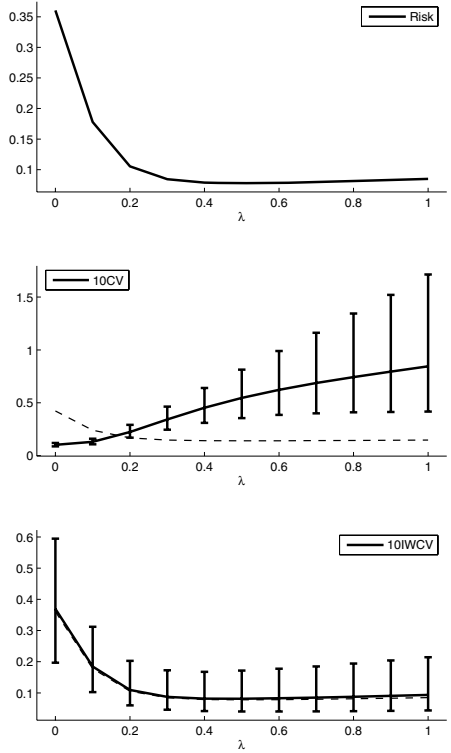
Figure 1-(C) depicts a learned function obtained by AIWLS with  $\lambda = 0.5$ , which yields much better estimation of the test output values than IWLS (AIWLS with  $\lambda = 1$ ) or OLS (AIWLS with  $\lambda = 0$ ).

---

<sup>1</sup> Usually an estimator is said to be unbiased if the expectation of the estimator agrees with the true parameter. For a misspecified model, we say that an estimator is unbiased if the expectation of the estimator agrees with the optimal parameter in the model (i.e., the optimal approximation of the learning target).



**Fig. 1.** An illustrative example of extrapolation by fitting a linear function. (A) The probability density functions of the training and test input points. (B)–(D) The learning target function  $f(x)$  (the solid line), the noisy training samples ('o'), a learned function  $\hat{f}(x)$  (the dashed line), and the (noiseless) test samples ('x').



**Fig. 2.** True risk and its estimations as functions of the tuning parameter  $\lambda$  in AIWLS. Dotted curves in the bottom two graphs depict the true risk for clear comparison.

As the above simple regression example demonstrates, AIWERM can work very well given  $\lambda$  is chosen appropriately. However,  $\lambda = 0.5$  is not always the best choice; a good value of  $\lambda$  may depend on the learning target, used model, noise in the training samples, etc. Therefore, for enhancing generalization capability under the covariate shift, *model selection* should be carried out: set the value of the tuning parameter  $\lambda$  so that the estimated risk (or the estimated generalization error) is minimized.

One of the popular techniques for estimating the risk in the machine learning community is *cross-validation* (CV). CV has been shown to give an almost unbiased estimate of the risk with finite samples [5]. However, this almost unbiasedness is no longer true under the covariate shift. This phenomenon is illustrated in Figure 2, which depicts the values of the true risk and its estimates as functions of the tuning parameter  $\lambda$  in AIWLS (the same toy regression example of Figure 1 is still used). The dotted curves in the bottom two graphs depict the true risk for clear comparison. In this example, the true risk hits the bottom at around  $\lambda = 0.5$  (see the top graph of Figure 2). On the other hand, CV gives a totally different, monotone increasing curve (see the second graph of Figure 2). As a result, CV chooses  $\lambda = 0$  as the best value, which appears to be a poor choice.

To cope with this problem, we propose using a novel variant of CV called *importance-weighted* CV (IWCV). We prove that IWCV is guaranteed to give an almost unbiased estimate of the risk even under the covariate shift. The bottom graph of Figure 2 shows the estimated risk obtained by IWCV. It gives much better estimation than ordinary CV, and therefore an appropriate value of  $\lambda$  may be chosen by IWCV.

## 2 Problem Formulation

In this section, we formulate the supervised learning problem and review existing learning methods.

### 2.1 Supervised Learning Under Covariate Shift

Let us consider the supervised learning problem of estimating an unknown input-output dependency from training samples. Let  $\mathcal{T} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$  be the training samples, where  $\mathbf{x}_i \in \mathcal{X} \subset \mathbb{R}^d$  is an i.i.d. training input point following a probability distribution with density  $p(\mathbf{x})$  and  $y_i \in \mathcal{Y} \subset \mathbb{R}$  is a training output value following a conditional probability distribution with conditional density  $r(y_i|\mathbf{x}_i)$ .

Let  $\ell(\mathbf{x}, y, \hat{y}) : \mathcal{X} \times \mathcal{Y} \times \mathcal{Y} \rightarrow [0, \infty)$  be the loss function, which measures the discrepancy between the true output value  $y$  at an input point  $\mathbf{x}$  and its estimate  $\hat{y}$ . In regression scenarios where  $\mathcal{Y}$  is continuous, the squared-loss is often used.

$$\ell(\mathbf{x}, y, \hat{y}) = (\hat{y} - y)^2. \quad (1)$$

On the other hand, in classification scenarios where  $\mathcal{Y}$  is discrete (i.e., categorical), the following 0/1-loss is a typical choice since it corresponds to the misclassification rate.

$$\ell(\mathbf{x}, y, \hat{y}) = \begin{cases} 0 & \text{if } \hat{y} = y, \\ 1 & \text{otherwise.} \end{cases} \tag{2}$$

Although the above loss functions are independent of  $\mathbf{x}$ , the loss can generally depend on  $\mathbf{x}$  [5].

Let us use a parameterized function  $\hat{f}(\mathbf{x}; \boldsymbol{\theta})$  for estimating the output value  $y$ , where  $\boldsymbol{\theta} \in \Theta \subset \mathbb{R}^p$ . The goal of supervised learning is to determine the value of the parameter  $\boldsymbol{\theta}$  so that the expected loss for the test samples (i.e., the risk or the generalization error) is minimized. Let  $(\mathbf{t}, u)$  be a test sample, where  $\mathbf{t} \in \mathcal{X}$  is a test input point and  $u \in \mathcal{Y}$  is a test output value following the conditional distribution with conditional density  $r(u|\mathbf{t})$ . Note that the conditional density  $r(\cdot|\cdot)$  is the same conditional density as the training output values  $\{y_i\}_{i=1}^n$ . Then the risk is expressed as

$$R^{(n)} = \mathbb{E}_{\{\mathbf{x}_i, y_i\}_{i=1}^n, \mathbf{t}, u} \left[ \ell(\mathbf{t}, u, \hat{f}(\mathbf{t}; \hat{\boldsymbol{\theta}})) \right], \tag{3}$$

where  $\mathbb{E}$  denotes the expectation. Note that the learned parameter  $\hat{\boldsymbol{\theta}}$  generally depends on the training set  $\mathcal{T} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ .

In standard supervised learning theories (e.g., [9,3,5]), the test input point  $\mathbf{t}$  is assumed to follow  $p(\mathbf{x})$ , which is the *same* probability density as the training input points  $\{\mathbf{x}_i\}_{i=1}^n$ . On the other hand, in this paper, we consider the situation under the *covariate shift*, i.e., the test input point  $\mathbf{t}$  follows a probability distribution with density  $q(\mathbf{t})$ , which is *different* from  $p(\mathbf{x})$ .

## 2.2 Empirical Risk Minimization and Its Importance-Weighted Variants

A standard method to learn the parameter  $\boldsymbol{\theta}$  would be empirical risk minimization (ERM):

$$\hat{\boldsymbol{\theta}}_{ERM} = \operatorname{argmin}_{\boldsymbol{\theta} \in \Theta} \left[ \frac{1}{n} \sum_{i=1}^n \ell(\mathbf{x}_i, y_i, \hat{f}(\mathbf{x}_i; \boldsymbol{\theta})) \right]. \tag{4}$$

If  $p(\mathbf{x}) = q(\mathbf{x})$ ,  $\hat{\boldsymbol{\theta}}_{ERM}$  is an asymptotically unbiased estimator of the optimal parameter. However, under the covariate shift where  $p(\mathbf{x}) \neq q(\mathbf{x})$ , ERM does not provide an asymptotically unbiased estimator anymore;  $\hat{\boldsymbol{\theta}}_{ERM}$  is biased even asymptotically:

$$\lim_{n \rightarrow \infty} \left\{ \mathbb{E}_{\{\mathbf{x}_i, y_i\}_{i=1}^n} \left[ \hat{\boldsymbol{\theta}}_{ERM} \right] \right\} \neq \boldsymbol{\theta}^*, \tag{5}$$

where  $\boldsymbol{\theta}^* = \operatorname{argmin}_{\boldsymbol{\theta} \in \Theta} \left\{ \mathbb{E}_{\mathbf{t}, u} \left[ \ell(\mathbf{t}, u, \hat{f}(\mathbf{t}; \boldsymbol{\theta})) \right] \right\}$ .

Under the covariate shift, the following importance-weighted ERM (IWERM) gives an asymptotically unbiased estimator [6]:

$$\hat{\boldsymbol{\theta}}_{IWERM} = \operatorname{argmin}_{\boldsymbol{\theta} \in \Theta} \left[ \frac{1}{n} \sum_{i=1}^n \frac{q(\mathbf{x}_i)}{p(\mathbf{x}_i)} \ell(\mathbf{x}_i, y_i, \hat{f}(\mathbf{x}_i; \boldsymbol{\theta})) \right], \tag{6}$$



which satisfies

$$\lim_{n \rightarrow \infty} \left\{ \mathbb{E}_{\{\mathbf{x}_i, y_i\}_{i=1}^n} \left[ \widehat{\boldsymbol{\theta}}_{IWERM} \right] \right\} = \boldsymbol{\theta}^*. \tag{7}$$

From here on, we assume that  $p(\mathbf{x})$  and  $q(\mathbf{x})$  are known and strictly positive (i.e., non-zero) for all  $\mathbf{x} \in \mathcal{X}$ .

Although the asymptotic unbiasedness is guaranteed in IWERM, it generally has larger variance than ordinary ERM [6]. Therefore, IWERM may not be optimal; a slightly biased variant of IWERM could have much smaller variance, and thus is more accurate than plain IWERM. The bias-variance trade-off may be controlled, for example, by weakening the weight (Adaptive IWERM; AIWERM):

$$\widehat{\boldsymbol{\theta}}_{AIWERM} = \operatorname{argmin}_{\boldsymbol{\theta} \in \Theta} \left[ \frac{1}{n} \sum_{i=1}^n \left( \frac{q(\mathbf{x}_i)}{p(\mathbf{x}_i)} \right)^\lambda \ell(\mathbf{x}_i, y_i, \widehat{f}(\mathbf{x}_i; \boldsymbol{\theta})) \right], \tag{8}$$

where  $0 \leq \lambda \leq 1$ .

The above AIWERM is just examples; there may be many other possibilities for controlling the bias-variance trade-off. However, we note that the methodology we propose in this paper is valid for *any* parameter learning method.

### 2.3 Cross-Validation Estimate of Risk

Now we want to determine the value of the tuning parameter, say  $\lambda$ , so that the risk  $R^{(n)}$  is minimized—but  $R^{(n)}$  is inaccessible. A standard approach to coping with this problem is to prepare some candidates  $\{\lambda_i\}$  of the tuning parameter, to estimate the risk for each candidate, and to choose the one with minimum estimated risk.

Cross-validation (CV) is a popular method to estimate the risk  $R^{(n)}$ . Let us divide the training set  $\mathcal{T} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$  into  $k$  disjoint non-empty subsets  $\{\mathcal{T}_i\}_{i=1}^k$ . Let  $\widehat{f}_{\mathcal{T}_j}(\mathbf{x})$  be a function learned from  $\{\mathcal{T}_i\}_{i \neq j}$ . Then the  $k$ -fold CV ( $k$ CV) estimate of the risk  $R^{(n)}$  is given by

$$\widehat{R}_{kCV}^{(n)} = \frac{1}{k} \sum_{j=1}^k \frac{1}{|\mathcal{T}_j|} \sum_{(\mathbf{x}, y) \in \mathcal{T}_j} \ell(\mathbf{x}, y, \widehat{f}_{\mathcal{T}_j}(\mathbf{x})), \tag{9}$$

where  $|\mathcal{T}_j|$  is the number of samples in the subset  $\mathcal{T}_j$ . When  $k = n$ ,  $k$ CV is particularly called the leave-one-out cross-validation (LOOCV).

$$\widehat{R}_{LOOCV}^{(n)} = \frac{1}{n} \sum_{j=1}^n \ell(\mathbf{x}_j, y_j, \widehat{f}_j(\mathbf{x}_j)), \tag{10}$$

where  $\widehat{f}_j(\cdot)$  is a function learned from  $\{(\mathbf{x}_i, y_i)\}_{i \neq j}$ .

It is known that, if  $p(\mathbf{x}) = q(\mathbf{x})$ , LOOCV gives an almost unbiased estimate of the risk; more precisely, LOOCV gives an unbiased estimate of the risk with  $n - 1$  samples [5].

$$\mathbb{E}_{\{\mathbf{x}_i, y_i\}_{i=1}^n} \left[ \widehat{R}_{LOOCV}^{(n)} \right] = R^{(n-1)} \approx R^{(n)}. \tag{11}$$

However, this is no longer true under the covariate shift with  $p(\mathbf{x}) \neq q(\mathbf{x})$ . In the following section, we give a novel modified cross-validation method which still maintains the ‘almost unbiasedness’ property even under the covariate shift.

### 3 Importance-Weighted Cross-Validation

Under the covariate shift, we propose using the following importance-weighted cross-validation (IWCV):

$$\widehat{R}_{kIWCV}^{(n)} = \frac{1}{k} \sum_{j=1}^k \frac{1}{|\mathcal{T}_j|} \sum_{(\mathbf{x}, y) \in \mathcal{T}_j} \frac{q(\mathbf{x})}{p(\mathbf{x})} \ell(\mathbf{x}, y, \widehat{f}_{\mathcal{T}_j}(\mathbf{x})), \tag{12}$$

or

$$\widehat{R}_{LOOIWCV}^{(n)} = \frac{1}{n} \sum_{j=1}^n \frac{q(\mathbf{x}_j)}{p(\mathbf{x}_j)} \ell(\mathbf{x}_j, y_j, \widehat{f}_j(\mathbf{x}_j)). \tag{13}$$

Below, we prove that LOOIWCV gives an almost unbiased estimate of the risk even under the covariate shift (its proof is given in a separate technical report [8]).

**Lemma 1**

$$\mathbb{E}_{\{\mathbf{x}_i, y_i\}_{i=1}^n} \left[ \widehat{R}_{LOOIWCV}^{(n)} \right] = R^{(n-1)}. \tag{14}$$

This lemma shows that the simple variant of CV called IWCV provides an unbiased estimate of the risk with  $n - 1$  samples even under the covariate shift. A similar proof is also possible for  $k$ IWCV, although its bias is larger than LOOIWCV.

The density ratio  $q(\mathbf{x})/p(\mathbf{x})$  also appears in *importance sampling*; an expectation  $\mathbb{E}_{\mathbf{t}}[f(\mathbf{t})]$  with  $\mathbf{t} \sim q(\mathbf{x})$  is computed by an equivalent quantity  $\mathbb{E}_{\mathbf{x}}[f(\mathbf{x})q(\mathbf{x})/p(\mathbf{x})]$  with  $\mathbf{x} \sim p(\mathbf{x})$ , where  $p(\mathbf{x})$  is chosen so that the variance is minimized. Therefore, the proposed IWCV method could be regarded as an application of the importance sampling identity in the CV framework. We expect that the relation between importance sampling and covariate shift may be further discussed in the context of *active learning* [7], where the training input density  $p(\mathbf{x})$  is designed by users so that the risk is minimized.

A weighted CV scheme has also been studied in robust statistics [1], where the effect of outliers in the CV score is deemphasized by assigning smaller weight to outliers. In the proposed IWCV scheme, the CV score is weighted by the density ratio, by which the difference between  $p(\mathbf{x})$  and  $q(\mathbf{x})$  can be systematically adjusted. Therefore, although using a weighted scheme in CV is a common feature, the aim is essentially different; we may even combine two schemes.

### 4 A Numerical Example

In this section, we experimentally investigate how IWCV works using a simple one-dimensional regression dataset (see Figure 1). Let the training and test input densities be  $p(x) = \phi_{1, (1/2)^2}(x)$  and  $q(x) = \phi_{2, (1/4)^2}(x)$ , where  $\phi_{\mu, c^2}(x)$  denotes the normal density with mean  $\mu$  and variance  $c^2$ . This setting implies that we are

**Table 1.** Extrapolation in the toy dataset. The mean and standard deviation of the test error obtained by each method are described. For reference, the test error obtained with the optimal  $\lambda$  (i.e., the minimum test error) is described as ‘OPT’.

10CV	10IWCV	OPT
$0.360 \pm 0.108$	$0.086 \pm 0.041$	$0.073 \pm 0.023$

considering an extrapolation problem (see Figure 1-(A)). We create the output value  $y_i$  following  $\phi_{f(x)1,(1/4)^2}(x)$ , where  $f(x) = \text{sinc}(x)$ . We use a simple linear model for learning:

$$\hat{f}(x; \theta_0, \theta_1) = \theta_0 + \theta_1 x, \tag{15}$$

where the parameters are learned by adaptive importance-weighted least-squares (AIWLS):

$$\underset{\theta_0, \theta_1}{\text{argmin}} \left[ \frac{1}{n} \sum_{i=1}^n \left( \frac{q(\mathbf{x}_i)}{p(\mathbf{x}_i)} \right)^\lambda \left( \hat{f}(\mathbf{x}_i; \theta_0, \theta_1) - y_i \right)^2 \right]. \tag{16}$$

Figure 1 (B)–(D) show the true function, a realization of training samples, learned functions by AIWLS with  $\lambda = 0, 0.5, 1$ , and a realization of (noiseless) test samples. For this particular case,  $\lambda = 0.5$  seems to work well.

Figure 2 depicts the means and standard deviations of the true risk and its estimates by 10-fold CV and 10-fold IWCV over 1000 runs, as functions of the tuning parameter  $\lambda$  in AIWLS. The graphs show that IWCV gives much accurate estimates of the risk than ordinary CV; the unbiasedness of IWCV is well satisfied and the variance of IWCV seems to be reasonable.

We then choose  $\lambda$  from  $\{0, 0.1, 0.2, \dots, 1\}$  so that the ordinary CV score or the IWCV score is minimized. The means and standard deviations of the test error finally obtained by ordinary CV and IWCV over 1000 runs are described in Table 1. The table shows that IWCV gives much smaller test errors than ordinary CV; the p-value between ordinary CV and IWCV by the t-test is far less than 0.01, stating that IWCV significantly outperforms ordinary CV. ‘OPT’ in the table shows the test error when  $\lambda$  is chosen optimally, i.e., so that the true test error is minimized. The result shows that the performance of IWCV is rather close to the optimal choice.

## 5 Application to Brain-Computer Interface

In this section, we apply IWCV to brain-computer interface (BCI) data.

BCI is a system which allows for a direct dialog between man and machine [11]. Cerebral electric activity is recorded via the electroencephalogram (EEG): electrodes, attached to the scalp, measure the electric signals of the brain. These signals are amplified and transmitted to the computer, which translates them into device control commands. The crucial requirement for the successful functioning of BCI is that the electric activity on the scalp surface already reflects motor intentions, i.e., the neural correlate of preparation for hand or foot movements. A BCI can detect the motor-related EEG changes and uses this information, for example, to perform a choice between two alternatives: the detection

**Table 2.** Misclassification rates for brain computer interface. All values are in percent. The values of the better method are described using a bold face.

Subject	Trial	# of training samples	# of unlabeled samples	# of test samples	AIWLDA		AIWLDA
					LDA	+ 10IWCV	+ OPT
1	1	280	112	112	9.8	<b>8.0</b>	8.0
	2	280	120	120	10.8	10.8	6.7
	3	280	35	35	5.7	<b>2.9</b>	2.9
2	1	280	113	112	43.4	43.4	43.4
	2	280	112	112	38.5	38.5	38.5
	3	280	35	35	28.6	28.6	28.6
3	1	280	91	91	39.6	<b>38.5</b>	37.4
	2	280	112	112	22.3	<b>19.6</b>	19.6
	3	280	30	30	20.0	20.0	20.0
4	1	280	112	112	24.1	24.1	23.2
	2	280	126	126	2.4	2.4	2.4
	3	280	35	35	8.6	8.6	8.6
5	1	280	112	112	<b>22.3</b>	25.0	22.3
	2	280	112	112	12.5	<b>11.6</b>	10.7

of the preparation to move the left hand leads to the choice of the first, whereas the right hand intention would lead to the second alternative. By this means it is possible to operate devices which are connected to the computer.

For classification of appropriately preprocessed EEG signals linear discriminant analysis (LDA) [3] has shown to work very well [2]. On the other hand, strong non-stationarity effects have been observed in brain signals between calibration and feedback sessions [10], which could be regarded as an example of the covariate shift. Therefore, it is expected that some importance-weighted method could further improve the BCI recognition accuracy.

LDA is actually equivalent to least-square fitting of a linear model using binary labels  $y_i = \pm 1$  [3]. Here we use its variant called adaptive importance-weighted LDA (AIWLDA):

$$\operatorname{argmin}_{\theta_0, \boldsymbol{\theta}} \left[ \frac{1}{n} \sum_{i=1}^n \left( \frac{q(\mathbf{x}_i)}{p(\mathbf{x}_i)} \right)^\lambda \left( \theta_0 + \boldsymbol{\theta}^\top \mathbf{x}_i - y_i \right)^2 \right]. \quad (17)$$

We test the above method with totally 14 data sets obtained from 5 different subjects (see Table 2). In BCI, the densities  $p(\mathbf{x})$  and  $q(\mathbf{x})$  are unknown. Here we estimate them by fitting the mixture of 5 Gaussians by the EM algorithm.  $p(\mathbf{x})$  is estimated using training samples and  $q(\mathbf{x})$  is estimated using unlabeled samples from the feedback period. The unlabeled samples are taken from the first half of each feedback period, herewith rendering the conditions for a BCI application realistic. This corresponds to an update of the used classifier in the second half of the experiment.

The misclassification rates for test samples by LDA (existing method which corresponds to AIWLDA with  $\lambda = 0$ ) and AIWLDA with  $\lambda$  chosen by 10IWCV

are given in Table 2. The results show that for the subjects 1 and 3, the combination of AIWLDA and 10IWCV highly improves the recognition accuracy over plain LDA. The accuracy is unchanged for the subjects 2 and 4, and comparable for the subject 5. Overall, the proposed method outperforms LDA for 5 out of 14 data sets and being outperformed for 1 data set.

Note that the degree of non-stationarity is highly subject specific. While—as expected—our method for compensating covariate shift effects yields highly significant improvements for some subjects, others exhibit no change due to the rather stationary nature of their brain signals.

## 6 Conclusions

In this paper, we discussed the supervised learning problem under the covariate shift paradigm: training input points and test input points are drawn from different distributions. Future studies will focus on the development of a realtime version of the current idea in order to ultimately obtain a fully adaptive learning system.

We acknowledge partial financial supports from MEXT (Grant-in-Aid for Young Scientists 17700142) and BMBF (FKZ 01IBE01A/B).

## References

1. C. Agostinelli. Robust model selection by cross-validation via weighted likelihood methodology. Technical Report 1999.37, Dipartimento di Scienze Statistiche, Università di Padova, 1999.
2. B. Blankertz, G. Dornhege, M. Krauledat, K.-R. Müller, and G. Curio. The Berlin brain-computer interface: Report from the feedback sessions. Technical Report 1, Fraunhofer FIRSST, 2005.
3. R. O. Duda, P. E. Hart, and D. G. Stor. *Pattern Classification*. Wiley, New York, 2001.
4. J. J. Heckman. Sample selection bias as a specification error. *Econometrica*, 47(1):153–162, 1979.
5. B. Schölkopf and A. J. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.
6. H. Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90(2):227–244, 2000.
7. M. Sugiyama. Active learning in approximately linear regression based on conditional expectation of generalization error. *Journal of Machine Learning Research*, 7(Jan):141–166, 2006.
8. M. Sugiyama, B. Blankertz, M. Krauledat, G. Dornhege, and K.-R. Müller. Importance-weighted cross-validation for covariate shift. Technical Report TR06-0002, Department of Computer Science, Tokyo Institute of Technology, Feb. 2006.
9. V. N. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.
10. C. Vidaurre, A. Schlögl, R. Cabeza, and G. Pfurtscheller. About adaptive classifiers for brain computer interfaces. *Biomedizinische Technik*, 49(1):85–86, 2004.
11. J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan. Brain-computer interfaces for communication and control. *Clinical Neurophysiology*, 113(6):767–791, 2002.

# Parameterless Isomap with Adaptive Neighborhood Selection

Nathan Mekuz and John K. Tsotsos

Center for Vision Research (CVR) and  
Department of Computer Science and Engineering,  
York University, Toronto, Canada M3J 1P3  
{mekuz, tsotsos}@cs.yorku.ca

**Abstract.** Isomap is a highly popular manifold learning and dimensionality reduction technique that effectively performs multidimensional scaling on estimates of geodesic distances. However, the resulting output is extremely sensitive to parameters that control the selection of neighbors at each point. To date, no principled way of setting these parameters has been proposed, and in practice they are often tuned ad hoc, sometimes empirically based on prior knowledge of the desired output. In this paper we propose a parameterless technique that adaptively defines the neighborhood at each input point based on intrinsic dimensionality and local tangent orientation. In addition to eliminating the guesswork associated with parameter configuration, the adaptive nature of this technique enables it to select optimal neighborhoods locally at each point, resulting in superior performance.

## 1 Introduction

Dimensionality reduction is a statistical tool commonly used to map data in high-dimensional space such as images, speech signals, etc. into lower dimensionality. The transformed data is often more suitable for regression analysis or classification than the original input data. Social sciences use dimensionality reduction extensively to uncover latent variables that explain observed phenomena. The underlying assumption is that observed high-dimensional samples lie on or near a lower-dimensional manifold embedded within the original high-dimensional space, and the purpose of the reduction is to project the high-dimensional data into a more compact representation while preserving certain properties of the data.

Traditional linear dimensionality reduction algorithms include Principal Component Analysis (PCA) [1] - a transformation that maximizes retained variance and Linear Discriminant Analysis (LDA) [2] - a projection that maximizes separation based on class labels. Nonlinear approaches include kernel PCA [3] - an application of linear PCA on data first transformed to typically higher dimensionality through some nonlinear kernel.

A recent surge in interest in locally linear manifold learning technique has resulted in the introduction of several new techniques, including, Isomap [4],

Locally Linear Embedding (LLE) [5] and its derivatives Laplacian eigenmaps [6], Hessian eigenmaps [7] and others, for goals ranging from visualization problems to classification. These techniques view the manifold as a patchwork of connected linear surfaces, and attempt to preserve certain properties in the projection. If the manifold is continuous and sufficiently well sampled, then using Taylor's theorem, small patches can be approximated as linear. If parts of the manifold are linear, globally nonlinear methods may be overly complex, and difficult to train due to the large number of parameters. On the other hand, locally linear techniques may model the manifold effectively by fitting parts of it separately if they are able to decompose it into linear components. However, local modeling is sensitive to noise and the modeling of noise remains a challenge. Consequently, most locally linear techniques do not address the issue of noise.

Isomap [4] is a popular locally linear technique that works by assuming isometry of geodesic distances in the manifold. The geodesic distance is defined as the distance of the shortest path between two points that passes on the embedded manifold [8]. Isomap estimates geodesic distances by constructing a graph with Euclidean distances between neighboring points as edge weights and computing shortest paths in the graph. Finally, classical MDS is applied to compute an optimal embedding. A computationally efficient implementation that computes shortest paths to only a subset of landmark data points is presented in [9].

A central problem in Isomap and many other locally linear techniques (e.g., [5,6,7,10]) lies in the selection of neighbors that form local patches. The shape of the manifold is in most cases unknown but a common assumption is that in small patches the surface is smooth, and that close neighbors of a data point likely lie on the same part of the manifold and have a similar orientation. Therefore, properties of the locality at each data point are commonly estimated using its nearest neighbors. Two formulations are commonly used: a fixed number of neighbors ( $k$  nearest neighbors), or all neighbors within a fixed radius ( $\epsilon$  hypersphere). The  $k$  nearest neighbors version is more common since the sparseness of the resulting structures is guaranteed. For example, the cost matrix used to compute an LLE embedding can have at most  $4kN$  nonzero elements. Efficient versions exist of the Dijkstra algorithm (used in Isomap) that take advantage of the sparseness of the input graph. On the other hand, if an  $\epsilon$  hypersphere is used, it is difficult to predict if a selected radius will include any neighbors at all at every point.

With either formulation, the choice of parameter typically has a dramatic effect on the transformation. If the neighborhoods are too small, disconnected clusters tend to form. Isomap maps the manifold in this case as a set of disjoint components, while LLE applies regularization on the cost matrix, but in both cases the global structure is lost. Since LLE performs a set of local optimizations, it is highly dependent on links created by sufficiently large neighborhoods to discern global structure. On the other hand, setting the neighborhood to a size that is too large creates links to parts of the manifold that are geodesically far. Isomap is especially sensitive to this problem since the shortest paths algorithm will tend to drain multiple paths through such shortcuts, affecting distance es-

timates globally. However, with small neighborhood sizes, the computed graph geodesic greatly overestimates the true geodesic distances in linear surfaces.

If the dimensionality reduction technique used assumes linear patches, then a good strategy for selecting these parameters needs to consider the (estimated) orientation of the manifold at each point. The selection should be data-driven and depend on such factors as curvature and density. But since curvature and density may vary over the manifold, one global setting may not work well for the entire manifold. In practice, examples such as the popular “Swiss roll” are presented where curvature and density are fairly constant everywhere. The parameters are often configured ad hoc, often by empirically evaluating the embeddings produced with different settings. However, if the Swiss roll is stretched (as in Figure 1), forming areas of varying curvature, then no global setting of  $k$  produces satisfying results.

In this paper, we describe a practical strategy for selecting a neighborhood size adaptively that does not require any parameters, based on estimates of intrinsic dimensionality and tangent orientation. We apply our technique to the Isomap algorithm and demonstrate simple manifolds where traditional neighborhood formulations fail, while our technique generates satisfactory mappings. The elimination of the parameter does not reduce the technique’s flexibility, since there is no way to configure this parameter automatically without prior knowledge of the desired output.

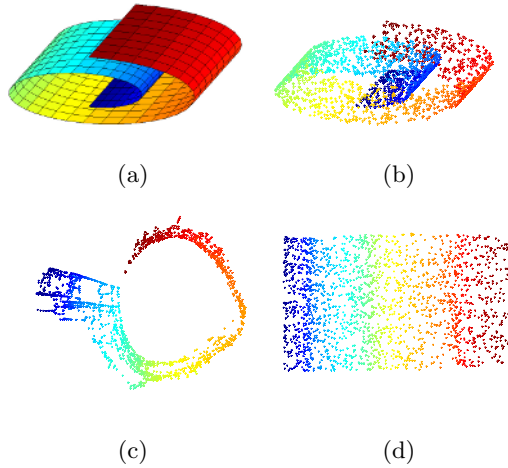
The rest of the paper is organized as follows: Section 2 reviews the estimation of intrinsic dimensionality and motivates its use in estimating local tangent space orientation. Section 3 discusses our technique for estimating the orientation of local tangent space and compares it against previous work in the field. Section 4 outlines our proposed neighborhood selection technique. In Section 5 we present experimental results on datasets and finally Section 6 concludes with a discussion.

## 2 Intrinsic Dimensionality

The intrinsic dimensionality of a data set is commonly defined as the smallest number of dimensions that can be used to adequately explain the data. What constitutes an adequate explanation is subjective and depends on the user and the application. Nevertheless, intrinsic dimensionality is key in dimensionality reduction, since knowledge of the intrinsic dimensionality in every part of the manifold eliminates over- or underfitting. For a complete treatment on the subject of dimensionality see [11].

Wang et al. [12] propose an adaptive neighborhood selection heuristic based on estimates of local tangent orientation, and apply it to a variation of LLE they call Local Tangent Space Alignment (LTSA). Their proposed technique assumes fixed intrinsic dimensionality everywhere that is equal to the target dimensionality specified by the transformation, and uses user-specified parameters to threshold the projection of points in the neighborhood onto the complement and tangent spaces. However, in some applications, it is convenient to separate these two variables,





**Fig. 1.** (a). The classic ‘Swiss roll’ manifold stretched to aspect ratio 0.4. Unlike the more common circular Swiss roll, curvature varies along the manifold. (b). A uniform-density sample drawn from the manifold,  $N = 2000$ . (c). The best embedding computed by Isomap. A setting of  $k = 4$  was used. Higher values result in more shortcuts created from data points to different parts of the manifold and more convoluted embeddings, while  $k < 3$  results in several disconnected components, with no unifying global structure. (d). The projection obtained by Isomap using our adaptive neighborhood technique successfully unrolls the Swiss roll into a flat surface.

e.g., visualization, where target dimensionality is typically limited to 2 or 3. We therefore use target dimensionality in the MDS step to produce the final embedding, but intrinsic dimensionality when estimating the local geometry at a point.

Several techniques have been proposed to estimate intrinsic dimensionality from data in problems where it is unknown. Techniques that apply PCA (globally or locally) and threshold the resulting eigenvalues include [14]. Geometric methods include Costa et al. [15] - an estimator based on the length of minimal spanning trees on graph geodesics. We have found a maximum likelihood estimator recently proposed by Levina and Bickel [16] to work well on our data. The technique assumes constant density in small neighborhoods and approximates the number of samples in hyperspheres of growing radius as a Poisson process. Then the rate of the process  $\lambda(t)$  at intrinsic dimensionality  $m$  can be expressed as,

$$\lambda(t) = \frac{f(x)\pi^{m/2}mt^{m-1}}{\Gamma(m/2 + 1)} \tag{1}$$

where  $f(x)$  is the sampling density and  $\Gamma(\cdot)$  is the Gamma function. A maximum likelihood estimate of the intrinsic dimensionality at point  $\mathbf{x}_i$  given  $c$  neighboring observations is then (see [16] for complete details),

$$\hat{m}_c(\mathbf{x}_i) = \left[ \frac{1}{c-1} \sum_{j=1}^{c-1} \log \frac{T_c(\mathbf{x}_i)}{T_j(\mathbf{x}_i)} \right]^{-1} \tag{2}$$

where  $T_j(\mathbf{x}_i)$  represents the distance from  $\mathbf{x}_i$  to its  $j$ 'th nearest observation. The authors propose averaging over all points to obtain the global intrinsic dimensionality. An optimal  $c$  can be obtained by minimizing the estimator's standard deviation over different sample sizes drawn from the data. The resulting estimator is asymptotically unbiased (negatively biased otherwise) but enjoys remarkably low variance, making it perhaps possible to apply it semi-locally or in clusters.

### 3 Estimating the Local Tangent Space

In this section we outline our technique for estimating local tangent space. Using knowledge or an estimate of the intrinsic dimensionality  $m$ , we seek to estimate the orientation of the manifold at each data point  $\mathbf{x}_i$  and compute an orthonormal basis  $\mathbf{A}_i$  for it. While here we use a global value for  $m$ , local values can be used if a reliable estimator exists.

Hyperplanes (if linearity is assumed) through a neighborhood may be fitted by retaining the vectors corresponding to the highest singular values (up to the desired dimensionality) of the singular value decomposition (SVD) of  $[\mathbf{x}_{i_1}, \dots, \mathbf{x}_{i_k}] - \mathbf{x}_i \mathbf{1}^T$  where  $\mathbf{x}_{i_j}$  are the neighbors of  $\mathbf{x}_i$  and  $\mathbf{1} = [1, \dots, 1]^T$ . Previous approaches that explicitly compute local tangent orientation include Medioni et al. [10], which estimates intrinsic dimensionality and orientation simultaneously at each point, by performing an eigen-decomposition at each data point, and retaining the largest eigenvectors up to the largest drop in the eigenvalues. Although a voting scheme is used to improve the estimator's variance, the resulting intrinsic dimensionality estimates are still too noisy for adaptive neighborhood selection. Wang et al. [12] compute a least squares fit about the mean of the neighborhood rather than  $\mathbf{x}_i$ , but the neighborhood size is indirectly controlled by several user-specified parameters.

Our technique also uses SVD to compute tangent orientation, but the neighbors used in the computation are selected based on estimated local sampling density, an approach inspired by [16]. Under ideal sampling conditions,  $m + 1$  points define an  $m$ -dimensional hyperplane. However, in practice, degenerate configurations are often observed where the points are not in general position, and thus define a rank-deficient space (e.g., collinearity). This is a likely scenario if marginal densities vary along different axes. If the singular values  $\lambda_j$  resulting from the decomposition are ordered in non-increasing order, such that  $\lambda_1 \geq \dots \geq \lambda_m \dots \geq \lambda_k$ , in order to ensure a sound  $m$ -dimensional basis,  $\lambda_m$  must be sufficiently high and  $\lambda_{m+1}$  low. Specifically, singular value  $\lambda_m$  must be significant enough so that it represents an observation that lies in a direction orthogonal to the directions represented by singular values  $\lambda_1 \dots \lambda_{m-1}$ , rather than leftovers from projections of other neighbors. Therefore, an appropriate threshold for  $\lambda_m$  is the expected radius to a neighboring point at  $\mathbf{x}_i$ , denoted  $\tilde{T}_1(\mathbf{x}_i)$ .

A coarse estimate of this radius may be obtained by taking the distance from  $\mathbf{x}_i$  to its nearest neighbor. However, this estimate is unreliable as it is based on only one observation. A better strategy is to infer the radius from a further

neighbor taking advantage of the robust properties of order statistics. If the volume of an  $m$ -dimensional hypersphere of radius  $r$  is  $\pi^{m/2}r^m[\Gamma(\frac{m}{2} + 1)]^{-1}$ , then the expected number of observations  $N(r)$  in the hypersphere follows,

$$E[N(r)] \propto r^m \tag{3}$$

and

$$T_{N(r)}(\mathbf{x}_i) \approx r \tag{4}$$

combining Eq. (3) and (4), we obtain,

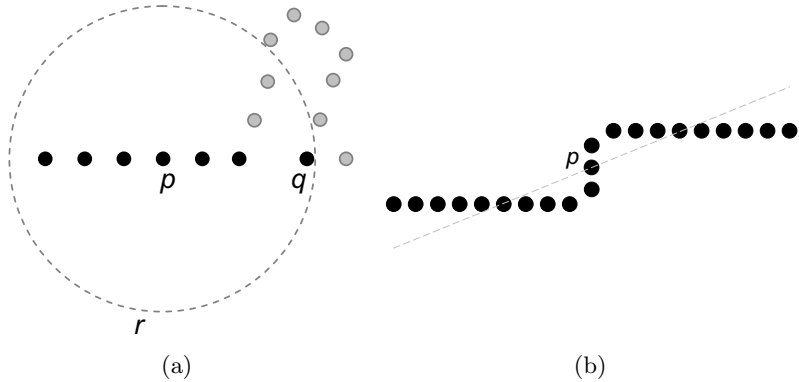
$$\tilde{T}_1(\mathbf{x}_i) = (1/k)^{(1/m)}T_k(\mathbf{x}_i) = (1/k)^{(1/m)}|\mathbf{x}_{i_k} - \mathbf{x}_i|_2 \tag{5}$$

We iteratively increase  $k$ , generating a new estimate  $\tilde{T}_1(\mathbf{x}_i)$  of the expected radius at  $\mathbf{x}_i$  in each iteration according to Eq. (5), until  $\lambda_m \geq \tilde{T}_1(\mathbf{x}_i)$ . Upon termination, basis  $\mathbf{A}_i$  is constructed from the vectors corresponding to the  $m$  highest singular values. This basis defines the estimated tangent space at  $\mathbf{x}_i$  and is used in the next step of our algorithm to select neighbors that are consistent with it.

## 4 Selection of Neighbors

Using the basis  $\mathbf{A}_i$  that defines the estimated tangent orientation at point  $\mathbf{x}_i$ , a neighborhood can be selected that includes nearby points that agree with the computed tangent. Wang et al. [12] examine the ratio of the matrix (Frobenius) norms of the projections of the points under consideration and compare it to a user-specified threshold  $\eta$ . While one global threshold can lead to adaptive neighborhood selection according to the local curvature and density at each point, it is unclear how  $\eta$  should be set. However, the optimal  $\eta$  probably depends on the intrinsic dimensionality since it determines the dimensionality of the projections (and hence their norms). Another anomaly is that the lowest possible ratio of norms (zero) is realized when exactly  $m$  neighbors are considered. To overcome this hitch, the neighborhood is initialized to a ‘sufficiently large’  $K$  (another user-specified parameter) and iteratively shrunk until the specified  $\eta$  ratio is reached. If this step fails, a neighborhood is selected such that the ratio is minimized. Then an expansion step is performed where points that were discarded in the previous step are added back as neighbors if their projection norm ratio satisfies the user-specified threshold  $\eta$ , while ‘skipping’ nearer neighbors whose ratios do not. This is perhaps done to accommodate noise, but as Figure 2(a) demonstrates, it is a dangerous strategy in techniques like Isomap, since it can potentially result in invalid shortcuts to different parts of the manifold and these are likely to have adverse global effects. Another pitfall is that ‘steps’ in the manifold may be smoothed out, as depicted in Figure 2(b). If the projection ratio at the initial  $K$  falls below the user-specified  $\eta$ , the algorithm is trapped in a local minimum and fails to uncover the correct orientation.

In contrast, our strategy for selecting neighbors at a point  $\mathbf{x}_i$  is a direct extension of our approach for estimating the tangent space, outlined in the previous



**Fig. 2.** (a). The neighbors (depicted as dark points) of point  $p$  defined according to [12]. Point  $q$  is considered a neighbor while closer points are excluded, resulting in an invalid shortcut from  $p$  to  $q$ . (b). Estimation of the tangent space at point  $p$  using [12]. Setting the initial  $K$  to a high value such that all the points in the figure are included results in a fit that appears good (high tangent space projection norm relative to complement space projection), but an incorrect tangent. The neighborhood contraction step in [12] is trapped in a local minimum and fails to uncover the correct vertical tangent.

section. We incrementally grow the neighborhood of  $\mathbf{x}_i$  one point at a time, monitoring each new point’s projection onto the complement space at  $\mathbf{x}_i$ , and testing the resulting norm against our estimate of radius to nearest neighbor  $\tilde{T}_1(\mathbf{x}_i)$ . New neighbors  $\mathbf{x}_{i_j}$  are added iteratively until

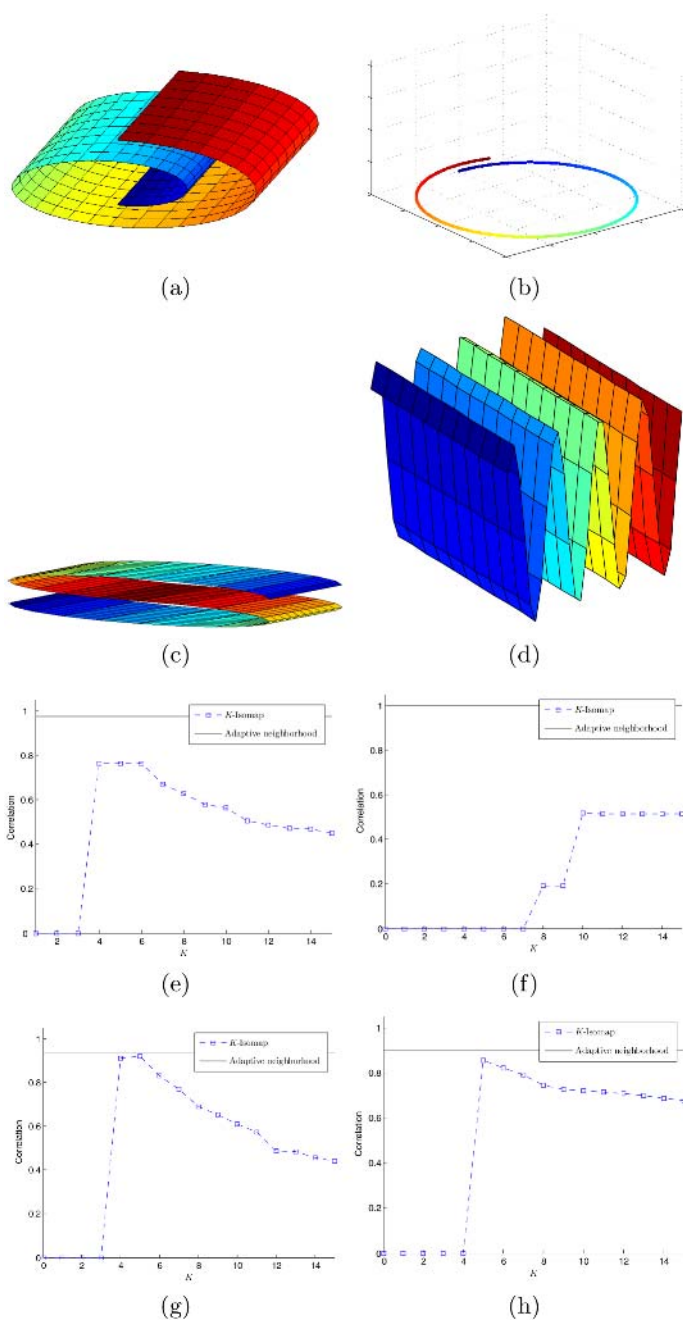
$$|(I - \mathbf{A}_i \mathbf{A}_i^T)(\mathbf{x}_{i_j} - \mathbf{x}_i)|_2 < \tilde{T}_1(\mathbf{x}_i) \tag{6}$$

or equivalently,

$$\sqrt{|\mathbf{x}_{i_j} - \mathbf{x}_i|_2^2 - |\mathbf{A}_i(\mathbf{x}_{i_j} - \mathbf{x}_i)|_2^2} < \tilde{T}_1(\mathbf{x}_i) \tag{7}$$

is violated. To avoid improper shortcutting as illustrated in Figure 2(a), the iteration terminates when a neighboring point breaks the above condition. This process can be viewed as the inclusion of points within a hypercylinder of radius  $\tilde{T}_1(\mathbf{x}_i)$  about the estimated tangent space defined by the basis  $\mathbf{A}_i$ . The estimate  $\tilde{T}_1(\mathbf{x}_i)$  may be further refined at each iteration as new neighbors are added, according to the criterion in Eq. (5).

In contrast to [12], our technique may add an unlimited number of neighbors as long as the linear tangent space assumption is upheld. In linear sections of the manifold, all points are added as neighbors. This is a desirable property for Isomap, since in planar regions, geodesic distances are now correctly estimated as Euclidean distances (whereas normally the graph geodesic significantly over-estimates distances). In fact, if the entire input manifold is linear, all geodesic distances are estimated as Euclidean distances, and Isomap degenerates into PCA.



**Fig. 3.** (a)-(d). Synthetic 2-D and 1-D manifolds embedded in 3-D. (e)-(h). Corresponding plots of the correlation between true geodesic distances and Isomap estimates using  $k$ -Isomap (*dashed line*) starting with the lowest value of  $k$  that yields a global mapping and our adaptive technique (*solid line*).

## 5 Experimental Results

We have tested our technique with the Isomap algorithm on several datasets. Our algorithm's performance on manifolds with relatively constant curvature and uniform sampling (the Swiss roll and S-curve) matched that of Isomap with manually-selected optimal values of  $k$ . Figure 3 depicts the correlations between true and estimated geodesic distances on the stretched Swiss roll and a 3-D spiral for different values of  $k$  and using our adaptive technique. For these structures, Isomap failed to compute satisfactory embeddings with any settings of the algorithm's parameters. On the other hand, our technique adaptively selected neighbors at each point resulting in superior interpolation in relatively flat surfaces while avoiding invalid shortcuts between different parts of the manifold. Here we report correlation to true distances as an objective qualitative measure. However, qualitatively, even small differences in correlation values translate into dramatic effects in terms of the resulting embedding. For example, the embedding produced by  $k$ -Isomap for the stretched Swiss roll in Figure 3(a) ( $k = 4$ , correlation=0.78) can be seen in Figure 1(c). As a sanity check, we also ran our algorithm on the Isomap face database (698 images of synthetic faces under varying illumination and pose). Our adaptive technique appears to produce a satisfactory embedding, but since the true manifold is unknown, quantitative analysis is not possible.

## 6 Summary

We have presented a parameterless adaptive technique for selecting a neighborhood at each point in nonlinear manifold learning, in particular Isomap. To date, nonlinear manifold learning techniques have relied on user-specified parameters that cannot be set in a principled way. Additionally, the use of one global setting results in suboptimal learning if curvature or density vary. Our technique eliminates the guesswork associated with tuning these parameters and enables modeling of manifolds that cannot be modeled effectively with one global setting. In addition to eliminating user-input parameters, our technique offers several advantages over previous work on adaptive selection.

We have demonstrated the effectiveness of the technique on several simulated and real datasets. The technique produces good results on our data and we are currently investigating several possible extensions. In its present form, the technique assumes that observations are sampled directly from the manifold with no noise. We are currently looking at ways to incorporate a noise model, as well as robust voting schemes to improve tangent space and neighborhood estimation at each point.

**Acknowledgments.** The authors thank Konstantinos Derpanis and Erich Leung for reviewing this manuscript and providing useful comments.

## References

1. Jolliffe, I.T.: *Principal Component Analysis*. Springer-Verlag, New York (1986)
2. Duda, R.O., Hart, P.E., Stork, D.G.: *Pattern Classification*. John Wiley and Sons, Inc., New York (2000)
3. Schölkopf, B., Smola, A., Mller, K.R.: Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation* **10**(5) (1998) 1299–1319
4. Tenenbaum, J.B., de Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. *Science* **290**(5500) (2000) 2319–2323
5. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. *Science* **290**(5500) (2000) 2323–2326
6. Belkin, M., Niyogi, P.: Laplacian eigenmaps and spectral techniques for embedding and clustering. In Dietterich, T.G., Becker, S., Ghahramani, Z., eds.: *Advances in Neural Information Processing Systems*. Number 14, Cambridge, MA, MIT Press (2002)
7. Donoho, D.L., Grimes, C.E.: Hessian eigenmaps: locally linear embedding techniques for high-dimensional data. *Proceedings of the National Academy of Arts and Sciences* **100** (2003) 5591–5596
8. Abraham, R., Marsden, J.E.: *Foundations of Mechanics*. Second edn. Addison-Wesley (1978)
9. de Silva, V., Tenenbaum, J.B.: Global versus local methods in nonlinear dimensionality reduction. In S. Becker, S.T., Obermayer, K., eds.: *Advances in Neural Information Processing Systems* 15. (MIT Press)
10. Medioni, G., Lee, M.S., Tang, C.K.: *Computational Framework for Segmentation and Grouping*. Elsevier Science Inc., New York, NY, USA (2000)
11. Falconer, K.: *Fractal Geometry: Mathematical Foundations and Applications*. John Wiley & Sons (1990)
12. Wang, J., Zhang, Z., Zha, H.: Adaptive manifold learning. In: *NIPS*. (2004)
13. Lorenz, E.: Deterministic nonperiodic flow. **20** (1963) 130–141
14. Fukunaga, K., Olsen, D.: An algorithm for finding intrinsic dimensionality of data. *IEEE Transactions on Computer* **20**(2) (1971) 176–183
15. Costa, J., Girotra, A., Hero, A.O.: Estimating local intrinsic dimension with k-nearest neighbor graphs. In: *IEEE Workshop on Statistical Signal Processing (SSP)*, Bordeaux (2005)
16. Levina, E., Bickel, P.J.: Maximum likelihood estimation of intrinsic dimension. In Saul, L.K., Weiss, Y., Bottou, l., eds.: *Advances in Neural Information Processing Systems* 17. MIT Press, Cambridge, MA (2005) 777–784

# Efficient Algorithms for Similarity Measures over Sequential Data: A Look Beyond Kernels

Konrad Rieck<sup>1</sup>, Pavel Laskov<sup>1</sup>, and Klaus-Robert Müller<sup>1,2</sup>

<sup>1</sup> Fraunhofer FIRST.IDA, Kekuléstraße 7, 12489 Berlin, Germany

<sup>2</sup> University of Potsdam, Am Neuen Palais 10, 14469 Potsdam, Germany  
{rieck, laskov, klaus}@first.fhg.de

**Abstract.** Kernel functions as similarity measures for sequential data have been extensively studied in previous research. This contribution addresses the efficient computation of distance functions and similarity coefficients for sequential data. Two proposed algorithms utilize different data structures for efficient computation and yield a runtime linear in the sequence length. Experiments on network data for intrusion detection suggest the importance of distances and even non-metric similarity measures for sequential data.

## 1 Introduction

Sequences are a common non-vectorial data representation used in various machine learning and pattern recognition applications, e.g. textual documents in information retrieval, DNA sequences in bioinformatics or packet payloads in intrusion detection. An essential procedure for analysis of such data is the efficient computation of pairwise similarity between sequences.

Beside specialized string distances [e.g. 1, 2] a large class of similarity measures for sequential data can be defined over contained subsequences by embedding them in a high-dimensional feature space. Previous research focused on computation of kernel functions in such feature spaces. For example, the inner-product over  $n$ -gram or word frequencies has been widely used for analysis of textual documents [e.g. 3, 4, 5] or host-based intrusion detection [e.g. 6]. The challenge of uncovering information in DNA has influenced further advancement of kernel functions, e.g. by exploring different sets of subsequences [e.g. 7, 8, 9, 10] or incorporating mismatches, gaps and wildcards [e.g. 11, 12, 13].

There exist, however, a large amount of learning algorithms which are not directly suitable for kernel functions. In principle, any inner-product induces a Euclidean distance in feature space [14], yet the richness of content in sequential data and the variability of its characteristics in feature spaces motivate application of other distance functions.

A general technique for computation of similarity measures suitable for kernels, distances and similarity coefficients is proposed in this contribution. It is based on incremental accumulation of matches and mismatches between subsequences comprising a feature space. Two algorithms are presented that utilize



different data structures for efficient computation: hash tables and tries. Both algorithms have linear runtime complexity in terms of sequence lengths.

The rest of the paper is organized as follows: Section 2 defines several similarity measures for sequential data including kernels, distances and similarity coefficients. Comparison algorithms and corresponding data structures are introduced in Section 3. Finally, experiments in Section 4 compare the efficiency of the introduced algorithms and illustrate their application in network intrusion detection.

## 2 Similarity Measures for Sequential Data

Given an alphabet  $\Sigma$  of size  $N$ , a sequence  $x$  is defined as a concatenation of symbols from  $\Sigma$ . The content of a sequence can be modeled as a set of possibly overlapping subsequences  $w$  taken from a finite language  $L \subset \Sigma^*$ . We refer to these extracted subsequences as *words*. The language  $L$  constitutes the basis for calculating similarity of sequences and typically corresponds to a bag of characters, words or  $n$ -grams. Given a sequence  $x$  and a language  $L$ , an embedding into feature space is performed by calculating  $\phi_w(x)$  for every  $w \in L$  appearing in  $x$ . Usually the function  $\phi_w(x)$  returns the frequency of  $w$  in  $x$ , however, other definitions returning a count or a binary flag for  $w$  are possible. Furthermore we define  $l$  to be the length of  $x$ .

We assume that the total length of words in every sequence  $x$  is proportional to  $l$ . This assumption is valid, for example, for  $n$ -grams of fixed length  $n$  and non-overlapping words, and ensures linear runtime of the proposed algorithms. In context of kernels several approaches have been investigated that do not make such an assumption [e.g. 9, 10, 11, 12, 13], however, some of them come at a cost of super-linear complexity.

By utilizing the feature space induced through  $\phi$ , one can adapt classical kernel and distance functions to operate on sequences. Table 1 lists kernel functions and Table 2 distance functions that are implemented using the algorithms presented in Section 3.

Yet another way of measuring similarity are so called similarity coefficients [e.g. 15, 16]. They are non-metric and have been primarily used on binary data.

**Table 1.** Kernel functions for sequential data

Kernel function	$k(x, y)$
Linear	$\sum_{w \in L} \phi_w(x) \phi_w(y)$
Polynomial	$\left( \sum_{w \in L} \phi_w(x) \phi_w(y) + \theta \right)^d$
RBF	$\exp \left( \frac{-d(x, y)^2}{\sigma} \right)$

**Table 2.** Distance functions for sequential data

Distance function	$d(x, y)$
Manhattan	$\sum_{w \in L}  \phi_w(x) - \phi_w(y) $
Canberra	$\sum_{w \in L} \frac{ \phi_w(x) - \phi_w(y) }{\phi_w(x) + \phi_w(y)}$
Minkowski	$\sqrt[k]{\sum_{w \in L}  \phi_w(x) - \phi_w(y) ^k}$
Chebyshev	$\max_{w \in L}  \phi_w(x) - \phi_w(y) $

**Table 3.** Similarity coefficients for sequential objects

Similarity coefficients	$s(x, y)$
Jaccard	$\frac{a}{a + b + c}$
Czekanowski	$\frac{2a}{2a + b + c}$
Sokal-Sneath	$\frac{a}{a + 2(b + c)}$
Kulszynski	$\frac{1}{2} \left( \frac{a}{a + b} + \frac{a}{a + c} \right)$

Similarity coefficients are constructed using three summation variables  $a, b$  and  $c$ . The variable  $a$  contains the number of positive matches (1-1),  $b$  the number of left mismatches (0-1) and  $c$  the number of right mismatches (1-0). The most common similarity coefficients are given in Table 3.

Similarity coefficients can be extended to non-binary data by modification of the summation variables. The degree of match for a word  $w \in L$  can be defined as  $\min(\phi_w(x), \phi_w(y))$  and the respective mismatches are defined as deviations thereof:

$$\begin{aligned}
 a &= \sum_{w \in L} \min(\phi_w(x), \phi_w(y)) \\
 b &= \sum_{w \in L} [\phi_w(x) - \min(\phi_w(x), \phi_w(y))] \\
 c &= \sum_{w \in L} [\phi_w(y) - \min(\phi_w(x), \phi_w(y))]
 \end{aligned}$$

### 3 Algorithms and Data Structures

In order to calculate the presented kernels, distances and similarity coefficients, one needs to establish a general model of similarity measures for sequential data.

**Table 4.** Generalized formulations of distances

Distance	$\oplus$	$m^+(p, q)$	$m_x^-(p)$	$m_y^-(q)$
Manhattan	+	$ p - q $	$p$	$q$
Canberra	+	$ p - q /(p + q)$	1	1
Minkowski <sup>k</sup>	+	$ p - q ^k$	$p^k$	$q^k$
Chebyshev	max	$ p - q $	$p$	$q$

**Table 5.** Generalized formulations of summation variables

Variable	$\oplus$	$m^+(p, q)$	$m_x^-(p)$	$m_y^-(q)$
$a$	+	$\min(p, q)$	0	0
$b$	+	$p - \min(p, q)$	$p$	0
$c$	+	$q - \min(p, q)$	0	$q$

A key instrument for computation of kernel functions is finding words  $w \in L$  present in two sequences  $x$  and  $y$  – we refer to these words as *matches*. For distances and similarity coefficients, we also need to consider words  $w \in L$  present in  $x$  but not in  $y$  (and vice versa) – we refer to these words as *mismatches*<sup>1</sup>.

Furthermore we introduce an outer function  $\oplus$  which corresponds to the global aggregation performed in many similarity measures, e.g. the summation in various kernel and distance functions. Given these definitions, we can express a generic similarity measure  $s$  as

$$s(x, y) = \bigoplus_{w \in L} m(x, y, w) \quad (1)$$

$$m(x, y, w) = \begin{cases} m^+(\phi_w(x), \phi_w(y)) & \text{if } w \text{ is a match} \\ m_x^-(\phi_w(x)) & \text{if } w \text{ is a mismatch in } x \\ m_y^-(\phi_w(y)) & \text{if } w \text{ is a mismatch in } y \end{cases} \quad (2)$$

We can now reformulate the set of distances given in Table 2 using the functions  $\oplus$ ,  $m^+$ ,  $m_x^-$  and  $m_y^-$ . The generalized formulations of some distances are presented in Table 4.

Adapting similarity coefficients to such a generic representation is even simpler, since only the three summation variables  $a$ ,  $b$  and  $c$  need to be reformulated, as shown in Table 5.

### 3.1 Hash-Based Sequence Comparison

The classical scheme for computation of similarity measures over sequences utilizes indexed tables, or in the more general case hash tables [e.g. 4]. The words extracted from a sequence and corresponding frequencies or counts are stored in

<sup>1</sup> The term “mismatch” herein corresponds to two sequences being unequal and not, as often used in bioinformatics, to inexact matching of sequences.

the bins of a hash table. Figure 1(a) shows two hash tables carrying the words {"bar", "barn", "card"} and {"car", "bank", "band", "card"} with corresponding counts.

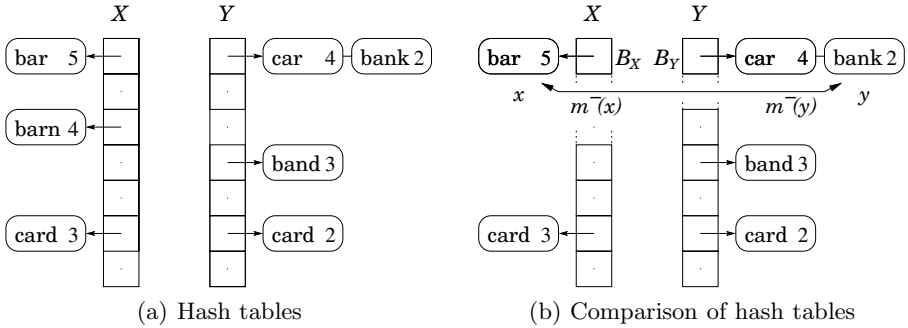


Fig. 1. Hash table data structures (a) and their comparison (Case 2) (b)

Algorithm 1 defines the comparison of two hash tables  $X$  and  $Y$  with fixed size  $M$ . The algorithm proceeds by looping over all  $M$  bins, checking for matching (cf. Algorithm 1: Case 1) and mismatching words (cf. Algorithm 1: Case 2 & 3). Figure 1(b) illustrates this process at the mismatches "bar" and "bank" which are stored in corresponding bins.

---

**Algorithm 1.** Hash-based Sequence Comparison

---

```

1: function COMPARE( $X, Y$ )
2:    $s \leftarrow 0$ 
3:   for  $i \leftarrow 1, M$  do
4:      $B_X \leftarrow \text{bins}[X, i]$ 
5:      $B_Y \leftarrow \text{bins}[Y, i]$ 
6:     if  $B_X \neq \text{NIL}$  and  $B_Y \neq \text{NIL}$  then
7:       for all  $x \in B_X$  and  $y \in B_Y$  do
8:         if  $x = y$  then
9:            $s \leftarrow s \oplus m^+(x, y)$  ▷ Case 1
10:        else
11:           $s \leftarrow s \oplus m^-(x) \oplus m^-(y)$  ▷ Case 2
12:        else if  $B_X \neq \text{NIL}$  then
13:          for all  $x \in B_X$  do
14:             $s \leftarrow s \oplus m^-(x)$  ▷ Case 3
15:        else if  $B_Y \neq \text{NIL}$  then
16:          for all  $y \in B_Y$  do
17:             $s \leftarrow s \oplus m^-(y)$  ▷ Case 3
18:   return  $s$ 

```

---

Since the size of the hash tables is fixed at  $M$ , the average runtime for a comparison is  $\Theta(M)$ . To avoid possible hash collisions, a high value of  $M \gg l$  must be chosen in advance, otherwise the chaining of bins (Case 2) results in  $O(l^2)$  worst-case runtime for  $O(l)$  extracted words per sequence.

### 3.2 Trie-based Sequence Comparison

A trie is an  $N$ -ary tree, whose nodes are  $N$ -place vectors with components corresponding to the elements of  $\Sigma$  [17]. Figure 2(a) shows two tries  $X$  and  $Y$  containing the same words as the hash tables in Figure 1(a). The nodes of the tries are augmented to carry a variable reflecting the count of the passing sequence. The end of each extracted word is indicated by a marked circle. Application of tries to computation of kernel functions has been considered by [18].

Depending on the applied similarity measure the trie nodes can be extended to store other aggregated values which speed up calculations involving subtrees, e.g. for the Minkowski distance  $\sum_w \phi_w(x)^k$  for all lower words  $w$ ,

Comparison of two tries can be carried out as in Algorithm 2: Starting at the root nodes, one traverses both tries in parallel, processing matching and mismatching nodes. If the traversal passes two equal and marked nodes, a matching word is discovered (Case 1), if only one node is marked a mismatch occurred (Case 2). The recursive traversal is stopped if two nodes do not match, and thus two sets of underlying mismatching words are discovered (Case 3). Figure 2(b) shows a snapshot of such a traversal. The nodes  $x$  and  $y$  are not equal, and the words {"bar", "barn"} and {"band", "bank"} constitute mismatches.

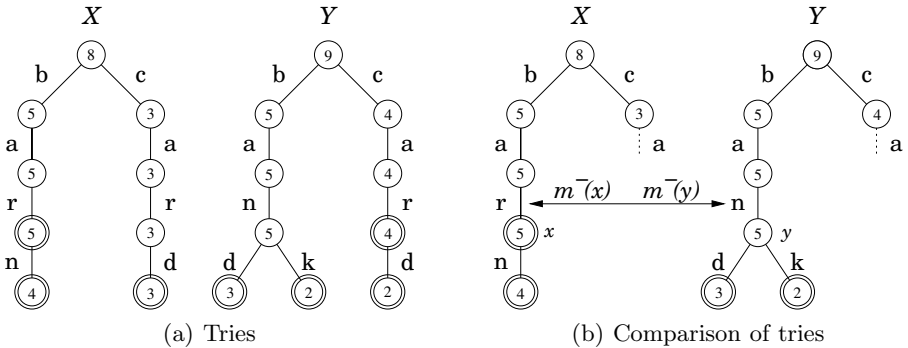


Fig. 2. Trie data structures (a) and their comparison (Case 3) (b)

As an invariant, the nodes under consideration in both tries remain at the same depth and thus the worst-case runtime is  $O(l)$ . An advantage of the trie data structure comes into play especially if the provided alphabet is large and a lot of mismatches occur. The traversal discovers mismatching words after passing the first few symbols and omits further unnecessary comparisons.

**Algorithm 2.** Trie-based Sequence Comparison

---

```

1: function COMPARE( $X, Y$ )
2:    $s \leftarrow 0$ 
3:   for  $i \leftarrow 1, N$  do
4:      $x \leftarrow \text{child}[X, i]$ 
5:      $y \leftarrow \text{child}[Y, i]$ 
6:     if  $x \neq \text{NIL}$  and  $y \neq \text{NIL}$  then
7:       if  $\text{end}[x]$  and  $\text{end}[y]$  then
8:          $s \leftarrow s \oplus m^+(x, y)$  ▷ Case 1
9:       else if  $\text{end}[x]$  then
10:         $s \leftarrow s \oplus m^-(x)$  ▷ Case 2
11:      else if  $\text{end}[y]$  then
12:         $s \leftarrow s \oplus m^-(y)$  ▷ Case 2
13:       $s \leftarrow s \oplus \text{COMPARE}(x, y)$ 
14:    else
15:      if  $x \neq \text{NIL}$  then
16:         $s \leftarrow s \oplus m^-(x)$  ▷ Case 3
17:      if  $y \neq \text{NIL}$  then
18:         $s \leftarrow s \oplus m^-(y)$  ▷ Case 3
19:    return  $s$ 

```

---

## 4 Experimental Results

### 4.1 Efficiency of Data Structures

Efficiency of the two proposed algorithms has been evaluated on four benchmark data sets for sequential data: DNA sequences of the human genome [19], system call traces and connection payloads from the DARPA 1999 data set [20] and news articles from the Reuters-21578 data set [21]. Table 6 gives an overview of the data sets and their specific properties.

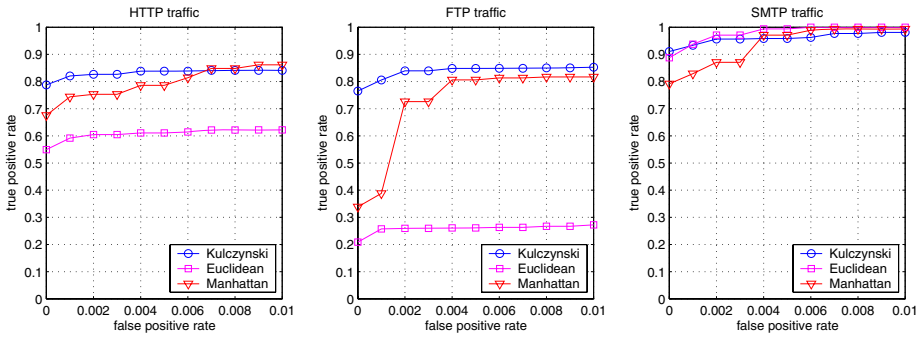
For each data set 100 sequences were randomly drawn and  $n$ -grams of lengths 3, 5 and 7 extracted. The  $n$ -grams of each sequence were stored in tries and hash tables with varying size from  $10^2$  to  $10^5$ . Subsequently the Canberra distance was calculated pairwise over the tries and hash tables using the proposed algorithms, resulting in 5000 comparison operations per setup. The procedure was repeated 10 times and the runtime was averaged over all runs. The experimental results are given in Table 7.

**Table 6.** Datasets of sequential objects

Name	Type	Alphabet	Min. length	Max. length
DNA	Human genome sequences	4	2400	2400
HIDS	BSM system call traces	88	5	129340
NIDS	TCP connection payloads	108	53	132753
TEXT	Reuters Newswire articles	93	43	10002

**Table 7.** Runtime experiments for Canberra distance

Dataset	$n$	Average runtime for 5000 comparisons (s)				
		Trie	Hash ( $10^2$ )	Hash ( $10^3$ )	Hash ( $10^4$ )	Hash ( $10^5$ )
DNA	3	<b>0.19</b>	0.22	0.28	0.66	6.04
	5	<b>2.21</b>	4.46	2.94	3.56	9.57
	7	10.72	37.63	13.02	<b>5.67</b>	9.43
HIDS	3	<b>0.06</b>	0.10	0.13	0.62	3.05
	5	0.15	0.15	0.16	0.66	5.23
	7	0.25	<b>0.19</b>	0.22	0.70	4.15
NIDS	3	<b>0.48</b>	1.70	1.07	1.43	5.12
	5	<b>0.86</b>	3.70	1.72	1.81	5.90
	7	<b>1.20</b>	4.83	2.10	2.42	6.08
TEXT	3	<b>1.12</b>	1.75	1.22	1.63	7.03
	5	1.65	3.85	1.64	1.89	7.58
	7	<b>2.13</b>	5.92	2.19	2.24	7.74

**Fig. 3.** Detection performance for network intrusion detection

The average runtime of the hash-based algorithm strongly depends on the size of the hash table. The optimal value varies for different data sets and values of  $n$ . However, in 10 of 12 cases the trie-based algorithm performs equally well or better than the best hash table setup, being independent of a parameter.

## 4.2 Application: Network Intrusion Detection

To demonstrate the proposed algorithms on realistic data, we conducted an experiment for unsupervised learning in network intrusion detection. The underlying network data was generated by the members of our laboratory using virtual network servers. Recent network attacks were injected by a penetration-testing expert.

A distance-based anomaly detection method [22] was applied on 5-grams extracted from byte sequences of TCP connections using different similarity measures: a linear kernel (Euclidean distance), the Manhattan distance and the

Kulczynski coefficient. Results for the common network protocols HTTP, FTP and SMTP are given in Figure 3.

Application of the Kulczynski coefficient yields the highest detection accuracy. Over 78% of attacks for each protocol are identified with no false-positives. In comparison the Euclidean distances fails to uncover good geometric properties for discrimination of attacks in this particular setup.

## 5 Conclusions

We have shown that, similarly to kernels, a large number of distances and similarity coefficients can be efficiently computed for sequential data. The use of such similarity measures allows one to investigate unusual metrics for application of machine learning in specialized problem domains. As an example, the best results in our experiments on unsupervised learning for network intrusion detection have been obtained with the Kulczynski coefficient over  $n$ -grams of connection payloads. Thus direct application of distances over sequential data may be favorable over implicit use of the Euclidean distance induced by kernels. Especially promising are further applications of the proposed algorithms in computer security and bioinformatics.

## Acknowledgments

The authors gratefully acknowledge the funding from *Bundesministerium für Bildung und Forschung* under the project MIND (FKZ 01-SC40A) and would like to thank Sören Sonnenburg, Julian Laub and Mikio Braun for fruitful discussions and support.

## References

- [1] Hamming, R.W.: Error-detecting and error-correcting codes. *Bell System Technical Journal* **29**(2) (1950) 147–160
- [2] Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions, and reversals. *Doklady Akademii Nauk SSSR* **163**(4) (1964) 845–848
- [3] Salton, G.: Mathematics and information retrieval. *Journal of Documentation* **35**(1) (1979) 1–29
- [4] Damashek, M.: Gauging similarity with  $n$ -grams: Language-independent categorization of text. *Science* **267**(5199) (1995) 843–848
- [5] Joachims, T.: Text categorization with support vector machines: Learning with many relevant features. Technical Report 23, LS VIII, University of Dortmund (1997)
- [6] Eskin, E., Arnold, A., Prerau, M., Portnoy, L., Stolfo, S.: A geometric framework for unsupervised anomaly detection: detecting intrusions in unlabeled data. In: *Applications of Data Mining in Computer Security*. Kluwer (2002)
- [7] Zien, A., Rätsch, G., Mika, S., Schölkopf, B., Lengauer, T., Müller, K.R.: Engineering Support Vector Machine Kernels That Recognize Translation Initiation Sites. *Bioinformatics* **16**(9) (2000) 799–807



- [8] Leslie, C., Eskin, E., Noble, W.: The spectrum kernel: A string kernel for SVM protein classification. In: Proc. Pacific Symp. Biocomputing. (2002) 564–575
- [9] Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., Watkins, C.: Text classification using string kernels. *Journal of Machine Learning Research* **2** (2002) 419–444
- [10] Vishwanathan, S., Smola, A.: Fast Kernels for String and Tree Matching. In: *Kernels and Bioinformatics*. MIT Press (2004) 113–130
- [11] Leslie, C., Eskin, E., Cohen, A., Weston, J., Noble, W.: Mismatch string kernel for discriminative protein classification. *Bioinformatics* **1**(1) (2003) 1–10
- [12] Leslie, C., Kuang, R.: Fast string kernels using inexact matching for protein sequences. *Journal of Machine Learning Research* **5** (2004) 1435–1455
- [13] Rousu, J., Shawe-Taylor, J.: Efficient computation of gapped substring kernels for large alphabets. *Journal of Machine Learning Research* **6** (2005) 1323–1344
- [14] Schölkopf, B.: The kernel trick for distances. In Leen, T., Diettrich, T., Tresp, V., eds.: *Advances in Neural Information Processing Systems 13*, MIT Press (2001)
- [15] Jaccard, P.: Contribution au problème de l’immigration post-glaciaire de la flore alpine. *Bulletin de la Société Vaudoise des Sciences Naturelles* **36** (1900) 87–130
- [16] Anderberg, M.: *Cluster Analysis for Applications*. Academic Press, Inc., New York, NY, USA (1973)
- [17] Knuth, D.: *The art of computer programming*. Volume 3. Addison-Wesley (1973)
- [18] Shawe-Taylor, J., Cristianini, N.: *Kernel methods for pattern analysis*. Cambridge University Press (2004)
- [19] Sonnenburg, S., Zien, A., Rätsch, G.: ARTS: Accurate recognition of transcription starts in human. *Bioinformatics* (2006) submitted.
- [20] Lippmann, R., Haines, J., Fried, D., Korba, J., Das, K.: The 1999 DARPA off-line intrusion detection evaluation. *Computer Networks* **34**(4) (2000) 579–595
- [21] Lewis, D.D.: Reuters-21578 text categorization test collection. AT&T Labs Research (1997)
- [22] Rieck, K., Laskov, P.: Detecting unknown network attacks using language models. In: Proc. DIMVA. (2006) to appear.

# Multi-scale Bayesian Based Horizon Matchings Across Faults in 3d Seismic Data

Fitsum Admasu and Klaus Tönnies

Computer Vision Group, University of Magdeburg, D-39016, Magdeburg, Germany

**Abstract.** Oil and gas exploration decisions are made based on inferences obtained from seismic data interpretation. While 3-d seismic data become widespread and the data-sets get larger, the demand for automation to speed up the seismic interpretation process is increasing as well. Image processing tools such as auto-trackers assist manual interpretation of horizons, seismic events representing boundaries between rock layers. Auto-trackers works to the extent of observed data continuity; they fail to track horizons in areas of discontinuities such as faults.

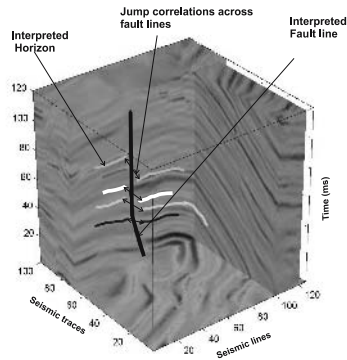
In this paper, we present a method for automatic horizon matching across faults based on a Bayesian approach. A stochastic matching model which integrates 3-d spatial information of seismic data and prior geological knowledge is introduced. A multi-resolution simulated annealing with reversible jump Markov Chain Monte Carlo algorithm is employed to sample from a-posteriori distribution. The multi-resolution is defined in a scale-space like representation using perceptual resolution of the scene. The model was applied to real 3-d seismic data, and has shown to produce horizons matchings which compare well with manually obtained matching references.

## 1 Introduction

Subsurface regions offering prospects for the existence of hydrocarbons undergo a seismic survey to get the profile of their underground structure. An exploration well is drilled to conclusively determine the presence or absence of oil/gas. Since drilling a well is an expensive, high-risk operation, all available information needs to be exploited before arriving at any exploration decision. A seismic survey consists of seismic data acquisition and interpretation. Seismic data are pictures showing subsurface seismic reflectivity. They are acquired by sending artificially created seismic wave signals from a ground surface and recording reflections from underground rock layers. The recorded seismic signal consists of amplitudes of various strength and sign (peaks or troughs). After several preprocessing steps the recorded signals are ready for interpretations.

3-d seismic data consist of numerous closely-spaced seismic lines in three dimensions: seismic lines, seismic traces, and time (see Fig.1). Each seismic line indicates the seismic shoot line and a seismic trace is the area covered by each seismic record. The area resolution ranges from 12.5 m to 25 m. The time dimension is measured in microseconds unit which is the time spent when the signals

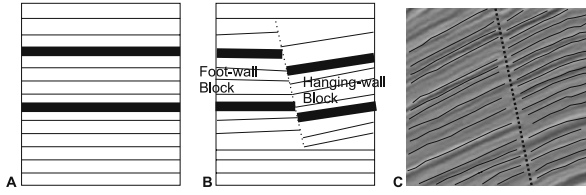
sent from the surface go down and return (Two Way Travel). Usually time sampling in 2ms is done for about 6 seconds resulting in about 3000 time slices. Structural interpretation of seismic data attempts to create 3-d subsurface models and involves the interpretation of faults and horizons [3]. Faults are fractures of rock layers. They are identified by finding the reflected event termination and are usually interpreted as straight lines or connected line segments (see fault lines in Fig.1). Horizons appear as linear structures and indicate interfaces between underground rock layers. Horizon tracking is performed by following continuity of high-amplitudes (see interpreted horizon on Fig.1). The horizon tracking also involves jump correlation of horizons across faults (jump correlation of horizons on Fig.1).



**Fig. 1.** Seismic data with manually interpreted faults and horizon

The research presented here is motivated by the demand for computer-assisted structural interpretation of seismic data. At present most parts of seismic interpretation are done manually. Seismic data have sizes of several gigabytes. Manual structural interpreting on each seismic slice is time consuming and error prone. It takes too long to build trusted models for exploration decisions. Further, manual interpretation results are interpreter-biased and lack well specified reliability measures. Computer-assisted interpretation has the advantage of providing faster interpretation and a consistent workflow.

We concentrate to automate matching of horizons across faults (see Fig.2). We assume that horizon surfaces in unfaulted regions are given (e.g. by applying an autotracker [4]) and fault surfaces have been generated (e.g. by a methodology suggested by [5]). We restrict our work to normal faults which is the most common fault type where one side of the fault block (hangingwall) moves down relative to the other side (footwall). Since the two sides of the fault may have undergone different geological processes, such as compression and erosion, scale differences between horizons on the two sides can be expected and some horizons on one side may not have matches on the other side of the fault. Automating horizon matching is very challenging due to non-dense seismic information, local distortions, and large number of possible configurations.



**Fig. 2.** A. Pre-fault and B. Post-fault configuration of rock layers. C. A seismic slice taken from 3-d data showing horizons curves cut by a fault (dashed line).

### 1.1 Previous Work

The horizon matching problem can not be solved using classical stereo correspondence or registration algorithms due to little intensity information to guide the utilization of optical flow and presence of local distortions. We are aware of only two works, which directly deal with the correlation across a fault. The first is the model-based scheme for matching horizons at normal faults in 2-d seismic images introduced in [2]. Well-defined horizons segments on both sides of the fault were extracted and matched based on local correlation of seismic intensity and geological knowledge. However, a pure 2-d approach lacks efficiency and is suitable only if the information of the 2-d seismic slice is sufficient for evaluation of the geological constraints. The second [1] formulates the horizon matching as a non-rigid continuous point matching between the two sides of the fault. However, it is computationally expensive and not sufficiently robust with respect to noise and artifacts in seismic data.

In this paper, we aim to exploit existing 3-d spatial relationships in the data for robust matching across faults. We introduce a stochastic model, which integrates a data term and prior geological constraints. The data term incorporates seismic observations through statistical measures of local homogeneities in 3-d space. The a-priori geological constraints are modelled through interactions of geometric primitives. The stochastic nature of the model provides quality measures to search the matching solution. We use a multi-resolution search strategy where strong horizons signals give guidance for matching the weaker.

## 2 Data Representation

Seismic data ready for structural interpretation can be represented as 3-d scene  $S_d = (V, F)$ , where  $V \subseteq R^3$  indicates the space covered by the seismic survey and  $F$  is a real-valued scalar field such that  $F : R^3 \rightarrow R$ . For  $c \in V$ ,  $c = (x, y, z)$  denotes a position in seismic data dependant coordinate system and  $x$ ,  $y$ , and  $z$  span resp. the seismic line, seismic trace, and time.  $F$  assigns to  $c$  a seismic amplitude value  $F(c)$ .

Considering geometries of a fault and horizons from both sides of the fault are given in the seismic data set as in Fig.2, another space  $S_p = (V, T)$  is defined such that  $T : V \rightarrow L$  and  $T(c) = l$ , for  $l \in L = \{p, H_{f1}, H_{f2}, \dots, H_{fm}, H_{h1}, H_{h2}, \dots, H_{hn}, 0\}$  where  $p$  is a label for fault plane voxels, and for  $i \in Z^+$ ,

$H_{f_i}$ s and  $H_{h_i}$ s represent respectively labels for horizons surfaces from foot- and hanging-walls blocks (Fig.3). The horizons labels are sorted in ascending order of their time coordinates such that  $0 < H_{f_1} < H_{f_2} < \dots < H_{f_m}$  and  $0 < H_{h_1} < H_{h_2} < \dots < H_{h_n}$ .  $T(c) = 0$  if the voxel  $c$  does not belong any of the primitives, and indicates background.

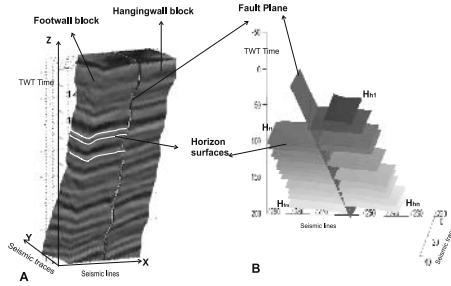


Fig. 3. A. Fault patch seismic section. B. Abstraction of horizons and fault surfaces.

### 3 Problem Representation

Horizons matching problem is to find a set of matching pairs  $x = \{x_1, \dots, x_i, \dots, x_s\} \in X$ , such that  $x_i = (H_{f_{p_i}}, H_{h_{q_i}})$  with  $1 \leq p_i \leq m$  and  $1 \leq q_i \leq n$  joins horizons which would have been continuous had the fault not been present.  $X$  is the search space obtained by permutations and combinations of  $H_f \times H_h$  where  $H_f = \{H_{f_1}, H_{f_2}, \dots, H_{f_m}\}$  and  $H_h = \{H_{h_1}, H_{h_2}, \dots, H_{h_n}\}$ . We refer  $x$  also as marked-point set [8] where  $H_f$  are points and  $H_h$  are marks.

Post-fault configurations imaged by the seismic data do not provide the complete information about the pre-fault configuration of the horizons. Therefore, the geological continuity of horizons across faults are established using additional non-observed geological information.

The matching problem is combinatorial with large search space,  $X$ . The size of  $X$  is estimated as  $|X| \simeq \sum_{d=1}^N |X_d|$  such that  $|X_d| = \binom{N}{d}$  and  $N = |H_f| * |H_h|$ .

### 4 Bayesian Formulation

We use the probability theory to decide the most likely configuration or matching solution. Let  $f_x$  represent the seismic amplitude features for a configuration  $x$ . Then using Bayes theorem, the logical connection between the marked-point set  $x$  and the data  $f_x$  are determined with  $P(x | f_x)$ .

$$P(x | f_x) = \frac{P(x)P(f_x | x)}{P(f_x)} \tag{1}$$

In following subsections, the a-priori model,  $P(x)$ , and data model,  $P(f_x | x)$ , are described.

### 4.1 The A-Priori Model

We consider the geometries of the horizon primitives as uniform Poisson point process. Then the marked-point set  $x$  are from a Gibbs random field and  $P(x)$  forms a Gibbs distribution [7]:

$$P(x) = \frac{1}{Z} \beta^{n(x)} e^{\left(\frac{-U(x)}{T}\right)} \tag{2}$$

where  $U(x)$  is an energy term evaluated by interactions of the geometric primitives,  $Z$  is normalization constant, and  $\beta$  stands for a scale parameter.  $T$  is the temperature of the system.

No priori knowledge regarding the topology of a single horizon can be defined. However, we can impose geological constraints when dealing with a sequence of horizon layers. These constraints are hard constraints ( $C_1$  and  $C_2$ ) and soft constraint ( $C_3$ ).  $C_1$  states that horizons must not cross each other, and is set to  $\infty$  if horizons cross; otherwise to 0.  $C_2$  states that offsets have only one direction, and is set to  $\infty$  if this constraint is violated; otherwise  $C_2$  is 0.  $C_3$  is imposed using a heuristically determined theoretical fault throw function. Fault throw is the vertical offset of the displaced horizons (Fig.4A). Fault throw is maximum at the mid of the fault surface and decreases to zero towards the tips of the fault surface. The fault plane may be approximated by an elliptic shape according to [10] (see Fig.4B). Then, the fault throw value at a given point on the fault surface may be estimated by a function  $f_d$  as follows:

$$f_d(r) = 2D(((1+r)/2)^2 - r^2)^2(1-r) \tag{3}$$

where  $r$  is the normalized radial distance from the fault center and  $D$  is the maximum fault throw value. Horizons offsets of each matching pairs are represented with a set  $D_x = \{d_1, d_2, \dots, d_N\}$  where  $d_i$  represents the offsets of the hanging-wall horizon in the matched pair  $x_i$ . Correspondingly, expected fault offsets  $U_x = \{u_1, u_2, \dots, u_N\}$  for the matching pairs are estimated using equation 3. The fault model contains unknown latent values,  $\Theta$  (fault width, length, center and maximum fault throw). Considering the gaussian mixture and applying the expectation maximization algorithm, the  $U_x$  and  $\Theta_{max}$  are estimated. Then

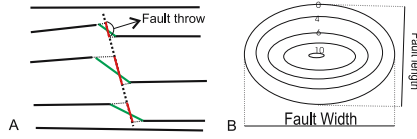
$$C_3(x) = \log\left(\sum_{j=1}^N P_{\Theta_{max}}(d_j | u_{\Theta_{max}})\right) \tag{4}$$

Finally a-priori energy,  $U(x)$ , in equation 2 is estimated as

$$U(x) = C_1(x) + C_2(x) + C_3(x) \tag{5}$$

### 4.2 Data Model

Horizons represent topological surfaces of homogenous rock layers. This homogeneity is usually reflected in the seismic data as similar amplitude values. Although this similarity does not mean identical amplitudes or iso-surfaces, a characterization can be derived which capture the local strong seismic similarity expected on horizon topological surfaces. Previous implicit horizon models utilized



**Fig. 4.** A. Matched horizons with their fault throws shown on bold on the dashed fault line. B. Fault throw contours on a fault surface have ideally elliptical shapes.

in auto-picking tools [4] and other horizon matching algorithms [2] [1] do not capture any spatial variability of horizon seismic signals. Here we introduce a novel explicit modelling of 3-d horizon surfaces for automation purpose.

The seismic signals observed on the topological surfaces of matched horizons with  $t$  are modelled as gaussian random process,  $h(t)$  where  $t \in V, h(t) = F(t)$  and  $t$  represents the voxels of the joined horizons surfaces. Then, the gaussian process is specified by mean  $\mu(t) = E(h(t))$  and its covariance function  $cov(t, t') = E[(h(t) - \mu(t))(h(t') - \mu(t'))]$ ,  $t' \in V$ . The gaussian model is selected for its flexibility and easiness to impose the priori knowledge of seismic similarity.

Ignoring the fault offsets, the spatial correlation of  $h(t)$  is estimated as

$$\gamma(d) = E\{h(t+d) - h(t)\}^2. \tag{6}$$

for a lag distance  $d$ . This expectation is estimated as the average squared difference of values separated by  $d$ .

$$\gamma(d) \simeq \frac{1}{N(d)} \sum_{N(d)} [h(t) - h(t+d)]^2 \tag{7}$$

where  $N(d)$  is the number of pairs for lag  $d$ . Then for a marked set  $x$  with  $t_x$  representing the joined horizons' voxels, the likelihood probability  $P(f_x | x)$  in equation 1 is estimated as

$$P(f_x | x) = P(h(t_x)) \propto \prod_{x_i} e^{-(\sum_d \gamma(d))} \tag{8}$$

## 5 Searching for Matching Solution

The optimal matching solution is a marked point set  $x_{max} = (x_1, \dots, x_i, \dots, x_s)$  that maximizes  $P(x | f_x)$  where

$$x_{max} = \arg \max_x P(x | f_x) \tag{9}$$

Using equation 8 and 2

$$x_{max} = \arg \min_x [U(x) + \sum_{x_i} (\sum_d \gamma(d))] \tag{10}$$

This is a MAP estimation problem and  $x_{max}$  is searched in space  $X$ . Since  $X$  is too big for a direct search, simulated annealing with a Reverse Jump Markov

Chain Monte Carlo (RJCMCMC) [6] sampling method is utilized. We introduce another random variable  $S$  such that  $S = |x|$  ( counts the number of matching pairs in  $x$ ), and have a poisson distribution. The RJCMCMC algorithm generates artificial Markov chains that transit between states of different dimension depending on the temperature (see algorithm 1).

**Algorithm 1:** Initialize  $init\_state = (x_1, ..x_i, ..x_s)$  and  $k = 0$ .  $H_{f'} \subseteq H_f$  and  $H_{h'} \subseteq H_h$  are set of labels such that for  $x = (f, h) \in init\_state$   $f \neq l$  if  $l \in H_{f'}$  or  $h \neq l$  for  $l \in H_{h'}$ . While the temperature  $T$  is above the minimum temperature, do the following,

1.  $k = k + 1$  and generate  $r \sim U[0, 1]$
2. Using  $r$  perform one of the following moves
  - (a) Select uniformly  $x_{pf}$  and  $x_{ph}$  respectively from  $H_{f'}$  and  $H_{h'}$  and form  $x_p = (x_{pf}, x_{ph})$  and set  $new\_state = init\_state \cup \{x_p\}$ .
  - (b) Select uniformly  $x_p$  from  $init\_state$  and set  $new\_state = init\_state \setminus \{x_p\}$ .
  - (c) Uniformly select  $x_p = (x_{pf}, x_{ph})$  from  $init\_state$  and  $x_{pfnew}$  from  $H_{f'}$  set  $new\_state = (init\_state \setminus \{x_p\}) \cup \{(x_{pfnew}, x_{ph})\}$ .
  - (d) Uniformly select  $x_p = (x_{pf}, x_{ph})$  from  $init\_state$  and  $x_{phnew}$  from  $H_{h'}$  set  $new\_state = (init\_state \setminus \{x_p\}) \cup \{(x_{pf}, x_{phnew})\}$ .
3. Let  $x_{new} = new\_state$  and  $x_{init} = init\_state$ , compute ratio probability,  $P$ , 
$$P = \min\{1, \frac{P(|x_{new}|) P(x_{new}|f_{x_{new}})}{P(|x_{init}|) P(x_{init}|f_{x_{init}})}\}$$
4. Accept or reject  $new\_state$  with a probability,  $P$ .
5. Decrease the temperature, and update  $init\_state$ ,  $H_{f'}$  and  $H_{h'}$ .

## 6 Multi-resolution Search

As the number of horizons increases, it takes too long for Algorithm 1 to find the global optimum solution. In some cases, as the algorithm’s parameters are fine tuned to achieve a generalization, considerable numbers of horizons may left unmatched. To solve this problem we use a multi-resolution version of the matching algorithm where strong horizons are matched at coarser level and their matching solution are utilized as priori for matching at finer level weaker horizons.

Approaches to multi scale representation such as wavelet analysis [11] and scale space theory [9] have been introduced. Wavelet analysis are not invariant under thickness thus not suitable for modelling faulted horizons. In scale space, the signal is represented from coarse to fine levels of detail by convolving it with a Gaussian kernel whose standard deviation plays the role of scale. Scale-spaces representation theory does not use any prior knowledge about the signals, whereas here we determine the horizons’ scales in semantic scale, i.e using a-priori knowledge about the horizons’ signals.

The semantic multi-resolution representation is understood in terms of the likelihood of the seismic features observed on the horizons topology. As in section 4.2, horizons seismic features are assumed to form a gaussian random process with strong correlation. At a coarser level, higher likelihoods are considered. At a finer level the threshold for the likelihood goes down and more horizons



topologies are considered (see Fig.5). Mathematically, the horizons scale space,  $\Omega$ , for scale parameter  $t$  is

$$\Omega_{f,\alpha}(\cdot, t) = \{H_{fi} \in H_f | L(V_{fi}) \geq \alpha; V_{fi} = \{T^{-1}(H_{fi})\}\} \quad (11)$$

where  $L(V_{fi})$  is the likelihood function and estimated as in equation 8. After obtaining the hierarchal representation, Algorithm 1 starts from lower resolution level. At the next higher resolution the lower resolution results are utilized to impose constraints  $C_1$  and  $C_2$ , and to initialize latent variables in the expectation maximization algorithm for estimating  $C_3$ .

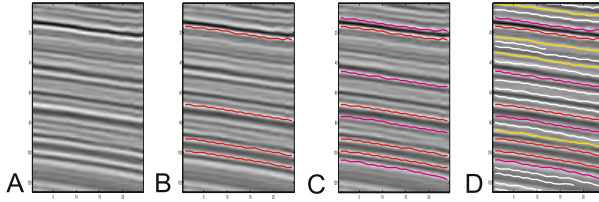


Fig. 5. Horizon segments defined on a seismic slice at different resolution levels

## 7 Experiments and Results

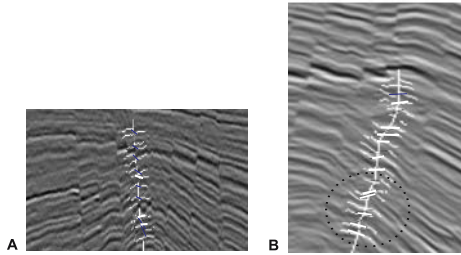
We have conducted experiments to evaluate the matching model developed in the previous sections. The evaluation criteria is the correctness of the matching solution. A correct solution has matched pairs which correspond with manually obtained reference solution, and has no mismatched pairs.

Fault regions are selected from a 3-d seismic volume of shallow-offshore Nigeria. Fault patches were isolated from the seismic data, which consist of large numbers of faults and fault systems. Each isolated fault patch contains a fault surface with seismic sections on the two sides of the surface. The seismic sections were considered to be displaced only under the influence of this single fault surface. The geometries of horizons and faults are assumed to be defined in the seismic section as in Fig.3B. The parameters for algorithm 1 are fine-tuned to the convergence using selected fault reference solution as samples of the a-posteriori distribution. A geometric temperature cooling schedule is utilized. The initial temperature is set to allow jumping from the possible higher energy to the lowest. The final temperature is set close to zero ( $\sim 0.005$ ), and the rate of temperature is 97%. The inner loop was varied from 10 to 100 with number of horizons to match. These optimization parameters are related to the time and fitness of the solution and they are set with some tolerances allowing few horizons unmatched. The method was implemented in MATLAB on Pentium IV pc running windows XP. The simulated annealing process takes from 30 sec to 10 min depending on the number of horizons.

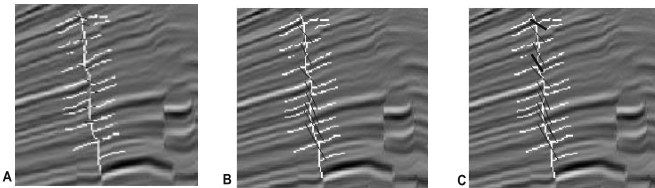
Simulated annealing test runs were executed for 17 fault patches. These fault patches are different from the fault patches utilized for estimating the parameters. Results on 14 fault patches were considered acceptable with less than 20%

possible matching pairs omissions compared to the reference solutions. Matching on three faults was unsuccessful because they contain mismatches. Fig.6 shows for two fault patches unacceptable matching results as there are many mismatches and omissions. The reasons for the failures are poor data quality and nearby faults interactions which distort the fault shape and hence our consistent direction based elliptical fault displacement model.

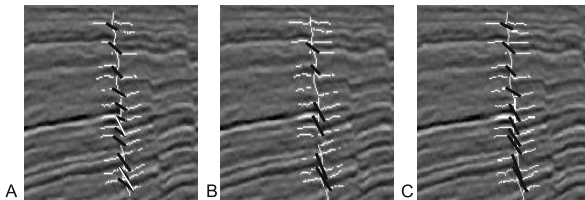
Fig.7 demonstrates the impacts of 3-d information and so its advantage over the 2-d method in [2]. The second method from [1] treats the horizon matching as a non-rigid point-based image registration. An implementation of this method was able to find satisfactory correlations only for 8 fault patches out of the 17 test fault patches. Our new method is more robust and faster because it



**Fig. 6.** Mismatches corrected by white lines are due to poor data quality in (A) and fault interactions in (B)



**Fig. 7.** A. Incorrect matchings due to insufficient information on 2d seismic slices. B. Correct matchings when more slices from the same fault patch are utilized. C. Reference manual matchings (the difference with B) are shown on bold arrow.



**Fig. 8.** A. Wrong matching pairs (corrected by white arrows) are obtained when all horizons are considered for matching search (i.e no multi-scale search). B. Coarse matching; few horizons selected and matched. C. Improved matchings starting from results of B.

matches segments which have more discriminating features than points. The multi-resolution aspect of our matching algorithm explained in section 6 helps to recover every possible matching pair of horizons (see Fig.8) and reduce the simulation time considerably from minutes to seconds.

## 8 Conclusions

Since the ultimate goal is production, good reliable and consistent measures of the interpretation are crucial. We have introduced a stochastic matching model which can facilitate seismic interpretations. The stochastic nature provides quality measures. The automatic matching results compare well with references obtained manually. Additional tests show the inclusions of 3-d spatial continuity and multi-resolution aspects of the dataset increases robustness.

Though the method fails in areas of fault interactions, it helps to bring the the attention of the interpreter and saves the interpreters time as the interpreter will give more attention for such areas than spending time in routine fault regions. Further, the erroneous results could also point to the incorrect fault definition hence pointing to misinterpretations at previous stages. The model can be further extended to incorporate other fault types and well-log observations.

## References

1. F. Admasu and K. Toennies, "Automatic method for correlating horizons across faults in 3d seismic data." *In Pro. the IEEE CVPR*, Washington DC, June, 2004.
2. M. Aurnhammer and K. Toennies, "A genetic algorithm for automated horizon correlation across faults in seismic images." *IEEE Tran. Evolutionary Computation*, vol. 9, No. 2, pp. 201-210, 2005.
3. A. Brown, *Interpretation of Three-Dimensional Seismic Data*, American Association of Petroleum Geologists, 5th edition, December, 1999.
4. G.A. Dorn, "Modern 3D Seismic Interpretation," *The Leading Edge*, Vol. 17, No. 9, pp. 1262-1273, 1998.
5. D. Gibson, M. Spann and J. Turner, "Automatic Fault Detection for 3D Seismic Data," *Proc. 7th Digital Image Computing: Techniques and Applications*, pp. 821-830, Sydney, December, 2003.
6. P. Green, "Reversible jump Markov chain Monte Carlo computation and Bayesian model determination," *Biometrika* 82, vol.57, pp.97-109, 1995.
7. C. Lacoste and X. Descombes and J. Zerubia, "Point Processes for Unsupervised Line Network Extraction in Remote Sensing," *PAMI*, 27(10):pages 1568-1579.
8. D. Stoyan, W.S. Kendall and J. Mecke, *Stochastic Geometry and its applications*, John Wiley & Sons, 1987.
9. T. Lindeberg, *Scale-Space Theory In Computer Vision*, Kluwer Acad. Pub., 1994.
10. J. Walsh and J. Watterson, "Analysis of the relationship between displacements and dimensions of faults," *Journal of Structural Geology*, Vol.10, pp.239-247, 1988.
11. S. Mallat, *A Wavelet Tour of Signal Processing*, Academic Press, 1999.
12. D.S. Lalush and B.M.W. Tsui, "Simulation evaluation of Gibbs prior distributions for use in maximum a posteriori SPECT reconstructions." *IEEE Trans. on Medical Imaging*, Vol. 11, No. 2, pp. 267-275, 1992.

# Robust MEG Source Localization of Event Related Potentials: Identifying Relevant Sources by Non-Gaussianity

Peter Breun<sup>1</sup>, Moritz Grosse-Wentrup<sup>2</sup>, Wolfgang Utschick<sup>1</sup>, and Martin Buss<sup>2</sup>

<sup>1</sup> Institute for Circuit Theory and Signal Processing, Technische Universität München, 80290 München, Germany

[breun@nws.ei.tum.de](mailto:breun@nws.ei.tum.de), [utschick@tum.de](mailto:utschick@tum.de)

<sup>2</sup> Institute of Automatic Control Engineering (LSR), Technische Universität München, 80290 München, Germany

[moritz@tum.de](mailto:moritz@tum.de), [mb@tum.de](mailto:mb@tum.de)

**Abstract.** Independent Component Analysis (ICA) is a frequently used preprocessing step in source localization of MEG and EEG data. By decomposing the measured data into maximally independent components (ICs), estimates of the time course and the topographies of neural sources are obtained. In this paper, we show that when using estimated source topographies for localization, correlations between neural sources introduce an error into the obtained source locations. This error can be avoided by reprojecting ICs onto the observation space, but requires the identification of relevant ICs. For Event Related Potentials (ERPs), we identify relevant ICs by estimating their non-Gaussianity. The efficacy of the approach is tested on auditory evoked potentials (AEPs) recorded by MEG. It is shown that ten trials are sufficient for reconstructing all important characteristics of the AEP, and source localization of the reconstructed ERP yields the same focus of activity as the average of 250 trials.

## 1 Introduction

Event related potentials (ERPs) are magnetic/electric fields of the brain elicited by an event such as presentation of a visual or an auditory stimulus. These fields can be measured outside the skull using magneto- or electroencephalography (MEG/EEG). While all concepts presented in this paper can be applied equally to MEG or EEG, we restrict our discussion to MEG for the sake of simplicity. In the study of ERPs, the process of source localization is concerned with determining the position of neural generators causing the ERP, which allows conclusions about brain areas involved in processing a stimulus. A wide range of methods has been developed for MEG source localization, ranging from simple dipole to distributed source models (see [1] for a review). Common to all methods is their susceptibility to noise, requiring a high signal-to-noise ratio (SNR) of the ERP.

ERPs, however, are cloaked by ongoing background MEG activity usually several times the magnitude of the signal of interest. To extract ERPs from the magnetic background activity numerous trials are recorded, in which the stimulus

is presented repeatedly to the subject. Under the assumption that only the ERP component of the magnetic field is invariant for every stimulus presentation, the average time course of all trials results in an unbiased estimate of the original ERP. The computation of this so called *grand average* ERP usually requires several hundred trials if a positive SNR is desired. Since this amount of data is not always available, source localization algorithms may perform poorly, rendering these approaches impractical for a large group of experimental setups.

For these reasons, the development of methods for source localization that are insensitive to MEG background activity is an active area of research. In this paper, we focus on using Independent Component Analysis (ICA) for this purpose. ICA is a special case of Blind Source Separation (BSS), that decomposes the measured data into maximally independent components (ICs) [2]. In the context of source localization, ICA is used as a preprocessing step to obtain estimates of the topographies of neural sources. The position of a source inside the brain can then be estimated from its topography without interference from other neural sources [3,4]. This approach introduces an error into the localization process if the neural sources are not fully statistically independent. As a remedy, we show that reprojecting of the relevant ICs onto the observation space, and subsequent source localization of the reprojected data, removes this error. This approach requires identification of the ICs contributing to the ERP. This can be done by estimating the non-Gaussianity of each IC.

The efficacy of our approach is tested on auditory evoked potentials (AEPs) recorded by MEG. Ten trials are randomly chosen from a total of 250 trials. Using ICA and reprojecting the most non-Gaussian IC onto the observation space is shown to result in a SNR of 5.52 dB in comparison to the grand average of 250 trials. Reconstructing the current density with a distributed source model results in identical maxima of current strength for the ERP reconstructed from ten trials and the grand average ERP of 250 trials.

The rest of this paper is organized as follows. We first introduce the source model, followed by a review of source localization using distributed source models and ICA. We will then show why correlations between neural sources introduce an error into the source localization procedure when using source topographies estimated by ICA. This motivates the reprojecting of relevant ICs onto the observation space, which renders the localization procedure more robust to source correlations. The relevant ICs contributing to the ERP are then identified by estimating their non-Gaussianity. In the results section, we apply the proposed procedure to AEPs recorded by MEG, and conclude with a brief discussion.

## 2 Methods

### 2.1 Source Model

The determination of the current density from MEG measurements is an ill-posed problem which has no unique solution. A first step to obtain a solution to this inverse problem is to constrain the possible current sources to be dipoles, since the synchronized pre-synaptic potentials of neurons in a cortical column,

that give rise to the MEG, can be approximated by a single current dipole [1]. We restrict ourselves to a spherical head model which allows for a simple determination of the magnetic field  $\mathbf{y}_k$  at the sensor positions generated by the  $k$ -th dipole with known location and orientation (forward problem) [5]:

$$\mathbf{y}_k = \mathbf{A}_k \mathbf{s}_k \in \mathbb{R}^M, \quad (1)$$

with the leadfield matrix  $\mathbf{A}_k \in \mathbb{R}^{M \times 3}$  of dipole  $k$  where  $M$  is the number of available MEG sensors and  $\mathbf{s}_k = [s_{k,x}, s_{k,y}, s_{k,z}]^T$  contains the dipole moment in  $x$ ,  $y$  and  $z$  direction.<sup>1</sup> The magnetic field generated by  $N$  dipoles is the superposition of  $\mathbf{y}_k, k = 1, \dots, N$ . Thus, the model to be used in the sequel is

$$\mathbf{y} = \mathbf{A} \mathbf{s}, \quad (2)$$

with  $\mathbf{A} = [\mathbf{A}_1, \dots, \mathbf{A}_N] \in \mathbb{R}^{M \times 3N}$  and  $\mathbf{s} = [\mathbf{s}_1^T, \dots, \mathbf{s}_N^T]^T \in \mathbb{R}^{3N}$ . The measurement noise is neglected.

Note that the moment vector  $\mathbf{s}_k \in \mathbb{R}^3$  can be written as  $\frac{\mathbf{s}_k}{\|\mathbf{s}_k\|_2} \|\mathbf{s}_k\|_2 = \mathbf{p}_k m_k$ , with the unit norm moment orientation vector  $\mathbf{p}_k \in \mathbb{R}^3$  and the scalar dipole moment  $m_k$ . If the orientation vectors are known or can be estimated simultaneously with the matrix  $\mathbf{A}$ , (2) becomes

$$\mathbf{y} = \mathbf{G} \mathbf{m}, \quad (3)$$

with  $\mathbf{m} = [m_1, \dots, m_N]^T$  and  $\mathbf{g}_k = \mathbf{A}_k \mathbf{p}_k$ , where  $\mathbf{g}_k$  is the  $k$ -th column of  $\mathbf{G} \in \mathbb{R}^{M \times N}$ .

## 2.2 Source Localization

Because of the complex mathematical structure of spatially unconstrained dipole fitting [1], we estimate a discrete approximation of the current density using a large number of current dipoles, placed on a regular grid. It remains to determine the contribution of each of these fixed dipoles to the measurement, thereby reducing the localization to a linear inverse problem (“distributed linear” or “imaging” methods [1]). Because there are typically much more dipoles than sensors, this approach leads to an underdetermined system of linear equations. In order to find a unique solution, it has to be regularized, usually by incorporating assumptions about the spatial properties of the solution, e. g., the norm [6], smoothness [7] or sparseness [8]. These algorithms often use one measurement in time, but there are also attempts to include the temporal evolution (e. g., [9]).

In this paper, we use a distributed approach to determine strongly localized solutions to the inverse problem.<sup>2</sup> Thus, we impose the constraint for the solution to be sparse. This is achieved by the following optimization [10]:

$$\min_{\mathbf{s}} \|\mathbf{A} \mathbf{s} - \mathbf{y}\|_2 + \lambda \|\mathbf{s}\|_1. \quad (4)$$

<sup>1</sup> Note that for MEG the matrix  $\mathbf{A}_k$  has rank 2, since radial dipoles do not contribute to the measurements [5]. Therefore, the dimension of the model could be reduced.

<sup>2</sup> This spatial assumption might not be valid for all neural sources (e.g. cognitive processes) but applies to the AEPs investigated in the results section.

The leadfield matrix  $\mathbf{A}$  is known here because it is fully determined by the fixed positions of the dipoles on the grid. The first term of (4) is used to find a solution which gives a good approximation of the measured data. The second term penalizes the  $\ell_1$  norm of the solution vector which is known to produce sparse solutions [10]. The regularization parameter  $\lambda$  trades the data approximation with the degree of sparsity. Note that this optimization would lead to a solution that is not only sparse in the overall dipole moment, but also in its  $x$ ,  $y$  and  $z$  components which has no physiological justification. Thus, a modified penalty on the norm (see, e. g., [9]) of the solution vector is introduced:<sup>3</sup>

$$\|\mathbf{s}\|_{2,1} \triangleq \sum_{n=1}^N \|\mathbf{s}_n\|_2 = \sum_{n=1}^N \left| \left( \sum_{k \in \{x,y,z\}} s_{n,k}^2 \right)^{\frac{1}{2}} \right|. \quad (5)$$

This produces a sparse solution in the overall dipole moments but not in the respective moment components since the  $\ell_2$  norm is not sparsity enforcing. The resulting optimization problem

$$\min_{\mathbf{s}} \|\mathbf{A}\mathbf{s} - \mathbf{y}\|_2 + \lambda \|\mathbf{s}\|_{2,1} \quad (6)$$

is a second order cone program and can be solved efficiently using standard numerical optimization tools.

### 2.3 Independent Component Analysis

More recently, it has been proposed to apply ICA [2] to perform a BSS of the dipole sources using the measurements of their superimposed activity [11]. This approach explicitly uses the fact that typically not only one MEG measurement is available, but a whole (sampled) time course, and decomposes the measured signal into statistically maximally independent components.

In order to get interpretable results from the ICA, we make the following *assumptions*: The time courses of the dipole moments of ERPs are non-Gaussian distributed and statistically independent to the non-event related background brain activity. This enables us to identify and separate sources as ICs contributing to the ERPs. Furthermore, we assume that there is only a small number  $L < N$  of sources with non-Gaussian dipole moment. This assumption is based on the observation that only a few ICs can be consistently reconstructed, which implies that the other sources have a Gaussian distribution [12]. Additionally—for ICA, not for the subsequent source localization—we restrict ourselves to the case of  $N \leq M$ , i. e., at least as many sensors as sources. If less sources than sensors are present, a Principal Component Analysis (PCA) of the data with a projection on the signal subspace can be performed [13]. Referring to the model introduced in (3), we have  $T$  measurements

$$\mathbf{y}[t] = \mathbf{G}\mathbf{m}[t], \quad t \in \{1, 2, \dots, T\}, \quad (7)$$

<sup>3</sup> Note that the absolute value in (5) is not necessary. It is included to emphasize the application of the  $\ell_1$  norm to the Euclidean ( $\ell_2$ ) norms of  $\mathbf{s}_n$ ,  $n \in \{1, 2, \dots, N\}$ .

where it is assumed that the matrix  $\mathbf{G}$  and thus the location and orientation of the source dipoles do not change over time.

In order to perform a BSS, we determine a so called unmixing matrix  $\mathbf{W}$  such that, applied to the measurements  $\mathbf{y}[t]$ , the components of the resulting vector are statistically independent. For the quadratic case ( $M = N$ ),  $\mathbf{W}$  is an estimate of the inverse of  $\mathbf{G}$ :<sup>4</sup>

$$\hat{\mathbf{m}}[t] = \mathbf{W}\mathbf{y}[t] = \mathbf{W}\mathbf{G}\mathbf{m}[t] = \mathbf{P}\mathbf{m}[t]. \tag{8}$$

This matrix can be found by minimizing the mutual information between the components of the vector  $\hat{\mathbf{m}}[t]$ , the ICs:

$$\min_{\mathbf{W}} I(\hat{m}_1, \dots, \hat{m}_N) = \sum_{n=1}^N H(\hat{m}_n) - H(\hat{\mathbf{m}}) \text{ s. t. } \|\mathbf{w}_n\|_2 = 1, n \in \{1, \dots, N\}, \tag{9}$$

with the differential entropy  $H(z) = -\int_{-\infty}^{\infty} p_z(u) \log(p_z(u)) du$  [14,15]. Note that only the  $L$  non-Gaussian sources can consistently be found by this method, the remaining  $N - L$  Gaussian sources are arbitrarily mixed together [12]. An estimate of the leadfield matrix  $\mathbf{G}$  is obtained by inverting  $\mathbf{W}$ , i. e.,  $\hat{\mathbf{G}} = \mathbf{W}^{-1}$ .

### 2.4 ICA for Source Localization

In the previous section, a method has been described to obtain an estimate  $\hat{\mathbf{G}}$  of the leadfield matrix and the ICs. In [4], the assumption that each IC corresponds to a single dipole and all sources have mutually independent time courses allowed for a decoupled dipole fit since the  $k$ -th column  $\hat{\mathbf{g}}_k$  of  $\hat{\mathbf{G}}$ , also called the topography of the  $k$ -th IC, only depends on the parameters of dipole  $k$ . If one assumes that  $\hat{\mathbf{g}}_k$  corresponds to a distributed source rather than a single dipole, a decoupled localization based on (6) is possible:<sup>5</sup>

$$\min_s \|\mathbf{A}\mathbf{s} - \hat{\mathbf{g}}_k\|_2 + \lambda \|\mathbf{s}\|_{2,1}, \quad k \in \{1, \dots, N\}. \tag{10}$$

A problem arises if the assumption of mutually independent sources contributing to the ERP is violated. Suppose a model with two sources which are obtained by a linear transformation from two statistically independent and non-Gaussian sources  $\mathbf{m}' \in \mathbb{R}^2$ :

$$\mathbf{m} = \mathbf{T}\mathbf{m}' \in \mathbb{R}^2. \tag{11}$$

This gives the observation

$$\mathbf{y} = \mathbf{G}\mathbf{m} = \mathbf{G}\mathbf{T}\mathbf{m}' = \mathbf{G}'\mathbf{m}', \tag{12}$$

---

<sup>4</sup> Note that  $\mathbf{W}\mathbf{G}$  is not the identity but the product  $\mathbf{P}$  of a permutation and diagonal matrix due to the insensivity of the cost function (9) w. r. t. this operation.

<sup>5</sup> The column  $\hat{\mathbf{g}}_k$  is not multiplied with the corresponding IC  $\hat{m}_k$  since this changes the minimizer of (10) only by a scalar factor.



with  $\mathbf{G}' = \mathbf{G}\mathbf{T}$ . The effective mixing matrix  $\mathbf{G}'$  can now be identified by ICA.<sup>6</sup> Assuming the transformation

$$\mathbf{T} = \begin{bmatrix} 1 & c \\ c & 1 \end{bmatrix}, \quad c \in \mathbb{R}, \quad (13)$$

the correlation of the components of  $\mathbf{m}$  is determined by the scalar  $c$  and the two columns of the matrix  $\mathbf{G}'$ , read as  $\mathbf{g}'_1 = \mathbf{g}_1 + c\mathbf{g}_2$  and  $\mathbf{g}'_2 = c\mathbf{g}_1 + \mathbf{g}_2$ . As long as  $|c| \neq 1$ , these two columns are linearly independent and can be identified by ICA. However, the columns  $\mathbf{g}'_1$  and  $\mathbf{g}'_2$  are not identical to  $\mathbf{g}_1$  and  $\mathbf{g}_2$ , and thus an error is introduced if they are used for a decoupled source localization (see 10). The effect on source localization demonstrated by this simple example might not be a problem if (10) is used, since only linear correlations are present. However, it is emphasized that care should be taken if the independence assumption does not hold in general. This phenomenon can occur, e. g., for the case of AEPs [16]. In the next subsection, we propose a method to deal with this problem.

## 2.5 ICA for Preprocessing of MEG Data

While the results of ICA may not be directly suitable for a source localization, they can still be used to extract relevant activity from the measured data. This “denoised” signal can then be the basis for a more accurate localization.

A major assumption of the presented ICA approach is that only neural activity of interest results in a non-Gaussian signal, while the Gaussian background activity results in Gaussian ICs. This gives a criterion to decide which components contribute to the signal of interest. Assuming that there are  $L$  non-Gaussian ICs, we get a “denoised” signal by computing

$$\hat{\mathbf{y}}[t] = \hat{\mathbf{G}}^{(L)} \hat{\mathbf{m}}^{(L)}[t], \quad (14)$$

where  $\hat{\mathbf{m}}^{(L)}[t] \in \mathbb{R}^L$  contains only the non-Gaussian ICs and  $\hat{\mathbf{G}}^{(L)} \in \mathbb{R}^{M \times L}$  the corresponding columns of  $\mathbf{W}^{-1}$ .<sup>7</sup> For the estimation of Gaussianity, we employ the Anderson-Darling test (see, e. g., [17]) which is based on a distance measure between the empirical distribution function of the available data and the cumulative distribution function to be tested for. This is done separately for each IC, since they are assumed to be statistically independent.

The advantage of this approach lies in the fact that the columns of  $\hat{\mathbf{G}}$ , which may be affected by correlations of the signal components, are not used for a decoupled source localization. Instead, we remove the signal portion assumed to be background activity that is statistically independent to the signal of interest and proceed with a simultaneous localization of the remaining sources. Assuming that the non-Gaussian ICs have been correctly identified, the signal  $\hat{\mathbf{y}}[t]$  is used as data for the source localization algorithm as described above.

<sup>6</sup> Up to permutation and scaling which is not considered here for simplicity.

<sup>7</sup> Note  $\mathbf{W}^{-1}$  is actually an estimate of  $\mathbf{G}'$  (with the problems mentioned in Section 2.4). Only for statistically independent sources, this is also an estimate of  $\mathbf{G}$ .

### 3 Results

To test the efficacy of the proposed approach, we used MEG data recorded during an auditory oddball task at the Biomagnetic Imaging Laboratory at the University of California, San Francisco. The data was collected with  $M = 132$  channels covering the right hemisphere, and was sampled at 4 kHz. Each trial started with presentation of an auditory stimulus to the left ear, and lasted 275 ms (see [18] for a description). The grand average ERP  $\mathbf{y}^*$  was calculated as the average of 250 trials and low-pass filtered with 16 Hz cut-off frequency [15]. The resulting time course at all channels is shown in Fig. 1 (left panel).

Ten trials were chosen randomly as test data. After low-pass filtering with 50 Hz cut-off frequency and subtracting the mean of each channel, a PCA was performed, only retaining the first 50 principal components [13].<sup>8</sup> The extended Infomax-ICA algorithm [14] was applied to the concatenated trials, resulting in 50 ICs and associated topographies  $\{\hat{m}_k[t], \hat{\mathbf{g}}_k \in \mathbb{R}^M\}$ ,  $k = 1, \dots, 50$ . After computing the mean time course across the ten trials for each IC, the non-Gaussianity of each averaged IC was estimated using the Anderson-Darling test mentioned in section 2.5.

A remarkable result is that only one IC showed a high degree of non-Gaussianity, which was 5.9 times the standard deviation (std) apart from the mean non-Gaussianity of all ICs. This IC was only ranked eighth in terms of explained variance of the original measurements as expected from the low SNR of the data set. Since the IC with the second highest non-Gaussianity was only 2.4 stds apart from the mean non-Gaussianity, only the most non-Gaussian IC was assumed to contribute to the ERP. Hence, we conclude that  $L = 1$ .

To reconstruct the ERP, the most non-Gaussian IC was reprojected onto the observation space (cf. 14), and the reconstructed ERP  $\hat{\mathbf{y}}$  was compared with the grand average ERP  $\mathbf{y}^*$  by computing the SNR, defined as [18]

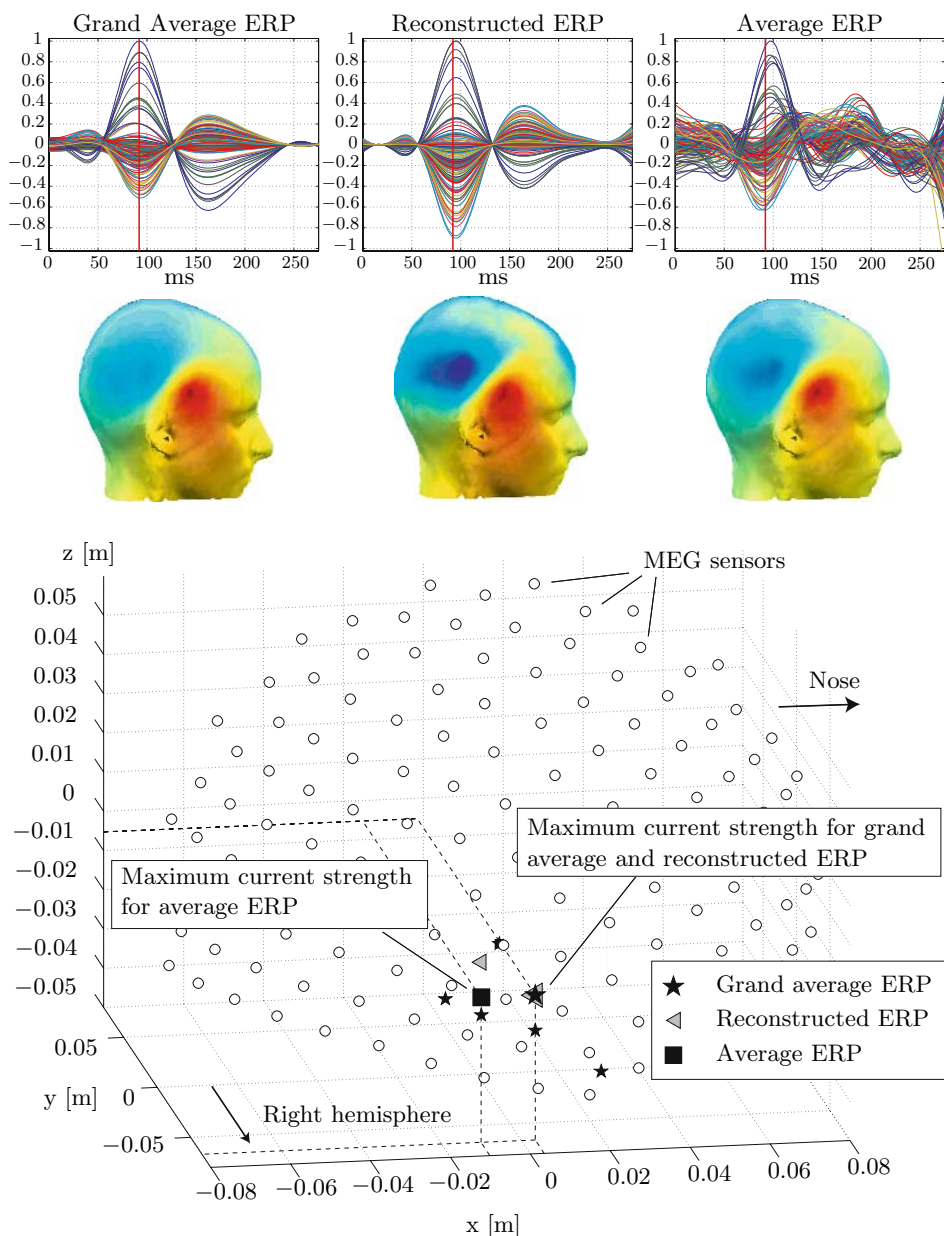
$$\text{SNR} = 10 \log_{10} \left[ \frac{1}{M} \sum_{i=1}^M \left( \sum_{t=1}^N y_i^*[t]^2 \right) / \left( \sum_{t=1}^N (y_i^*[t] - \hat{y}_i[t])^2 \right) \right] \text{ (dB)}. \quad (15)$$

This resulted in a SNR of 5.52 dB and the time course shown in the middle panel of Fig. 1. In comparison with only averaging the ten trials and low-pass filtering the resulting raw average with 16 Hz cut-off frequency (Fig. 1, right panel), an increase in SNR of 5.79 dB was achieved.<sup>9</sup>

The effect of the denoising for source localization was assessed by estimating the current distribution for all three data sets (i. e., grand average, reconstructed and average, see Fig. 1), using the distributed source localization procedure described in the methods section. A spherical head model was assumed, with the MEG sensors located at a radius of 8.5 cm. A regular grid with an inner radius

<sup>8</sup> Here, we have to set  $M = 50$  for ICA and can therefore reconstruct up to  $N = 50$  ICs. Note that  $M$  and  $N$  have different values for the subsequent source localization.

<sup>9</sup> Note that for comparison the grand average, reconstructed and average only, ERPs were all normalized to their first major peak occurring around 95 ms.



**Fig. 1.** Time course, topography at time point of grand average peak amplitude and current distribution for grand average, denoised and raw average ERP datasets. The vertical line in the first row shows the time index of the topographies and source localization results in the two lower panels. The topographies were plotted using EEGLab [15].

of 5.5 cm and outer radius of 7.1 cm was placed inside the spherical head model, with a distance of 4.5 mm between each grid point. This resulted in  $N = 7437$  dipoles. The time instant of maximum amplitude of the grand average ERP (92 ms) was chosen for localization for all three data sets. The same regularization parameter was used for all data sets, with  $\lambda$  chosen to achieve a good trade-off between sparsity and approximation of the measurements (resulting in a residual variance of 0.73% for the grand average, 7.6% for the reconstructed and 1.9% for the average only ERP). The results of the localization are shown in the bottom row of Fig. 1. The location of maximum current strength for the grand average, the reconstructed and the average ERP are indicated by the major star, triangle and square, respectively. Smaller indicators represent grid points with a current strength of at least 50% of the maximum current strength.

The grid point of maximum current strength of the reconstructed ERP coincided with the maximum grid point of the grand average ERP. The maximum current strength for the average ERP on the other hand was located three grid points (1.35 cm) apart from the focus of activity of the grand average ERP. Five points of the grand average ERP showed a current strength of at least 50% of the maximum current strength, while this was the case for only one more point for the reconstructed ERP.

## 4 Discussion

To summarize the results, the proposed approach proved capable of achieving a significant increase of SNR in comparison to only averaging and low-pass filtering the data (+5.79 dB). While the peak latencies and amplitudes of the AEP were difficult to identify in the averaged data set, the reconstructed ERP showed a similar time course as the grand average ERP, with all peaks clearly identifiable. We conclude, that with the proposed approach a small number of trials is sufficient to reconstruct the most important characteristics of ERPs. Furthermore, the results support our assumption that non-Gaussianity of ICs is a good criterion for identifying ICs contributing to an ERP.

In terms of source localization, the focus of activity of the reconstructed ERP coincided with the grand average ERP, while the focus of activity of the average ERP was shifted by 1.35 cm. It should be noted that AEPs have a relatively high SNR compared to other ERPs, resulting in acceptable results by averaging only. Further studies have to show, how our approach performs in comparison to only averaging if more complex experimental setups are being investigated.

Finally, it should be noted that the data set used in this study was not well suited for investigating the issue of correlations between neural sources discussed in section 2.4. Since one IC was sufficient to reconstruct the grand average ERP, it can be concluded that no correlations between neural sources contributing to the ERP existed.<sup>10</sup> This issue will also be investigated in further simulation studies and studies with more complex experimental setups.

<sup>10</sup> Note that this does not exclude the possibility of the most non-Gaussian IC representing a distributed source.

## References

1. Baillet, S., Mosher, J.C., Leahy, R.M.: Electromagnetic brain mapping. *IEEE Signal Processing Magazine* **18**(6) (2001) 14–30
2. Comon, P.: Independent component analysis, a new concept? *Signal Processing* **36**(3) (1994) 287–314
3. Cao, J., Murata, N., Amari, S., Cichocki, A., Takeda, T.: Independent component analysis for unaveraged single-trial MEG data decomposition and single-dipole source localization. *Neurocomputing* **49** (2002) 255–277
4. Zhukov, L., Weinstein, D., Johnson, C.: Independent component analysis for EEG source localization - an algorithm that reduces the complexity of localizing multiple neural sources. *IEEE Engineering in Medicine and Biology Magazine* **19**(3) (2000) 87–96
5. Mosher, J.C., Lewis, P.S., Leahy, R.M.: Multiple Dipole Modeling and Localization from Spatio-Temporal MEG Data. *IEEE Transactions on Biomedical Engineering* **39**(6) (1992) 541–557
6. Hämmäläinen, M.S., Ilmoniemi, R.J.: Interpreting magnetic fields of the brain: minimum norm estimates. *Medical & Biological Engineering & Computing* **32** (1994) 35–42
7. Pascual-Marqui, R.D., Michel, C.M., Lehmann, D.: Low resolution electromagnetic tomography: a new method for localizing electrical activity in the brain. *International Journal of Psychophysiology* **18** (1994) 49–65
8. Gorodnitsky, I.F., George, J.S., Rao, B.D.: Neuromagnetic source imaging with FOCUSS: a recursive weighted minimum norm algorithm. *Journal on Electroencephalography and clinical Neurophysiology* **95**(4) (1995) 231–251
9. Cotter, S.F., Rao, B.D., Egan, K., Kreutz-Delgado, K.: Sparse Solutions to Linear Inverse Problems With Multiple Measurement Vectors. *IEEE Transactions on Signal Processing* **53**(7) (2005) 2477–2488
10. Chen, S.S., Donoho, D.L., Saunders, M.A.: Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing* **20**(1) (1998) 33–61
11. Makeig, S., Bell, A.J., Jung, T.P., Sejnowski, T.J.: Independent Component Analysis of Electroencephalographic Data. In Touretzky, D.S., Mozer, M.C., Hasselmo, M.E., eds.: *Advances in Neural Information Processing Systems 8*, MIT Press (1996) 145–151
12. Grosse-Wentrup, M., Buss, M.: Subspace Identification Through Blind Source Separation. *IEEE Signal Processing Letters* **13**(2) (2006) 100–103
13. Joho, M., Mathis, H., Lambert, R.H.: Overdetermined blind source separation: Using more sensors than source signals in a noisy mixture. In: *Proc. Independent Component Analysis and Blind Signal Separation ICA*. (2000) 81–86
14. Bell, A.J., Sejnowski, T.J.: An information maximization approach to blind separation and blind deconvolution. *Neural Computation* **7**(6) (1995) 1129–1159
15. Delorme, A., Makeig, S.: EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics. *Journal of Neuroscience Methods* **134** (2004) 9–21
16. Dalal, S.S., Sekihara, K., Nagarajan, S.S.: Modified Beamformers for Coherent Source Region Suppression. *IEEE Transactions on Biomedical Engineering* (2006) *Accepted for future publication*.
17. Stephens, M.A.: EDF Statistics for Goodness of Fit and Some Comparisons. *Journal of the American Statistical Association* **69**(347) (1974) 730–737
18. Nagarajan, S.S., Attias, H.T., Hild II, K.E., Sekihara, K.: A graphical model for estimating stimulus-evoked brain responses from magnetoencephalography data with large background brain activity. *Neuroimage* **30**(2) (2006) 400–416

# Classifying Event-Related Desynchronization in EEG, ECoG and MEG Signals

N. Jeremy Hill<sup>1</sup>, Thomas Navin Lal<sup>1</sup>, Michael Schröder<sup>2</sup>,  
Thilo Hinterberger<sup>3</sup>, Guido Widman<sup>4</sup>, Christian E. Elger<sup>4</sup>,  
Bernhard Schölkopf<sup>1</sup>, and Niels Birbaumer<sup>3,5</sup>

<sup>1</sup> MPI for Biological Cybernetics, Spemannstr. 38, 72076 Tübingen  
{jeremy.hill, navin.lal, bernhard.schoelkopf}@tuebingen.mpg.de

<sup>2</sup> Fraunhofer FIRS IDA group, Kekuléstr. 7, 12489 Berlin  
michael.schroeder@first.fraunhofer.de

<sup>3</sup> Inst. of Medical Psychology and Behavioral Neurobiology, University of Tübingen  
{thilo.hinterberger, niels.birbaumer}@uni-tuebingen.de

<sup>4</sup> Department of Epileptology, University of Bonn  
{guido.widman, christian.elger}@ukb.uni-bonn.de

<sup>5</sup> NIH Human Cortical Physiology Unit, Bethesda, USA

**Abstract.** We employed three different brain signal recording methods to perform Brain-Computer Interface studies on untrained subjects. In all cases, we aim to develop a system that could be used for fast, reliable preliminary screening in clinical BCI application, and we are interested in knowing how long screening sessions need to be. Good performance could be achieved, on average, after the first 200 trials in EEG, 75–100 trials in MEG, or 25–50 trials in ECoG. We compare the performance of Independent Component Analysis and the Common Spatial Pattern algorithm in each of the three sensor types, finding that spatial filtering does not help in MEG, helps a little in ECoG, and improves performance a great deal in EEG. In all cases the unsupervised ICA algorithm performed at least as well as the supervised CSP algorithm, which can suffer from poor generalization performance due to overfitting, particularly in ECoG and MEG.

## 1 Introduction

Several different technologies exist for measuring brain activity, any of which might be potentially useful in the design and implementation of Brain-Computer Interface (BCI) systems. Each system has its own particular set of advantages and limitations as regards spatial and temporal resolution, as well as cost, portability and risk to the user. Comparative studies are required in order to guide development, to explore the trade-offs between these factors.

Here we present a comparative study of motor-imagery BCI experiments based on electroencephalography (EEG), electrocorticography (ECoG) and magnetoencephalography (MEG). In all three, our goal is to develop techniques of analysis that could be used for efficient exploratory screening of potential users,

using a simple binary synchronous (trial-based) paradigm, to determine whether subsequent lengthy training in motor imagery might be worthwhile. This setting requires that we obtain good classification performance as quickly as possible, ideally within the duration of a single recording session, and in particular we will be interested to find out how long such a session needs to be. Since the user will have little time to adapt in one session, performance must be maximized by careful choice and optimization of pattern recognition algorithms.

## 2 Data Collection

Three experiments form the basis for this report:

- **EEG:** 8 healthy subjects each performed 400 trials seated in an armchair in front of a computer monitor. Signals from 39 silver/silver chloride electrodes were digitized at 256 Hz.
- **ECoG:** 4 patients with epilepsy (implanted short-term with ECoG electrode arrays in order to locate their epileptic foci prior to surgery) each performed 100–200 trials seated in their hospital bed facing a monitor. Signals from 64–84 subdurally implanted platinum electrodes were digitized at 1000 Hz.
- **MEG:** 10 healthy subjects each performed 200 trials, seated in the MEG scanner in front of a projector screen. Signals from 150 superconducting magnetometers were digitized at 625 Hz.

Each trial began with a small fixation cross displayed at the centre of the screen, indicating that the subject should not move, and blink as little as possible. One second later the randomly chosen task cue was displayed for 500 msec, instructing the subject to imagine performing one of two different movements: these were left hand and right hand movement for the EEG study, and movement of either the left little finger or the tongue for the MEG and the ECoG studies (ECoG grids were implanted on the right cerebral hemisphere). The imagined movement phase lasted at least 3 seconds, then the fixation point was extinguished, marking the end of the trial. Between trials was a short relaxation phase of randomized length between 2 and 4 seconds. For further methodological details of the three experiments, see Lal et al (2004, 2005b,a).

## 3 Preprocessing and Classification

The problem setting is the familiar one of binary classification: each trial is a data point, and its label tells us which of two imagined movements the subject was attempting to perform.

For each number of trials  $n$  from 25, in steps of 25, up to the maximum available, we attempt to classify the first  $n$  trials performed by the subject. Classification performance is assessed using 10-fold cross-validation, conducted twice with different random seeds. On each of these 20 folds, the test fold was excluded from training, feature and model selection: where necessary, model

and feature selection was performed by a second level of 10-fold cross-validation, *within* the training fold of the outer cross-validation.

We expect label-relevant changes to manifest themselves in the power spectra of the signals (so-called Event-Related Desynchronization, ERD) as well as in slower drifts of electrical potential (Movement-Related Potentials, MRP). For simplicity, since only a small minority of our subjects demonstrated useable MRPs, we focus our attention on ERD, which is a movement- or imagined-movement-related drop in the sensorimotor rhythms that typically dominate motor and premotor cortex activity in the 8–12 Hz and 18–22 Hz ranges.

Starting 500 msec after offset of the visual task cue, we extract a window of length 2 seconds, which we downsample to 100 Hz<sup>1</sup> and then linearly detrend. A spatial filter is then computed (see below), and applied to both the training and test trials. Then, amplitude spectra are computed by Welch’s short-time Fourier transform (STFT) method, averaging the spectra of 5 50%-overlapping 670-msec windows. This gives us a vector of log amplitudes at 65 different frequencies for each sensor (or for each spatially filtered linear combination of sensors, which we will call a “channel”) on each trial, as inputs to the classifier.

We use a Support Vector Machine (SVM) as the classifier. Kernel methods such as the SVM are particularly well-suited to BCI data: training time depends on number of points rather than number of features, which is expedient because datasets are typically high-dimensional but small (the number of trials that can be obtained per session is usually limited to a few hundred, and pooling data across sessions usually results in a drop in performance). They also allow the easy modelling of non-linear relationships, although it has generally been observed in BCI classification applications (for example, see Müller et al, 2003) that, given a well-chosen sequence of preprocessing steps (an explicit feature mapping), a further implicit mapping via a non-linear kernel is usually unnecessary: thus it is commonly reported that a linear classifier performs about as well as any non-linear classifier one might attempt. Indeed we have generally found it to be the case in the current application. We therefore use a linear kernel for the current study.

First, the SVM’s regularization parameter is optimized using 10-fold cross validation within the training trial subset. Then we select relevant channels using the technique of recursive channel elimination (RCE) first described by Lal et al (2004). This is a variant of the recursive feature elimination (RFE) method proposed by Guyon et al (2002), with the features being grouped into subsets that corresponding to channels, and one whole subset being eliminated at each step. We perform 10-fold cross-validated RCE within the training subset, testing every trained SVM on the inner test fold in order to obtain an estimate of performance as a function of the number of features. RCE also gives us a rank order of the channels—based on this, we reduce the number of channels, choosing the

---

<sup>1</sup> Based on an examination of the ROC scores of individual frequency features in each subject’s data set, and also of the weight vector of a linear classifier trained on the data, we did not find any indication that information in the power spectrum above 50 Hz helped in separating classes in the current task, so we do not believe any class-relevant information was lost by downsampling.



minimum number for which the estimated error is within 2 standard errors of the minimum error during elimination. This procedure is described in more detail in Lal et al (2005a). Finally, the regularization parameter is re-optimized on the data set after channel rejection and the classifier is ready to be trained on the training subset of the outer fold, in order to make predictions on the test subset.

### 3.1 Spatial Filtering

A spatial filter is a vector of weights specifying a linear combination of sensor outputs. We represent our signals as an  $s$ -by- $t$  matrix  $X$ , consisting of  $s$  time series, each of length  $t$ , recorded from  $s$  different sensors. Spatial filtering amounts to a premultiplication  $X' = WX$ , where  $W$  is an  $r$ -by- $s$  matrix consisting of  $r$  different spatial filters. If an appropriate spatial filter is applied before any non-linear processing occurs (such as the non-linear step of taking the absolute values of a Fourier transform to obtain an amplitude spectrum), then class separability according to the resulting features will often improve. We compare three spatial filtering conditions: no spatial filtering (where  $W$  is effectively the identity matrix, so we operate on the amplitude spectra of the raw sensor outputs), Independent Components Analysis and Common Spatial Pattern filtering.

**Independent Component Analysis (ICA):** Concatenating the  $n$  available trials to form  $s$  long time series, we then compute a (usually square) separating matrix  $W$  that maximizes the independence of the  $r$  outputs. Linear blind source separation is popular in the analysis of EEG signals since cortical activity is measured through several layers of bone and tissue, spatially blurring the signals to yield highly correlated (roughly linear) mixtures of the signals of interest at the sensors.

We use an ICA algorithm that optimizes  $W$  using the Infomax criterion, implemented as part of EEGLAB (see Delorme and Makeig, 2004) which we find to be comparable to most other available first-order ICA algorithms in terms of the improvement in resulting classification performance, while at the same time having the advantage of supplying more consistent spatial filters than many others.

**Common Spatial Pattern (CSP) Analysis:** This technique (due to Koles et al, 1990) and related algorithms (Lemm et al, 2004; Dornhege et al, 2004, 2006) are supervised methods for computing spatial filters whose outputs have maximal between-class differences in variance. The input to the algorithm must therefore be represented in such a way that class-dependent changes in the signal are reflected in a change in signal variance. For ERD, this can be achieved by applying a band-pass filter capturing the part of the spectrum in which sensorimotor rhythms are expressed: the variance of the filtered signal, which has zero mean, is a measure of amplitude in the chosen band. Here we use a bandpass filter between 7 and 30 Hz (we generally found that this broad band performed approximately as well as any specifically chosen narrow band).

Like ICA, CSP can be seen as a whitening followed by a rotation of time-samples represented in the  $s$ -dimensional space of sensors. The whitening step is cheap in both, requiring only a matrix  $P$  such that  $P\Sigma P^T = I$ , where  $\Sigma$  is the  $s$ -by- $s$  covariance matrix of the filtered, concatenated training trials. The rotation

step in CSP is also cheap: we diagonalize the covariance matrix of only one class (for example,  $X_+$ , the concatenated training trials from only the positive class) in the whitened space:  $P\Sigma_+P^\top = RDR^\top$ , where  $RR^\top = I$  and  $D$  is diagonal. The eigenvalues  $\text{diag}(D)$  are in the range  $[0, 1]$  and, since  $P\Sigma_-P^\top = R(I - D)R^\top$ , eigenvalues close to 0 or 1 indicate directions that maximize variance in one class while simultaneously minimizing it in the other. See for example Lemm et al (2004) for a more in-depth account and an extension of the algorithm.

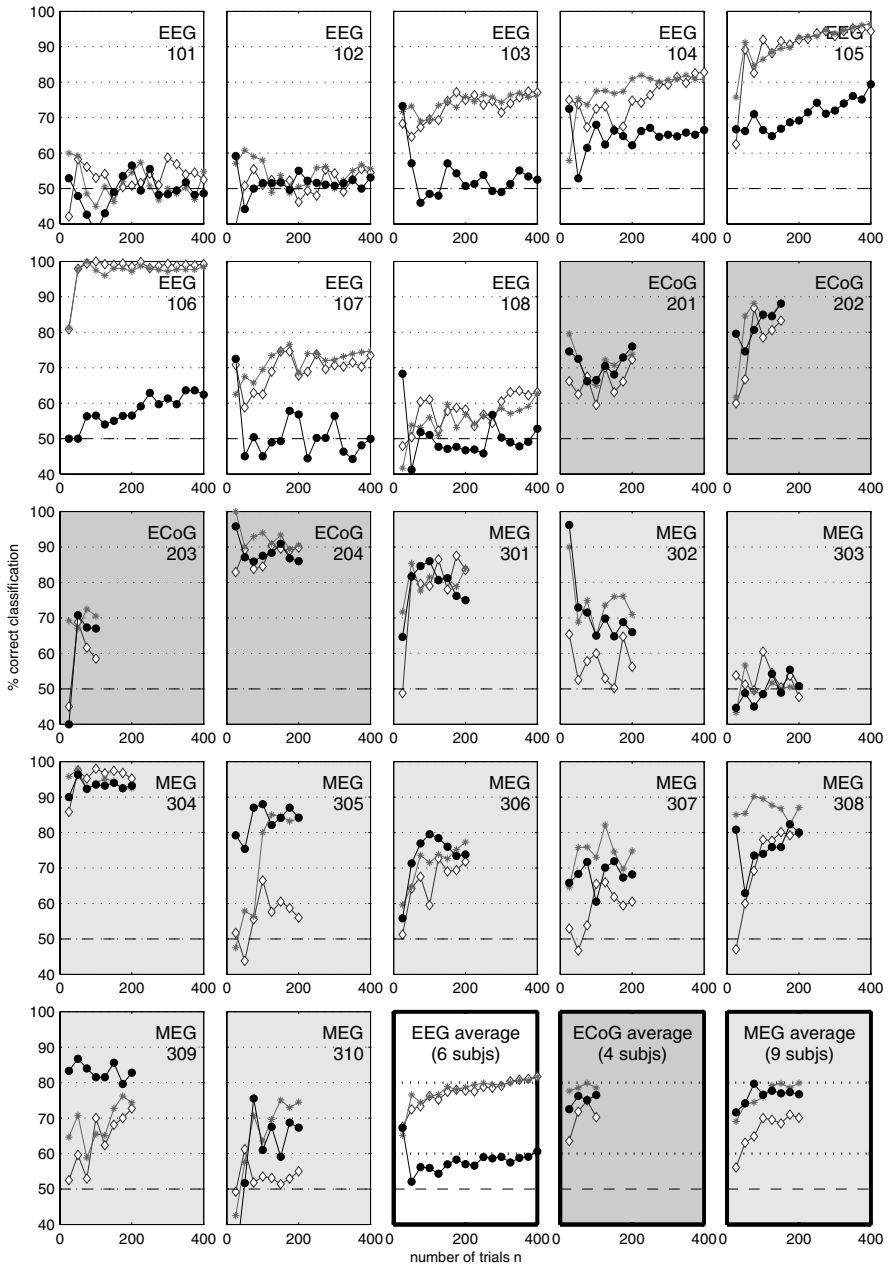
Since CSP uses label information, it must be performed once for each outer training/test fold, using the training subset only. The drawback to CSP is its tendency to overfit. Since this depends on the number of entries in  $R$  that can be optimized relative to the amount of data, the overfitting effect is worse when there are a larger number of channels relative to the number of trials, as is the case in our ECoG and MEG data sets. In practice, the eigenvalues are a fairly good and often-used predictor of the generalization performance of each spatial pattern. One common approach is to take only the first  $k$  patterns, in the order of preference indicated by the eigenvalues, number  $k$  being either fixed, or determined by cross-validation of the CSP algorithm within the training set. We employ this strategy with  $k$  fixed at 5, and have generally found little difference in performance between this and a cross-validation strategy. We also allow the RCE stage to reduce the number of channels further, if indicated by the criterion described above.

## 4 Classification Performance Results

In figure 1, classification accuracy is plotted as a function of  $n$  for each subject, along with average performance in each of the three experiments (EEG, ECoG and MEG). We plot the time-course of the *overall* effectiveness of the experimental setup, subject and classifier taken all together: our curves are obtained by computing performance on the *first* 25 trials performed by the subject, then recomputing based on the first 50 trials, and so on (instead of on a random 25 trials, then a random 50 trials). As a result the observed changes in performance with increasing  $n$  reflect not only the effect of the amount of input on classifier performance, but also changes in the subjects' performance, whether due to practice, fatigue or transient random influences.

Note that, for 2 out of 8 subjects in the EEG condition (subjects 101 and 102), and 1 out of 10 in MEG (subject 303), we were never able to classify at significantly better than chance level.<sup>2</sup> These subjects were omitted from the averaging process: from the point of view of ascertaining the effect of trial number, they would only add noise.

<sup>2</sup> As a screening, our study might suggest that other (non-motor-imagery) BCIs should be tried for these three subjects. This decision would in practice be determined by comparative screening, in which preliminary results from a number of different BCI strategies are compared for a given subject. Even if all approaches fail at the screening stage, however, it may still be worth attempting training, since users training can improve results even for users who start from chance performance level (see for example Kübler et al, 2005).



**Fig. 1.** For each subject, classification accuracy is plotted as a function of the number of trials performed and the spatial filtering method employed: filled circles denote no spatial filtering, asterisks denote ICA, and open diamonds denote CSP. The last three plots show averages, for the EEG, ECoG and MEG experiments respectively, across all subjects for whom classification had been possible at all.

In the EEG experiment, both ICA (grey asterisks) and CSP (open diamonds) allow very large improvements in performance relative to the condition in which no spatial filtering was used (filled circles). This effect is clear in the averaged data as well as in the individual subject plots. In ECoG, the difference between ICA and no spatial filtering is slight, although ICA is at least as good as no spatial filtering for all four subjects; CSP is consistently a little worse than either. In MEG, there is no consistent benefit or disadvantage to ICA over the raw sensor outputs, and again CSP is worse, this time by a larger margin.

The failure of CSP in ECoG and MEG is likely to be related to the overfitting effect mentioned above. This is clearest for subject 310 when 200 trials are used: although spatial filters exist (and have been found by ICA) which can improve classification performance, CSP fails to find any patterns which help to classify the data, because useless (overfitted) spatial patterns dominate the decomposition of the class covariance matrices.

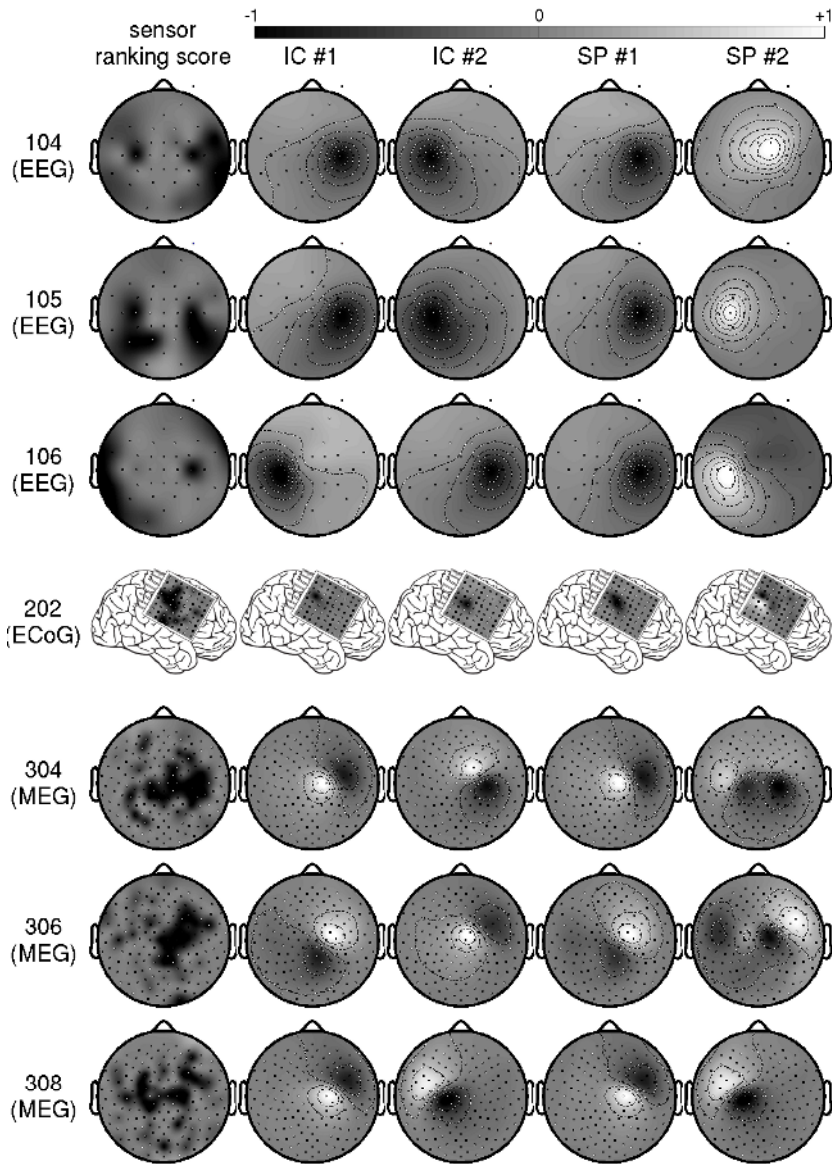
Overall, maximum performance can be achieved using about 200 trials in EEG and 75–100 trials in MEG. For ECoG, though it is harder to draw strong conclusions due to the smaller number of subjects and trials, it generally appears that the curves are even flatter: the best results can already be obtained with only 25–50 trials.

## 5 Topographic Interpretation of Results

Figure 2 shows topographic maps of the features selected by our analysis, for seven of our subjects. Sensor ranking scores were obtained by Recursive Channel Elimination on data that had not been spatially filtered: each of the 20 outer training/test folds of the analysis returned a channel ranking, and these ranks were averaged across folds and then divided by their standard deviation across folds. The result indicates which channels were ranked highly most consistently (darker colours indicating channels ranked as more influential). We also plot spatially interpolated projected amplitudes<sup>3</sup> for the top two independent components (selected by Recursive Channel Elimination in the first outer training/test fold) and the first two spatial patterns (indicated by the best two eigenvalues in the first outer fold).

In general, we see that ICA and CSP recover very similar patterns of activation which are consistent with the modulation of activity in motor and pre-motor cortical areas. In EEG, both algorithms recover patterns centred on C4/CP4 in the right hemisphere (where we would expect modulation associated with imagined left hand movement) and C3/CP3 in the left (imagined right hand movement). In the ECoG, the top two independent components and the top spatial pattern are all highly localized, activation in each case being focused on just three or fewer electrodes located above the motor cortex. In MEG, we see patterns consistent with parietal-central and central-frontal dipoles in the right hemisphere where we would expect to see modulation associated with imagined

<sup>3</sup> Each map is spline-interpolated from a single column of the mixing matrix  $W^{-1}$ , the inverse of the spatial filter matrix. The column corresponding to a given estimated source tells us the measured amplitude of that particular source as a function of sensor location.



**Fig. 2.** Topographic maps showing the ranking or weighting of sensors at different spatial locations, for three EEG subjects, one ECoG subject, and three MEG subjects. Sensor ranking scores (first column) are obtained by Recursive Channel Elimination on the data when no spatial filtering is used. The top two independent components (columns 2–3) are selected by Recursive Channel Elimination after Independent Component Analysis. The top two spatial patterns (columns 4–5) are selected using the eigenvalues returned by the CSP algorithm. Topographic maps are scaled from -1 (black) through 0 (grey) to 1 (white) according to the maximum absolute value in each map.

left hand movement. For subject 308, a left-hemisphere source is also identified, perhaps corresponding to imagined tongue movement.<sup>4</sup>

The ranking scores of the raw sensors, while presenting a somewhat less tidy picture, generally show a similar pattern of sensor importance to that indicated by the ICA and CSP maps (note that the ranking score patterns may reflect information from influential sources beyond just the first two components that we have shown). The sensors ranked most consistently highly are to be found in lateralized central and pre-central regions, bilaterally for the EEG experiment and for subject 308, and with a right-hemisphere bias for the others. For further examination of the performance of Recursive Channel Elimination in the identification of relevant source locations, see Lal et al (2004, 2005b,a).

## 6 Summary

We have compared the classifiability of signals obtained by EEG, ECoG and MEG in a binary, synchronous motor-imagery-based Brain-Computer Interface. We held the time interval, and (after failing to find any information useful for the classification in frequencies above 50 Hz) also the sampling frequency, constant across sensor types, and classified event-related desynchronization effects in the signals' amplitude spectra using regularized Support Vector Machines and automatic feature selection.

We varied the number of trials used, in order to see how quickly we might reach maximum classification performance with our unpractised subjects. Maximum performance, averaged across subjects, was roughly equal across sensor types at around 80%, although subject groups were small and between-subject variation was large, so we attach no particular weight to this observation. Performance levelled off after about 200 trials in EEG, 75–100 trials in MEG, and 25–50 trials in ECoG.

Performance was affected by spatial filtering strategy in a way that depended on the recording hardware. For EEG, spatial filtering is crucial: large gains in classification accuracy were possible using either first-order Independent Component Analysis or the Common Spatial Pattern algorithm, the performance of these two approaches being roughly equal. For ECoG and MEG, as one might expect from systems that experience less cross-talk between channels, spatial filtering was less critical: the MEG signals were the “cleanest” in this regard, in that there was no appreciable difference in performance between classification of the raw sensor outputs and classification of any of the linear combinations of sensors we attempted. First-order spatial filtering appears largely redundant for detecting ERD in these systems.

---

<sup>4</sup> In general, however, as also reported in Lal et al (2005a) we found little evidence of desynchronization, and accordingly few spatial patterns, specifically correlated with the “tongue” class in ECoG and MEG. Classification may be based on “hand vs. nothing” for many of the subjects, leading us to conclude that the finger-vs-tongue strategy may not be the best choice.

Across all three conditions, ICA was the best (or roughly equal-best) spatial filtering strategy. CSP suffered badly from overfitting in the ECoG and MEG conditions, resulting in poor generalization performance. We did not find a convincing advantage, with any of the three sensor types, of supervised optimization of the spatial filters over blind source separation.

## Acknowledgements

We thank Hubert Preissl, Jürgen Mellinger, Martin Bogdan, Wolfgang Rosenstiel, and Jason Weston for helpful discussions. We gratefully acknowledge the financial support of the *Max-Planck-Gesellschaft*, the *Deutsche Forschungsgemeinschaft* (SFB550/B5 and RO1030/12), the European Community IST Programme (IST-2002-506778 under the PASCAL Network of Excellence), and the *Studienstiftung des deutschen Volkes* (grant awarded to T.N.L.).

A more detailed version of this report will appear in G. Dornhege, J. Millán, T. Hinterberger, D. McFarland and K.-R. Müller (Eds), “Towards Brain-Computer Interfacing,” MIT Press, 2006 (in press).

## References

- Delorme A, Makeig S (2004) EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics. *Journal of Neuroscience Methods* 134:9–21
- Dornhege G, Blankertz B, Curio G, Müller KR (2004) Increase information transfer rates in bci by csp extension to multi-class. *NIPS* 16
- Dornhege G, Blankertz B, Krauledat M, Losch F, Curio G, Müller KR (2006) Optimizing spatio-temporal filters for improving Brain-Computer Interfacing. *NIPS* 18
- Guyon I, Weston J, Barnhill S, Vapnik V (2002) Gene selection for cancer classification using support vector machines. *Machine Learning* 46:389–422
- Koles ZJ, Lazar MS, Zhou SZ (1990) Spatial patterns underlying population differences in the background EEG. *Brain Topography* 2(4):275–284
- Kübler A, Nijboer F, Mellinger J, Vaughan T, Pawelzik H, Schalk G, McFarland D, Birbaumer N, Wolpaw J (2005) Patients with ALS can use sensorimotor rhythms to operate a braincomputer interface. *Neurology* 64:1775–1777
- Lal T, Schröder M, Hinterberger T, Weston J, Bogdan M, Birbaumer N, Schölkopf B (2004) Support vector channel selection in BCI. *IEEE TBME* 51(6):1003–1010
- Lal T, Schröder M, Hill N, Hinterberger T, Mellinger J, Rosenstiel W, Hofmann T, Birbaumer N, Schölkopf B (2005a) A brain computer interface with online feedback based on magnetoencephalography. *ICML* 22:465–472
- Lal TN, Hinterberger T, Widman G, Schröder M, Hill NJ, Rosenstiel W, Elger CE, Schölkopf B, Birbaumer N (2005b) Methods towards invasive human brain computer interfaces. *NIPS* 17
- Lemm S, Blankertz B, Curio G, Müller KR (2004) Spatio-spectral filters for robust classification of single trial EEG. *IEEE TBME* 52(9):993 – 1002
- Müller KR, Anderson CW, Birch GE (2003) Linear and nonlinear methods for brain-computer interfaces. *IEEE TNSRE* 11(2):165–169

# Optimizing Spectral Filters for Single Trial EEG Classification

Ryota Tomioka<sup>1,2</sup>, Guido Dornhege<sup>2</sup>, Guido Nolte<sup>2</sup>,  
Kazuyuki Aihara<sup>1</sup>, and Klaus-Robert Müller<sup>2</sup>

<sup>1</sup> Dept. Mathematical Informatics, IST, The University of Tokyo,  
Hongo 7-3-1, Bunkyo-ku, Tokyo, 113-8656, Japan

<sup>2</sup> Fraunhofer FIRST.IDA, Kekuléstr. 7, 12 489 Berlin, Germany

**Abstract.** We propose a novel spectral filter optimization algorithm for the single trial ElectroEncephaloGraphy (EEG) classification problem. The algorithm is designed to improve the classification accuracy of Common Spatial Pattern (CSP) based classifiers. The algorithm is based on a simple statistical criterion, and allows the user to incorporate any prior information one has about the spectrum of the signal. We show that with a different preprocessing, how a prior knowledge can drastically improve the classification or only be misleading. We also show a generalization of the CSP algorithm so that the CSP spatial projection can be recalculated after the optimization of the spectral filter. This leads to an iterative procedure of spectral and spatial filter update that further improves the classification accuracy, not only by imposing a spectral filter but also by choosing a better spatial projection.

## 1 Introduction

A Brain-Computer Interface (BCI) system provides a direct control pathway from human intentions to computer. Recently, a considerable amount of effort has been done in the development of a BCI system [1,2,3,4,5]. We will be focusing on non-invasive, electroencephalogram (EEG) based BCI systems. Such a device can give disabled people direct control over a neuroprosthesis or over a computer application as tools for communicating solely by their intentions that are reflected in their brain signals (e.g. [2]).

Recently, machine learning approaches to BCI have proven to be effective by making the subject training required in the classical framework unnecessary and compensating for the high inter-subject variability.

The task in this approach is to extract subject-specific discriminability patterns from high-dimensional spatio-temporal signals. With respect to the topographic patterns of brain rhythm modulations, the Common Spatial Patterns (CSP) (see [6,7]) algorithm has proven to be very useful in extracting discriminative spatial projections. On the other hand, the frequency band on which the classifier operates is either selected manually or unspecifically set to a broad band filter [7,3]. Naturally, an automatic method also for the selection of the



frequency band is highly desirable [8,9]. Here, we present a method for the spectral filter optimization problem, which is based on a simple statistical criterion. The proposed method is capable of handling arbitrary prior filters based on neurophysiological insights. The proposed method outperforms broad-band filtered CSP in most datasets. Moreover, a detailed validation shows how much of the gain is obtained by the theoretically obtained filter and how much is obtained by imposing a suitable prior filter. Based on the spectral filter obtained by the proposed method, one can also recalculate the CSP projection; this leads to iterative updating of spatio-spectral filter. We show that further improvements in the classification accuracy can be achieved by iteratively updating.

## 2 The Algorithm

Let us denote by  $X \in \mathbb{R}^{d \times T}$  the EEG signal of a single trial of imaginary motor movement<sup>1</sup>, where  $d$  is the number of electrodes and  $T$  is the number of sampled time-points in a trial. We consider a binary classification problem where each class, e.g. right or left hand imaginary movement, is called positive (+) or negative (−) class. The task is to predict the class label for a single trial  $X$ .

Throughout this paper, we use a feature vector, namely *log-power features*, defined as follows:

$$\phi_j(X; \mathbf{w}_j, B_j) = \log \mathbf{w}_j^\dagger X B_j B_j^\dagger X^\dagger \mathbf{w}_j \quad (j = 1, \dots, J), \quad (1)$$

where the upper-script  $\dagger$  denotes a conjugate transpose or a transpose for a real matrix,  $\mathbf{w}_j \in \mathbb{R}^d$  is a spatial projection that projects the signal into a single dimension and  $B_j \in \mathbb{R}^{T \times T}$  denotes the linear time-invariant temporal filter, which is an identity matrix in the case of conventional CSP algorithm. The training of a classifier is composed of two steps. In the first step, the coefficients  $\mathbf{w}_j$  and  $B_j$  are optimized. In the second step, the Linear Discriminant Analysis (LDA) classifier is trained on the feature vector.

We use Common Spatial Pattern (CSP) algorithm [6,7], a well known technique for the spatial filter optimization. Given a set of trials and the labels  $\{X_i, y_i\}_{i=1}^n$  ( $X_i \in \mathbb{R}^{d \times T}$ ,  $y_i \in \{+1, -1\}$ ), the CSP is formulated so that the projection maximize the power of the projected signal for one class and minimize that for the other class. This principle can be written as follows:

$$\max_{\mathbf{w} \in \mathbb{R}^d} \frac{\mathbf{w}^\dagger \langle X X^\dagger \rangle_+ \mathbf{w}}{\mathbf{w}^\dagger \langle X X^\dagger \rangle_- \mathbf{w}}. \quad (2)$$

where the angled brackets denote expectation within a class. Furthermore, it is known that the solution is easily obtained by solving the following generalized eigenvalue problem:

$$\Sigma^+ \mathbf{w} = \lambda \Sigma^- \mathbf{w}, \quad (3)$$

---

<sup>1</sup> For simplicity, we assume that the trial mean is already subtracted and the signal is scaled by the inverse square root of the number of time-points. This can be achieved by a linear transformation  $X = \frac{1}{\sqrt{T}} X_{\text{original}} (I_T - \frac{1}{T} \mathbf{1}\mathbf{1}^\dagger)$ .

where we call  $\Sigma^c := \langle XX^\dagger \rangle_c \in \mathbb{R}^{d \times d}$  ( $c \in \{+, -\}$ ) the sensor covariance matrix. The eigenvector corresponding to the largest eigenvalue of Eq. (3) is the optimum of the problem (2). In addition, the minimization of the problem (2) gives another projection that may be equivalently powerful in the classification. Moreover, it is often observed that the second or the third eigenvectors have fairly good discrimination. Therefore, we take the eigenvectors corresponding to the largest and the smallest  $n_{\text{of}}$  eigenvalues for each side. Thus,  $J = 2n_{\text{of}}$  in Eq. (1).

Given a spatial projection, the next question is how to optimize the temporal filter  $B$  in Eq. (1). We formulate this problem in the frequency domain, because any time-invariant operation  $B$  is diagonalized in the frequency domain. We state the problem as follows:

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & \frac{\langle s(\mathbf{w}, \boldsymbol{\alpha}) \rangle_+ - \langle s(\mathbf{w}, \boldsymbol{\alpha}) \rangle_-}{\sqrt{\text{Var}[s(\mathbf{w}, \boldsymbol{\alpha})]_+ + \text{Var}[s(\mathbf{w}, \boldsymbol{\alpha})]_-}}, \\ \text{s.t. } \quad & \alpha_k \geq 0 \quad (\forall k = 1, \dots, T), \end{aligned} \tag{4}$$

where we write the power spectrum of the signal projected with  $\mathbf{w}$  as  $\{s_k(\mathbf{w})\}_{k=1}^T$ , the spectrum of the filter as  $\boldsymbol{\alpha} := \{\alpha_k\}_{k=1}^T$  and  $s(\boldsymbol{\alpha}, \mathbf{w}) := \sum_{k=1}^T \alpha_k s_k(\mathbf{w})$ .

The optimal filter coefficient is explicitly written as follows:

$$\alpha_k^{(+)\text{opt}} \propto \begin{cases} \frac{\langle s_k(\mathbf{w}) \rangle_+ - \langle s_k(\mathbf{w}) \rangle_-}{\text{Var}[s_k(\mathbf{w})]_+ + \text{Var}[s_k(\mathbf{w})]_-} & \langle s_k(\mathbf{w}) \rangle_+ - \langle s_k(\mathbf{w}) \rangle_- \geq 0, \\ 0 & \text{otherwise,} \end{cases} \tag{5}$$

because the spatio-temporally filtered signal  $s(\mathbf{w}, \boldsymbol{\alpha})$  is linear with respect to the spectral filter coefficients  $\{\alpha_k\}_{k=1}^T$  and we additionally assume that the signal is a stationary Gaussian process, where the frequency components are independent to each other for a given class label; thus  $\text{Var}[s(\mathbf{w}, \boldsymbol{\alpha})]_c = \sum_{k=1}^T \alpha_k^2 \text{Var}[s_k(\mathbf{w})]_c$ . Note that the labels (+ and -) are exchanged for  $\{\alpha_k^{(-)\text{opt}}\}_{k=1}^T$ , the filter for the “-” class. The norm of the filter coefficients cannot be determined from the problem (4). Therefore, in practice we normalize the coefficients so that they sum to one.

Furthermore, we can incorporate our prior knowledge on the spectrum of the signal during the task. This can be achieved by generalizing from Eq. (5) to:

$$\alpha_k^{(c)} = \left( \alpha_k^{(c)\text{opt}} \right)^q \cdot (\beta_k)^p \quad (c \in \{+, -\}), \tag{6}$$

where  $\{\beta_k\}_{k=1}^T$  denotes the prior information, which we define specific to a problem (see Sec. 3). The optimal values for  $p$  and  $q$  should depend on the data, preprocessing, and the prior information  $\{\beta_k\}_{k=1}^T$ . Therefore one can choose them by cross validation.

Now, using the CSP projection  $\mathbf{w}$  and the optimized spectral filter  $\boldsymbol{\alpha}$ , the log-power feature (Eq. (1)) is written as follows:

$$\phi_j(X; \mathbf{w}_j, \boldsymbol{\alpha}_j) = \log \sum_{k=1}^T \alpha_k^{(j)} \mathbf{w}_j^\dagger \hat{\mathbf{x}}_k \hat{\mathbf{x}}_k^\dagger \mathbf{w}_j \quad (j = 1, \dots, J), \tag{7}$$

where  $\hat{\mathbf{x}}_k \in \mathbb{C}^d$  denotes the  $k$ -th component of the Fourier transform of  $X$ .

### 3 Results

#### 3.1 Experimental Setup

**Data Acquisition.** We use 162 datasets of motor-imagery BCI experiment from 29 healthy subjects. Each dataset contains EEG signal recorded during 70-600 (varying from a dataset to another at median 280) trials of one of the pairwise combinations of three motor imagination tasks, namely right hand (R), left hand (L) or foot (F) (see [9,5] for the detail).

**Preprocessing of the Signals.** We band-pass filter the signal from 7-30Hz and cut out the interval of 500-3500ms after the appearance of the visual cue on the screen, which instructs the subject which imagination to perform, from the continuous EEG signal for each execution of imaginary movement as a trial. Only in Sec. 3.3, we also use the signal without the band-pass filter step, in order to investigate the effect of assuming this band (7-30Hz) on the design of a filter; except the band-pass filtering, the signal was equally processed as described above.

**Classification.** We use the log-power feature (Eq. (1)) with  $n_{of} = 3$  features for each class and LDA as a classifier.

**Prior Information.** We test two prior filters  $\{\beta_k\}_{k=1}^T$ , namely:

- with the wide-band 7-30Hz assumption:

$$\beta_k = I_k^{[7, 30]} \cdot (\langle s_k(\mathbf{w}) \rangle_+ + \langle s_k(\mathbf{w}) \rangle_-) / 2, \quad (8)$$

- without the assumption:

$$\beta_k = (\langle s_k(\mathbf{w}) \rangle_+ + \langle s_k(\mathbf{w}) \rangle_-) / 2, \quad (9)$$

where  $I_k^{[7, 30]}$  is an indicator function that takes value one only in the band 7-30Hz, and otherwise zero. Since we have already band-pass filtered the signal in order to calculate CSP, it is reasonable to restrict the resulting filter to take values only within this band. The second term, which is the average activity of two classes, express our understanding that in the motor imagery task, good discrimination is most likely be found at frequency bands that correspond to strong rhythmic activities, i.e.,  $\mu$ - and  $\beta$ -rhythms; the modulation of these rhythms is known as Event Related Desynchronization (ERD) and well studied. However, this might not be the case if we don't suppose the interesting signal to lie within the 7-30Hz interval as in the second prior filter (Eq. (9)). The comparison is shown in Sec. 3.3.

Furthermore, since the optimal filter (Eq. (5)) and the prior filter (Eqs. (8) or (9)) scale with powers  $-1$  and  $1$  of the spectrum, respectively, we reparameterize the hyperparameters as  $p = p' + q'$  and  $q = q'$ . Thus, if  $p' = c$  the filter scales with the power  $c$  regardless of which  $q'$  is chosen. Therefore, the contributions of the scale and the discriminability are separated in the new parameterization.

Now, for the prior filter (8), using  $p'$ , the scaling exponent of the filter and  $q'$ , the intensity of the label information, we can write Eq. (6) as follows:

$$\alpha_k \propto I_k^{[7, 30]} \cdot \begin{cases} \left( \frac{(s_k^{(+)} - s_k^{(-)}) (s_k^{(+)} + s_k^{(-)})}{v_k^{(+)} + v_k^{(-)}} \right)^{q'} \cdot (s_k^{(+)} + s_k^{(-)})^{p'} & s_k^{(+)} - s_k^{(-)} \geq 0 \\ 0 & \text{otherwise.} \end{cases} \quad (10)$$

where the following short-hands are used:  $s_k^{(c)} := \langle s_k(\mathbf{w}) \rangle_c$  and  $v_k^{(c)} := \text{Var} [s_k(\mathbf{w})]_c$ . The filter with the prior filter (9) is simply Eq. (10) without the indicator  $I_k^{[7, 30]}$ .

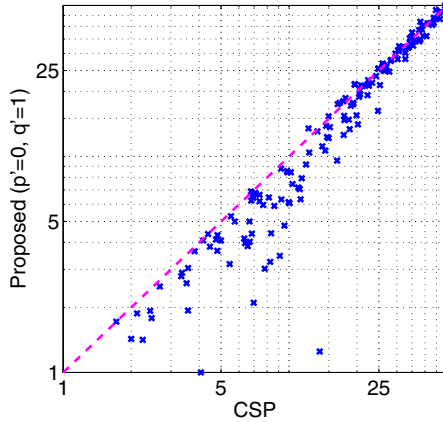
### 3.2 Comparison with CSP

First, we compare the proposed method with the prior filter (Eq. (8)) with conventional CSP [6,7] algorithm. The spatial projection for the proposed method is the CSP itself. Therefore, the only difference is that we incorporate a non-homogeneous weighting of the spectrum (see Eq. (7)). The hyperparameters for the proposed method were fixed at  $p' = 0$  and  $q' = 1$  ( $p = 1$  and  $q = 1$  in the original parameterization), which corresponds to the direct product of Eqs. (5) and (8).

Figure 1 shows the  $10 \times 10$  cross-validation errors of CSP and the proposed method for each dataset as a single point. Data-points lower than the diagonal correspond to datasets where the proposed method outperforms CSP.

As a visualization, the spectral filter corresponding to conventional CSP, the theoretically obtained filter (Eq. (5)), the prior filter (Eq. (8)) and the resulting spectral filter are shown in Fig. 2 for a CSP projection in a single dataset. The conventional CSP is purely an operation in the spatial domain. Therefore, as a spectral filter it has a flat spectrum as shown in the top-left corner. The proposed method (bottom-left corner) is a combination of the theoretically obtained filter (Eq. (5)) shown in the bottom center and the prior filter (Eq. (8)) shown in the bottom-right corner. The theoretically obtained filter (Eq. (5)) scales with the power  $-1$  of the spectrum. This means that it compares frequency components with different ranges in a fair manner; the signal is first scaled down by a factor  $1 / \sqrt{v_k^{(+)} + v_k^{(-)}}$  (whitening) and then summed with a weighting  $(s_k^{(+)} - s_k^{(-)})_+ / \sqrt{v_k^{(+)} + v_k^{(-)}}$ . This effect is clearly seen in the bottom center. The theoretically obtained filter has two peaks, one approximately at 12Hz and the other at 24Hz, although in the original scale the difference between two classes around 24Hz is hardly seen (top center). The scale  $-1$  is also favorable from another point of view, namely invariance; one can apply an arbitrary (non-zero) spectral filter to the signal before calculating Eq. (5) yet the effect is canceled out by Eq. (5). On the other hand, since the signal is already band-pass filtered from 7-30Hz, a prior filter is always peaked at frequency components corresponding to strong rhythmic activities (e.g.,  $\mu$ - or  $\beta$ -rhythm) regardless of whether they have discriminative information or not. The resulting filter (bottom left), which is a direct product of the two filters in this case (because  $p = 1$  and  $q = 1$ ), has

two peaks but the peak at 12Hz is larger than the peak at 24Hz. The optimal combination of the theoretical optimum and the prior filter is discussed in the next session.



**Fig. 1.** The  $10 \times 10$  cross-validation errors of conventional CSP and the proposed method on 162 datasets. Points lower than the diagonal correspond to datasets where the proposed method outperforms CSP. The conventional CSP weights the spectrum homogeneously (Eq. (1) with  $B_j = I_T$  or Eq. (7) with  $\alpha_k = 1 (\forall k)$ ) while the proposed method weights the spectrum according to Eq. (10)). The hyperparameters were fixed at  $p' = 0$  and  $q' = 1$  (the direct product of Eqs. (5) and (8)).

### 3.3 Comparison of the Two Prior Filters

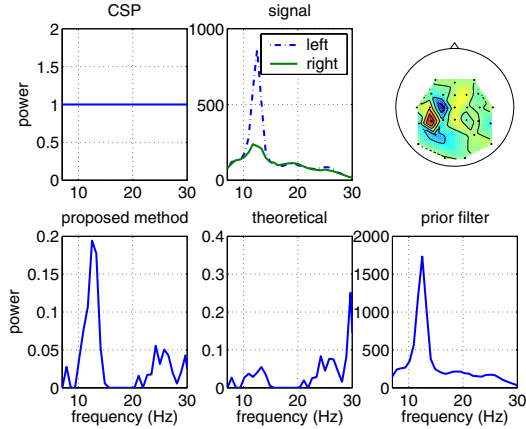
In the previous section, we have shown that the combination of the theoretical optimum (Eq. (5)) and the prior filter (Eq. (8)) outperforms CSP in most datasets. However, it is still unclear whether the hyperparameters  $p' = 0$  and  $q' = 1$  are optimal or not. Furthermore, the range of validity of the prior filter (Eq. (8)) is not clear.

Therefore, in this section, we investigate two prior filters (Eqs. (8) and (9)). The first prior filter (Eq. (8)) focuses on the strong activity within the interval 7-30Hz. The second filter (Eq. (9)) also focuses on the strong activity but without the constraint, i.e., the wide-band 7-30Hz assumption.

In order to compare these two prior filters appropriately, we take the following two steps approach. In the first step, we optimize the spatial filter. Each dataset is band-pass filtered from 7-30Hz and the CSP projection with  $n_{\text{of}} = 3$  patterns for each class is calculated on the whole dataset and fixed. In the second step, in order to investigate the optimal design of a spectral filter, we conduct a cross validation on the signal without pre-filtering.

Note that this validation differs from that in the previous section in two folds: firstly, the optimization of the spatial filter was done on the whole dataset in the first step and fixed during the validation, secondly, the spatial filter was calculated on the pre-filtered signal but applied on the signal without pre-filtering.

Figures 3(a) and 3(b) show the contour plot of the average cross-validation error for all combinations of  $p' \in [-2, 2]$  and  $q' \in [0, 8]$  on a 0.2 interval grid for

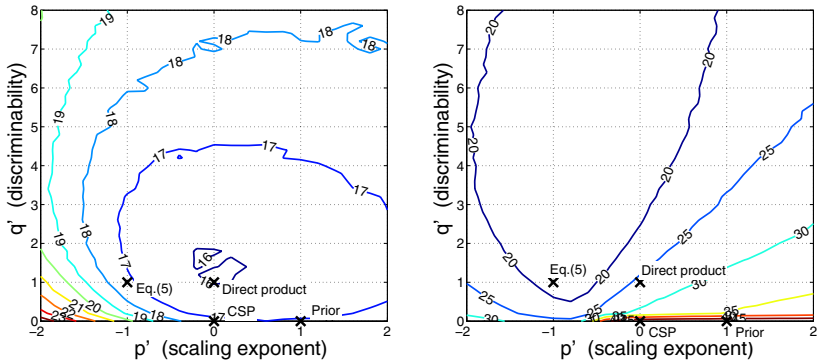


**Fig. 2.** (top center) The class-averaged spectrum of the original signal projected with a CSP projection shown in the top right corner. (top left) The conventional CSP in the spectral domain. (bottom left) The filter spectrum obtained by the proposed method. (bottom center) The theoretically obtained filter (Eq. (5)). (bottom right) The prior filter (Eq. (8)). (top right) The CSP projection topographically mapped on a head viewed from above. The head is facing the top of the paper.

the two prior filters (Eqs. (8) and (9)), respectively. Figure 3(a) shows that the non-homogeneous weighting of the spectrum improves the classification accuracy ( $p' = 0, q' = 1$  is better than  $p' = 0, q' = 0$ ), which is consistent with Fig. 1, and incorporating the prior filter is also effective ( $p' = 0, q' = 1$  is better than  $p' = -1, q' = 1$ ). On the other hand, Fig. 3(b) shows a completely different picture. Since the wide-band assumption is not adopted in the prior filter (9), it weights not only  $\mu$ - or  $\beta$ -band but also the strong brain activity lower than 7Hz, which does not correspond to motor imagery task or even which cannot be considered a rhythmic activity. Thus the prior information is not so much useful anymore. The basin of the classification error is now shifted to approximately  $p' = -1$  where the spectrum is whitened. The theoretical optimum (Eq. (5)) is now in the region that gives minimum classification error. Note that however the overall error is lower in Fig. 3(a) compared to that in Fig. 3(b). Therefore, in practice the wide-band assumption appears to help though the aim of this section was to show that in general, without the wide-band assumption, it is necessary that one scales down the filter inversely to the power of the signal (Eq. (5)).

### 3.4 Iterative Update of Spatio-spectral Filter

Although we have so far used the CSP projection as a spatial projection and focused on the optimization of the spectral filter, one can also recalculate the spatial projection after the optimization of the spectral filter. In order to incorporate the spectral filter, we generalize the definition of the sensor covariance matrix  $\Sigma^c$ . Since the covariance matrix of the temporally filtered signal can be written



(a) with the wide-band 7-30Hz as- (b) without the assumption (see  
 sumption (see Eq. (8)). Eq. (9)).

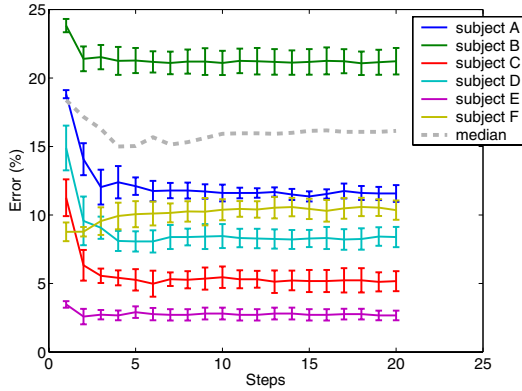
**Fig. 3.** The contour plot of the average cross-validation errors over 162 datasets in the two dimensional hyperparameter space. Unlike in Sec. 3.2 or in Sec. 3.4 the cross-validation was carried out on the signal without pre-filtering with a pre-computed spatial pattern. Points corresponding to the CSP, the theoretically derived filter (Eq. (5)), the prior filter, and the direct product of the two filters ( $p' = 0, q' = 1$ ) are marked. The cross validation is  $4 \times 4$ .

as  $V(\alpha) := \sum_{k=1}^T \alpha_k V_k$ , where  $V_k = \hat{\mathbf{x}}_k \hat{\mathbf{x}}_k^\dagger$  ( $k = 1, \dots, T$ ) are the cross spectrum matrices, we solve the generalized eigenvalue problem  $\Sigma^+(\alpha)\mathbf{w} = \lambda \Sigma^-(\alpha)\mathbf{w}$  instead of Eq. (3) for the recalculation of the spatial projection, where  $\Sigma^+(\alpha) := \langle V(\alpha) \rangle_c$ . Starting from uniform spectral coefficients  $\alpha_k = 1$  ( $\forall k$ ), we alternately update the spectral filter and the spatial projection until convergence, because both steps depend on each other.

Figure 4 shows the improvements in the cross-validation error by iteratively updating spatio-spectral filter for six subjects. The odd steps correspond to the spatial projection updates and the even steps are spectral updates. Since the first step is CSP with homogeneous spectral filter and the second step is the proposed method without recalculation of the spatial projection, one can see that the major improvements occur by imposing a spectral filter (the second step). However, further improvements after the third step (e.g., in subject C) were observed for many datasets. For some subjects (e.g., in subject F) no improvement in the cross-validation was observed, most likely due to artifacts whose effects are not localized in the frequency spectrum (e.g. blinking, chewing or other muscle movements).

### 4 Conclusion

In this paper, we have proposed a novel spectral filter optimization technique for CSP [6,7] based single-trial EEG classifiers. The method is formulated in the spectral domain, based on a simple statistical criterion. Thus the result is highly interpretable. The method is capable of handling arbitrary prior filter,



**Fig. 4.** The cross-validation errors of the iterative updating method for each step are shown for six subjects from very good classification accuracy (subject E) to moderate accuracy (subject B). The median over 162 datasets is also shown (dashed line). The hyperparameters were fixed at  $p' = 0$  and  $q' = 1$  (the direct product of Eqs. (5) and (8)). The odd steps correspond to spatial projection updates and the even steps are spectral filter updates. Note that the first step is the conventional CSP itself and the second step is the proposed method without the recalculation of spatial projection.

which one can design based on the neuro-physiological understanding of the EEG signal during the task.

The cross validation on 162 BCI datasets show improved classification accuracy compared to the conventional CSP [6,7]. In comparison to CSP, we have shown that the non-homogenous weighting of the spectrum improves the classification accuracy.

Moreover, we have investigated the best combination of the theoretically obtained filter (Eq. (5)) and the prior filter. We have tested two prior filters, namely the filter with the wide-band 7-30Hz assumption (Eq. (8)) and that without the assumption (Eq. (9)). We have found that with the wide-band assumption (Eq. (8)), the best combination is achieved approximately at  $p' = 0, q' = 1$ , which corresponds to the direct product of the theoretically obtained filter (Eq. (5)) and the prior filter (Eq. (8)); it is better than the conventional CSP ( $p' = 0, q' = 0$ ), the theoretical optimum alone ( $p' = -1, q' = 1$ ) or the prior filter ( $p' = 1, q' = 0$ ). However, without the wide-band assumption, the prior filter, which assumes the discrimination to be found at frequency regions that is strongly active, fails because the activity below 7Hz will tend to dominate without contributing to discriminability. On the other hand, the theoretically optimal scale  $p' = -1$ , which whitens the signal, has proved to be favorable without the assumption. Thus, the prior filter is only valid with the wide-band assumption. In fact, we note that either CSP or the best combination  $p' = 0, q' = 1$  already incorporates this prior knowledge that “strong activity implies good discrimination”, because both of them have the scale  $p' = 0$ .

Furthermore, we have tested an iterative updating algorithm of spatio-spectral filter. We have generalized the CSP algorithm to incorporate a non-homogeneous



weighting of the cross spectrum matrices. The spatial filter and the spectral filter were updated alternately. We have found by cross validating each step of iteration that although for most datasets the major drop in the cross-validation error is observed when a spectral filter (the second step) was imposed on the original CSP pattern (the first step), further improvements in the cross-validation error were observed after the recalculation of CSP pattern (the third step) in many datasets.

The proposed method gives highly interpretable spatial filter naturally because we solve the generalized CSP problem. In addition, the spectral representation of the temporal filter is favorable not only from the interpretability but also from providing possibility to incorporate any prior information about the spectral structure of the signal as we have demonstrated in Sec. 3.

**Acknowledgment.** This research was partially supported by MEXT, Grant-in-Aid for JSPS fellows, 17-11866, 2006, by BMBF-grant FKZ 01IBB02A and by the PASCAL Network of Excellence (EU # 506778).

## References

1. Wolpaw, J.R., Birbaumer, N., McFarland, D.J., Pfurtscheller, G., Vaughan, T.M.: Brain-computer interfaces for communication and control. *Clin. Neurophysiol.* **113** (2002) 767–791
2. Birbaumer, N., Ghanayim, N., Hinterberger, T., Iversen, I., Kotchoubey, B., Kübler, A., Perelmouter, J., Taub, E., Flor, H.: A spelling device for the paralysed. *Nature* **398** (1999) 297–298
3. Pfurtscheller, G., Neuper, C., Guger, C., Harkam, W., Ramoser, R., Schlögl, A., Obermaier, B., Pregenzer, M.: Current trends in Graz brain-computer interface (BCI). *IEEE Trans. Rehab. Eng.* **8**(2) (2000) 216–219
4. Blankertz, B., Dornhege, G., Schäfer, C., Krepki, R., Kohlmorgen, J., Müller, K.R., Kunzmann, V., Losch, F., Curio, G.: Boosting bit rates and error detection for the classification of fast-paced motor commands based on single-trial EEG analysis. *IEEE Trans. Neural Sys. Rehab. Eng.* **11**(2) (2003) 127–131
5. Blankertz, B., Dornhege, G., Krauledat, M., Müller, K.R., Kunzmann, V., Losch, F., Curio, G.: The Berlin Brain-Computer Interface: EEG-based communication without subject training. *IEEE Trans. Neural Sys. Rehab. Eng.* **14**(2) (2006) in press.
6. Koles, Z.J.: The quantitative extraction and topographic mapping of the abnormal components in the clinical EEG. *Electroencephalogr. Clin. Neurophysiol.* **79** (1991) 440–447
7. Ramoser, H., Müller-Gerking, J., Pfurtscheller, G.: Optimal spatial filtering of single trial EEG during imagined hand movement. *IEEE Trans. Rehab. Eng.* **8**(4) (2000) 441–446
8. Lemm, S., Blankertz, B., Curio, G., Müller, K.R.: Spatio-spectral filters for improved classification of single trial EEG. *IEEE Trans. Biomed. Eng.* **52**(9) (2005) 1541–1548
9. Dornhege, G., Blankertz, B., Krauledat, M., Losch, F., Curio, G., Müller, K.R.: Combined optimization of spatial and temporal filters for improving brain-computer interfacing. *IEEE Trans. Biomed. Eng.* (2006) accepted.

# Probabilistic De Novo Peptide Sequencing with Doubly Charged Ions

Hansruedi Peter, Bernd Fischer, and Joachim M. Buhmann

Institute of Computational Science  
ETH Zurich, Switzerland  
<http://www.ml.inf.ethz.ch>

**Abstract.** Sequencing of peptides by tandem mass spectrometry has matured to the key technology for proteomics. Noise in the measurement process strongly favors statistical models like `NOVOHMM`, a recently published generative approach based on factorial hidden Markov models [1,2]. We extend this hidden Markov model to include information of doubly charged ions since the original model can only cope with singly charged ions. This modification requires a refined discretization of the mass scale and, thereby, it increases its sensitivity and recall performance on a number of datasets to compare favorably with alternative approaches for mass spectra interpretation.

## 1 Introduction

Proteins control all metabolic processes in biological cells of living organisms. To understand the dynamics and interaction of these processes biologists have to identify all involved proteins, i.e. determine their sequence and their abundance. From a computer science perspective a protein is a string over an alphabet of 20 amino acids. Even for short strings, this combinatorics generates an incredibly high number of possible combinations. The identification of these sequences is becoming increasingly important also for medical research. In biomarker discovery based on gene expression micro-arrays physiologists rely on tissue samples from the affected tissue. Recent research results provide evidence that protein based biomarker discovery can be performed solely on blood samples in the near future [3]. This diagnostics will then be a great research step in early detection of cancer and we might even call it “remote sensing of cancer”.

The most promising method of high-throughput protein sequencing is tandem mass spectrometry. The proteins are biologically broken in short sequences by enzymatic digestion. For each peptide a mass spectrum is generated that includes mass measurements of fragments of the peptide. In addition, a rough estimate of the peptide mass is available from liquid chromatography (LC/MS). Peptide sequencing aims at inferring the underlying amino acid sequence given the mass spectrum and the mass of the peptide.

Today, the genomes of many organisms have already been sequenced. Given the DNA sequence we can compile a database of protein sequences that can be transcribed and translated from these genes. In a first analysis step biologists will infer the peptide sequences using side information of the protein databases ([4,5]). This procedure, although conceptually very appealing, has some difficulties: (i) the databases are still

incomplete or have some errors from the sequencing process; (ii) there can be unknown splice variants or even unknown genes in the DNA; (iii) there exist post-translational modifications. Due to combinatorial limits one can not enumerate all possibilities of genes, splice variants and post-translational modifications. Therefore a cascading search strategy is recommended [6], where less and less side-information is used to narrow down the possible amino acid sequences. As a complement to database search, peptide sequences can be inferred from mass spectrometry data in a *de novo* fashion and we are following this strategy here. The only information used in *de novo* peptide sequencing is the alphabet of 20 amino acids and the mass spectra as input.

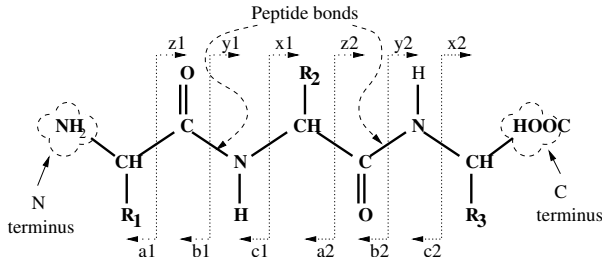
LUTEFISK [7,8] and PEAKS [9] are two widely used *de novo* peptide sequencing methods. LUTEFISK creates a weighted graph using a simple scoring scheme. The sequence is the shortest path in the generated graph. PEAKS creates a similar weighted graph and generates a candidate list of peptides by searching the weighted graph. A refined search is performed afterwards on the candidate list. Dancik [10] proposed the first probabilistic scoring scheme for *de novo* sequencing. It simply estimates the fragmentation pattern of the peptides at a small number of positions. The scoring scheme was improved by adding a noise model and a Bayesian network to model the fragmentation patterns [11]. Fischer *et al.* [1,2] proposed a generative hidden Markov model (NovoHMM) of mass spectra. This model can only describe singly charged fragment ions which is a clear shortcoming since about 10-25 percent of the ions are doubly or triply charged. We will substantially extend NovoHMM to include doubly charged fragment ions by refining the discretization of the mass scale.

The next section will summarize the essentials of tandem mass spectrometry as far as our modelling is concerned. The hidden Markov model is presented in section 3. In section 4 we give our extensions of a refined discretization and the inclusion of information from doubly charged fragment ions. The new model is tested on different datasets (sec. 5). It clearly outperforms all its competitors. Furthermore, we show that the model can be applied to triply charged peptide ions.

## 2 Tandem Mass Spectrometry

The proteomics process pipeline based on mass spectrometry contains the following steps: first, the proteins are digested with an enzyme (typically Trypsin). This chemical process yields a sample of small peptides. A peptide  $P$  consists of a short sequence of  $n$  amino acid residues  $P = a_1a_2a_3, \dots, a_n$ , with an additional H-atom at the N-terminus and an OH-group at the C-terminus. In figure 1 a small peptide composed of three amino acids is depicted. Typically peptides have 10-20 amino acid residues and they are separated by liquid chromatography.

In a first measurement step the mass of the peptides can be read out from a mass spectrum. Ions (peptides) from a small mass window are selected and they are fragmented in typically two pieces by collision with a noble gas. As shown in figure 1, the most common fragment ions which are denoted as a-,b-,c-,x-,y- and z-ions, are generated by breaking the peptide backbone. The tandem mass spectrum contains peaks at mass positions corresponding to the different fragment ions. The inference of the underlying sequence given a mass spectrum is the goal of peptide sequencing.



**Fig. 1.** A simple peptide with three amino acids, an additional H-atom at the N-terminus and an OH-group at the C-terminus. The amino acids are connected by peptide bonds. Depending on the internal link a peptide is broken, the corresponding fragment ions are called a-/b-/c-ions when containing the N-terminus respectively x-/y-/z- ions when containing the C-terminus.

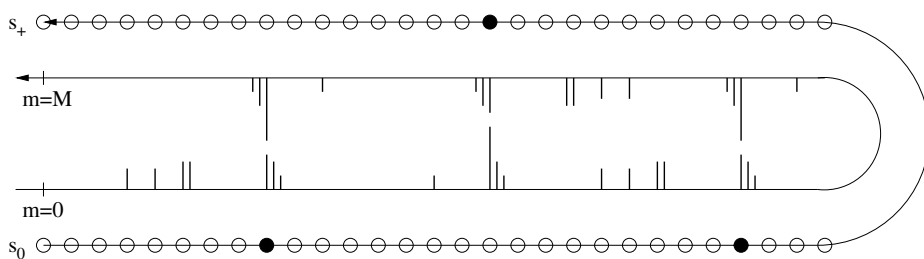
To measure a peptide by mass spectrometry, the peptide has to be charged. Most common the peptide is double positively charged. After the fragmentation most fragment ions are singly charged. In 10–25 percent of the cases the resulting fragment ions are doubly charged. The mass spectra consists of a list of mass-charge ratios  $m/z$  and their corresponding intensities. Doubly charged ions appear at half of the position of the corresponding singly charged ions. This shift induces long-range dependencies that are difficult to model for standard hidden Markov models, e.g., the model of Fischer *et al.* [2] can only describe singly charged ions. In this paper we will extend the model for doubly charged ions.

### 3 Factorial Hidden Markov Model (NovoHMM)

In the hidden Markov model of [2] the (time-)step is 1 elementary mass unit (the mass of a proton or neutron). With the transition probabilities a distribution over all amino acid sequences is modeled. The hidden random variables (representing the peptide sequence) have 2375 states. For each amino acid there are as many states as the amino acid has elementary particles. This chain of states for each amino acid is a counter for the mass. In NovoHMM only peptide sequences with the given peptide mass are considered. This constraint is implemented by introducing a positive and negative end state. After as many time steps as the peptide mass indicates, the model ends in the positive end state for all sequences that obey the peptide mass constraint.

Hypothetical spectra, composed of mass peaks from prefix fragment ions only, can be interpreted in straight forward way: the different possible fragment ions (a-, b-, c-ions) are assigned to specific counter positions on the mass scale. The b-ion is placed at the last counter state of each amino acid. The a-ion (It is a b-ion without a carbon monoxide) is placed 28 counter states before the b-ion. Additional fragments like neutral water loss can be modeled in the same fashion.

There exists, however, a strong dependence between prefix and suffix ions via the total peptide mass. For every prefix fragment ion with mass  $m$  there exists with high probability a mass peak of a suffix fragment ion at mass position  $M - m$ , where  $M$  is the total peptide mass. To overcome these long-range dependencies, the Markov model



**Fig. 2.** The internal mirror symmetry of the problem is illustrated by folding the spectrum in the middle. The Markov chain models a sequence with four amino acids. The filled circles correspond to the amino acid boundaries. Around each amino acid boundary a peak pattern is generated once for the  $N$ -terminal fragments and once for the  $C$ -terminal fragments.

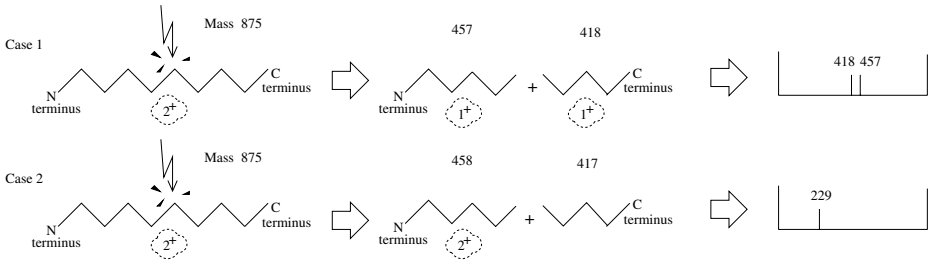
is duplicated (see fig. 2). One Markov model starts at the beginning of the sequence and another Markov model starts in the end of the sequence. In the figure a peptide sequence with four amino acids is depicted. The black dots denote the amino acid boundaries. The boundary at mass  $m$  generates a peak pattern around mass  $m$  and  $M - m$ . The left part of figure 5 shows the graphical model. Both emission variables  $x_i$  and  $x_{M-i}$  depend on the sequence variables  $s_i$  and  $s_{M-i}$ . The new model is still a hidden Markov model but it has a factorial structure [12].

The number of hidden states of the new model is squared compared to the simple model. To reduce the model complexity and the runtime the model is approximated by a mixture model. A second set of hidden variables is introduced: The binary variables  $B_i$  decide for each peak  $x_i$  if it is generated by a prefix- or a suffix fragment. The emission probability is now a mixture of a prefix-distribution and a suffix-distribution instead of the joint distribution of both.

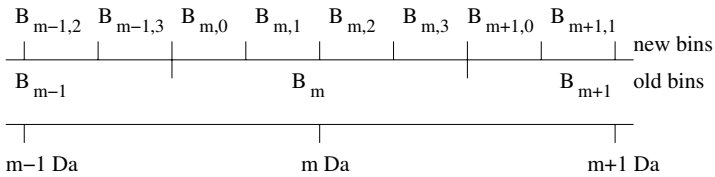
## 4 Doubly Charged Fragment Ions

Before the fragmentation process, most peptides are double positively charged. During the collision with the noble gas, the peptide ions are broken in two (or more) fragments as depicted in figure 3. In the figure two possible outcomes of the fragmentation process are shown. In the first case the peptide ion breaks in two parts. Both parts are single positively charged. This can result in two peaks in the spectrum, e.g. at mass-over-charge ratios 418 and 457. In the second case the peptide is broken in the same prefix and suffix parts, but now the prefix is double positively charged and the suffix is not charged. The prefix peak will appear at mass-over-charge ratio 229. Since a mass spectrometer can only measure charged ions, a suffix ion peak does not appear in the spectrum.

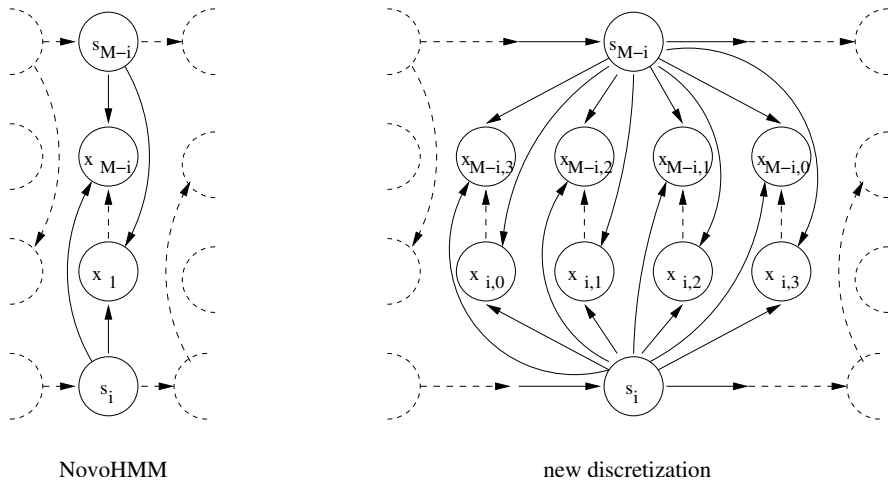
In a first step, the discretization of the spectra has to be refined to handle the smaller distance between the doubly charged fragment peaks. In NOVOHMM, a spectrum is discretized into bins of approximately 1 Dalton width, where the peaks are placed (by definition) in the middle of the bin. Doubly charged fragment ions are spaced with a minimal distance of one half mass-over-charge unit. To consider this effect, we have to discretize the leftmost half of the spectrum in a different way, as shown in figure 4.



**Fig. 3.** Cleavage of a peptide by collision with a noble gas. Two possibilities are shown: splitting into two singly charged fragments (Case 1); and into a doubly charged and an uncharged fragment (Case 2). On the right, the corresponding mass spectra are depicted.

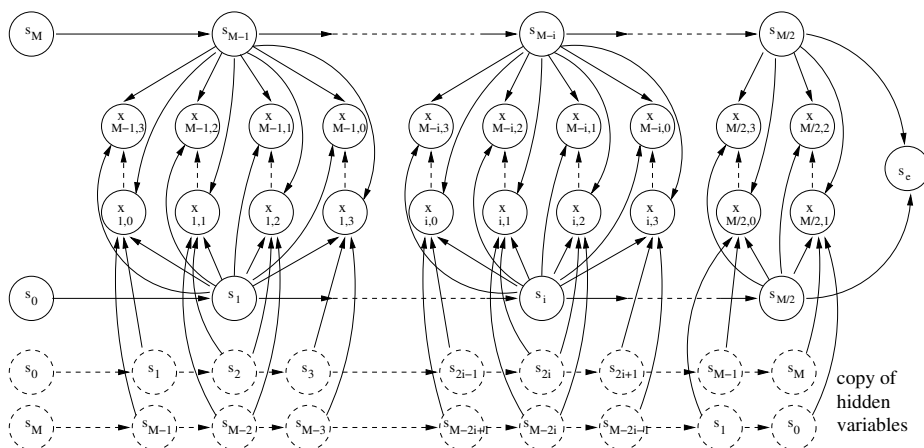


**Fig. 4.** Discretization of a spectrum for the new model (new bins) and NovoHMM (old bins). In the new discretization, each bin is divided into four sub-bins.



**Fig. 5.** Part of the dependency structure of NovoHMM (at the left) with the corresponding part in the model with the new discretization. Each emission variable is replaced by four variables for a finer discretization of the spectrum.

In the graphical model (fig. 5) the hidden variables (representing the sequence) are not changed. The emission variables are replaced by four variables each. Note that doubly charged ions can only occur in the first half of the spectrum. Therefore it would be enough to discretize the first half of the spectrum in this way. For reasons of model consistency we decided to discretize the complete spectrum in such a way. As we will see later in the experimental section, the new discretization itself will improve the prediction accuracy of the model.

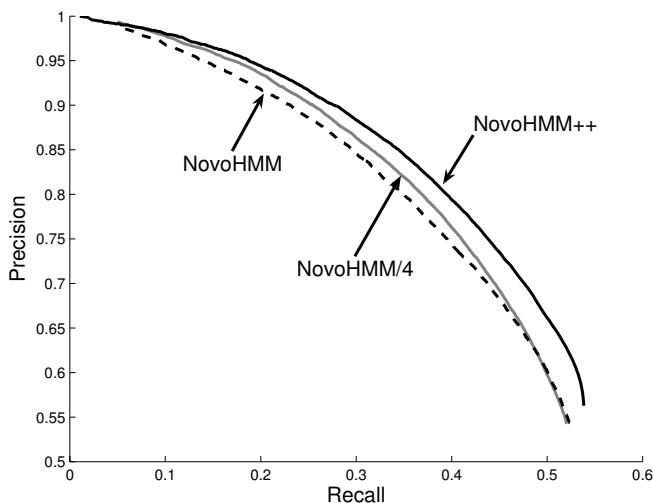


**Fig. 6.** The dependency structure of the factorial HMM for the model including doubly charged ions. The sequence variables in the last two rows are the same as in the model above. They are only copied for visualization reasons. Each emission variable in the first half of the spectrum depends now on four different sequence variables.

In a second step we can now include the information from doubly charged ions. The emission variables representing the first half of the spectrum have additional dependencies. The variable  $x_m$  does additionally depend on  $s_{2m}$  and  $s_{M-2m}$  which encode the corresponding doubly charged prefix and suffix ion. The upper half of figure 6 shows the graphical model for the new discretized factorial hidden Markov model. To draw the dependencies in a more transparent way, we have copied the sequence variables in the plot. The lower two rows of variables are the same as the sequence variables in the upper model. The problem is again approximated by a mixture model. For each mass bin in the lower half of the spectrum there is now an assignment variable with four different states: 1-charged prefix, 1-charged suffix, 2-charged prefix and 2-charged suffix. The assignment problem is solved by the expectation-maximization algorithm. In the E-step the expectation over all hidden variables (sequence and assignment variables) is computed. Since this is computationally intractable, we decomposed the E-step into two stages. In the first stage, the expectation of the assignment variables is estimated; in the second stage the expectation of the sequence variables is computed by the forward-backward algorithm as proposed in [2].

## 5 Experiments

To justify our model extension of `NovoHMM`, we first present experimental results for the model which includes doubly charged fragment ions. Furthermore, the prediction quality of this model (`NovoHMM++`) is compared to the prediction quality of `NovoHMM` and other *de novo* sequencing methods on two different datasets. The first data set is composed by Frank and Pevzner [11]. It contains 972 spectra in the training dataset and 280 spectra in the test dataset. The second dataset contains more than 5000 spectra from an unpublished proteomics experiment. We performed 10-fold cross validation on the second dataset and we used the proposed splitting in training and test spectra for the first dataset.



**Fig. 7.** The precision-recall curves achieved for cross-validation on a the second dataset

When nothing else is mentioned, we consider two amino acids to be correct if the difference in mass position of an amino acid in the original spectrum and in the predicted spectrum is less than or equal to 2.5 Dalton (see [11]). Furthermore, no distinction has been made between leucine (L) and isoleucine (I), or between lysine (K) and glutamine (Q), as they have almost the same mass and cannot be distinguished by low mass resolution tandem mass spectrometry. The precision value is defined as the number of correct amino acids divided by the number of predicted amino acids. The recall value is defined as the number of correct amino acids divided by the true number of amino acids. In the plots shown later in this section the precision and recall values are varied by changing a threshold on the posterior value computed with the forward-backward algorithm.

To simplify notation, we introduce names for the different versions of our algorithm:

- `NovoHMM` – The original version of `NovoHMM` as it was presented in [2].
- `NovoHMM/4` – `NovoHMM` with fine discretization in quarters of a Dalton, but without a model for doubly charged fragment ions.



- **NovoHMM++** - **NovoHMM** with fine discretization in quarters of a Dalton and with an additional model for doubly charged fragment ions.

Figure 7 depicts the precision-recall curves on the second dataset. The readers can easily convince themselves that the new discretization alone improves already the prediction accuracy. The accuracy is further boosted by information from doubly charged ions as they are included in **NovoHMM++**.

**NovoHMM++** was also tested on a dataset with 1020 triply charged peptides from the second dataset. With an average length of 2414.5 Dalton these peptides are clearly longer than the peptides in the datasets of doubly charged peptides. A triply charged peptide is expected to split up into a singly charged fragment ion and a doubly charged fragment ion. We achieved a precision of 0.216 at a recall of 0.200. At a first glance, these precision and recall values may look low, but one has to consider the substantial lengths of the peptides in the dataset which clearly influences the predictive power of the algorithm. For comparison, **NovoHMM** achieved about 10% precision and recall on this dataset.

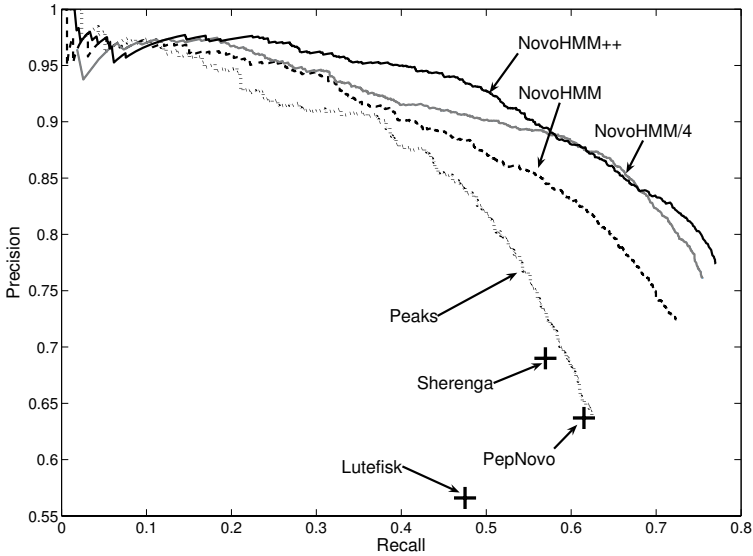
**NovoHMM++** clearly outperforms all competing algorithms on the first dataset (see table 1). Table 2 presents the relative frequency of correctly labeled subsequences of length at least  $x$ . Whereas **PepNovo** is slightly superior for short subsequences, **NovoHMM++** clearly exceeds all its competitors for long peptides. In figure 8, the precision-recall curves of **NovoHMM++** and **NovoHMM/4** are compared with other *de novo* sequencing methods. The closer the curves approach (1, 1) values, the better is the recall performance. In general, **NovoHMM/4** and **NovoHMM++** are relatively

**Table 1.** Comparison of the performance of our algorithms with other *de novo* sequencing methods on the first dataset

Algorithm	<b>NovoHMM++</b>	<b>NovoHMM/4</b>	<b>NovoHMM</b>	<b>PepNovo</b>	<b>Sherenga</b>	<b>Peaks</b>	<b>Lutefisk</b>
Correctly predicted symbols (of 2935)	<b>2293</b>	2244	2160	2063	1673	1943	1394
Precision	<b>0.787</b>	0.770	0.737	0.727	0.690	0.673	0.566
Recall	<b>0.781</b>	0.765	0.736	0.703	0.570	0.662	0.475

**Table 2.** Percentage of correct subsequences of length at least  $x$

Algorithm	Predictions with correct subsequences of at least							
	$x = 3$	$x = 4$	$x = 5$	$x = 6$	$x = 7$	$x = 8$	$x = 9$	$x = 10$
<b>NovoHMM++</b>	0.943	0.861	0.796	<b>0.714</b>	<b>0.621</b>	<b>0.539</b>	<b>0.429</b>	<b>0.300</b>
<b>NovoHMM/4</b>	0.943	<b>0.871</b>	0.793	0.686	0.607	0.532	0.396	0.264
<b>NovoHMM</b>	0.911	0.829	0.743	0.632	0.546	0.464	0.336	0.229
<b>PepNovo</b>	<b>0.946</b>	<b>0.871</b>	<b>0.800</b>	0.654	0.525	0.411	0.271	0.193
<b>Sherenga</b>	0.821	0.711	0.564	0.364	0.279	0.207	0.121	0.071
<b>Peaks</b>	0.889	0.814	0.689	0.575	0.482	0.371	0.275	0.179
<b>Lutefisk</b>	0.661	0.521	0.425	0.339	0.268	0.200	0.104	0.057



**Fig. 8.** The precision-recall curves for our algorithms compared with other de novo sequencing methods. Tolerance criterion: exact elementary mass.

close to each other, whereas for high recall values `NovoHMM++` seems to be superior to `NovoHMM/4`.

The model described in the paper includes doubly charged prefix- and suffix ions in a generative hidden Markov model to interpret mass spectra. In addition, we tested the same model including just doubly charged prefix ions or just doubly charged suffix ions. The performance, when including suffix ions only, is almost the same as when we incorporate both ions. On the other hand, with the doubly charged prefix ions only, the model does not demonstrate the performance of `NovoHMM++`. Therefore we conclude that mainly the doubly charged suffix ions matter are responsible for the performance increase of the model. This observation is biologically plausible, since peptides (and thus suffix ions) digested by Trypsin often end with lysine (K) or arginine (R). These amino acids are known to attract positively charged ions.

## 6 Conclusion

Factorial Hidden Markov Models provide a flexible framework to explain mass spectra which are gathered from proteomics experiments. An extension of `NovoHMM` is presented in this paper which contains an additional model for doubly charged fragment ions and a refined discretization. This new model `NovoHMM++` increases the accuracy of the predicted sequences by up to 5% in precision and recall on different datasets. On a benchmark test [11], `NovoHMM++` substantially and significantly outperform the

most prominent *de novo* sequencing algorithms in terms of prediction accuracy. In addition, NovoHMM++ was shown to reliably explain also mass spectra containing triply charged peptides.

**Acknowledgement.** We thank Jonas Grossmann, Sacha Baginski and Wilhelm Gruissem (Inst. of Plant Sciences, ETH Zurich) for providing the mass spectrometry data.

## References

1. Fischer, B., Roth, V., Buhmann, J.M., Grossmann, J., Baginsky, S., Gruissem, W., Roos, F., Widmayer, P.: A hidden markov model for de novo peptide sequencing. In: Neural Information Processing Systems. Volume 17., USA, MIT press (2005) 457–464
2. Fischer, B., Roth, V., Roos, F., Grossmann, J., Baginsky, S., Widmayer, P., Gruissem, W., Buhmann, J.M.: NovoHMM: A hidden markov model for de novo peptide sequencing. *Analytical Chemistry* **77**(22) (2005) 7265–7273
3. Chen, R., Pan, A., Brentnall, T., Aebersold, R.: Proteomic profiling of pancreatic cancer for biomarker discovery. *Molecular and Cellular Proteomics* **4**(4) (2005) 523–533
4. Eng, J.K., McCormack, A.L., III., J.R.Y.: An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database. *American Society for Mass Spectrometry* **5**(11) (1994) 976–989
5. Hirose, M., Hoshida, M., Ishikawa, M., Toya, T.: Mascot: multiple alignment system for protein sequences based on three-way dynamic programming. *Computer Applications in the Bioscience* **9**(2) (1993) 161–167
6. Sadygov, R.G., Cociorva, D., III., J.R.Y.: Large-scale database searching using tandem mass spectra: Looking up the answer in the back of the book. *Nature Methods* **1**(3) (2004) 195–202
7. Taylor, J.A., Johnson, R.S.: Sequence database searches via de novo peptide sequencing by tandem mass spectrometry. *Rapid Communications in Mass Spectrometry* **11** (1997) 1067–1075
8. Taylor, J.A., Johnson, R.S.: Implementation and uses of automated de novo peptide sequencing by tandem mass spectrometry. *Analytical Chemistry* **73** (2001) 2594–2604
9. Ma, B., Zhang, K., Hendrie, C., Liang, C., Li, M., Doherty-Kirby, A., Lajoie, G.: Peaks: Powerful software for peptide de novo sequencing by tandem mass spectrometry. *Rapid Communications in Mass Spectrometry* **17**(20) (2003) 2337–2342
10. Dancik, V., Addona, T.A., Clauser, K.R., Vath, J.E.: De novo peptide sequencing via tandem mass spectrometry. *Journal of Computational Biology* **6** (1999) 327–342
11. Frank, A., Pevzner, P.: Pepnovo: De novo peptide sequencing via probabilistic network modeling. *Analytical Chemistry* **77**(4) (2005) 964–973
12. Zoubin Ghahramani, M.I.J.: Factorial hidden markov models. *Machine Learning* **29**(2–3) (1997) 245 – 273

# Direct Estimation of the Wall Shear Rate Using Parametric Motion Models in 3D

Markus Jehle<sup>1,2</sup>, Bernd Jähne<sup>1,2</sup>, and Ulrich Kertzscher<sup>3</sup>

<sup>1</sup> Interdisciplinary Center for Scientific Computing (IWR), Im Neuenheimer Feld 368,  
D-69120 Heidelberg, Germany

<sup>2</sup> Institute for Environmental Physics (IUP), Im Neuenheimer Feld 229,  
D-69210 Heidelberg, Germany

{Markus.Jehle, Bernd.Jaehne}@iwr.uni.heidelberg.de

<sup>3</sup> Labor für Biofluidmechanik, Charité, Universitätsmedizin  
D-14050 Berlin, Germany

Ulrich.Kertzscher@charite.de

**Abstract.** We present a new optical-flow-based technique to estimate the wall shear rate using a special illumination technique that makes the brightness of particles dependent on the distance from the wall. The wall shear rate is derived directly (that means, without previous calculation of the velocity vector field) from two of the components of the velocity gradient tensor which in turn describes the kinematics of fluid flows up to the first order. By incorporating this into a total least squares framework, we can apply a further extension of the structure tensor technique. Results obtained both from synthetical and real data are shown, and reveal a substantial improvement compared to conventional techniques.

## 1 Introduction

Optical-flow [1] based techniques were established as powerful tools in the field of fluid flow analysis in recent years [2]. Using these methods it is possible to evaluate image sequences using continuous tracer, for example concentration [3] or heat [4], or rigid particles [5]. Under certain circumstances optical-flow based techniques are superior to correlation based techniques, such as Particle Image Velocimetry (PIV) [6], which is quite common in experimental fluid mechanics. For a comparative analysis of correlation based techniques and optical-flow based techniques in the field of computer vision see [7]. In this paper we adopt a novel approach based on an extended version of the generalized brightness change constraint equation which was applied to an image sequence recorded in the context of biofluidmechanics. Short explanations of the medical application and of the considered experiment are given in this introduction.

The investigation of the flow near the wall of a blood vessel or an artificial organ is of great interest, since a close relationship is known to exist between the characteristics of the flow such as magnitude and direction of the wall shear stress, and biological phenomena such as thrombus formation or atherosclerotic events. The wall shear stress can be considered as the force, which the viscous fluid exerts tangentially on the wall-surface. It plays an important role, since it influences the structure and function of the endothelial cells as well as the behavior of platelets. The measurement of the wall shear

stress is a requirement to our understanding of atherosclerotic events and also for the ability of avoiding thrombus generation in artificial organs.

[8] points out, that previous to her work, there existed no technique, which was capable to measure the influence of the flow close to the wall on biological and pathological events. Firstly this is due to the fact, that we deal with instationary flows at curved walls, secondly, it is not sufficient, to conduct pointwise measurements, but it is necessary to yield temporally and spatially resolved 2D-information of the wall shear stress, which has to be extracted taking into account the 3D-nature of the flow field.

The method presented here is based on the observation and the digital recording of buoyant, light-reflecting, spherical particles suspended within the fluid. The particles are all exactly 300  $\mu\text{m}$  in diameter. In contrast to conventional 2D-PIV the entire flow near the wall is illuminated from the outside with monochromatic diffuse light, so that all particles near the wall become visible. A dye is added to the fluid, which limits the penetration depth of the light into the flow model according to Beer-Lambert's law. The intensity respective gray value  $g_p$  of the light approaching the particle is

$$g_p(z) = g_0 \exp -z/\tilde{z}_* ,$$

where  $g_0$  is the light's intensity before penetrating into the fluid,  $z$  is the distance of the particle's surface from the wall, and  $\tilde{z}_*$  is the penetration depth (Figure 1). The light is reflected by the particle, and passes through the distance  $z$  again, before approaching the wall with the intensity

$$g(z) = g_p(z) \exp -z/\tilde{z}_* = g_0 \exp -2z/\tilde{z}_* = g_0 \exp -z/z_* , \tag{1}$$

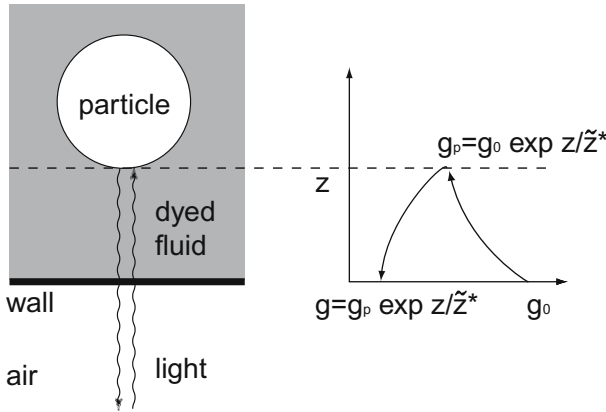
where an effective penetration depth  $z_* = \tilde{z}_*/2$  was introduced for convenience. Within the illuminated layer the particles appear more or less bright, depending on their normal distance to the wall: Particles near the wall appear brighter, i. e. have a higher gray value than particles farther away from the wall. The correlation between the gray value of a particle and its distance to the wall, which is expressed in terms of the hypothetical grayvalue  $g_0$  of the particle at the wall and  $z_*$ , can be assessed experimentally.

If the concentration of the dye, the illumination and the size of the particles are chosen properly, the particles closest to the wall fall within a region where the velocity distribution is considered to be proportional to the wall distance. This permits the calculation of the wall shear stress  $\tau_w$  according to Newton's shear stress formula, using the measured velocity component tangential to the wall  $u$ , the normal distance to the wall  $z$  and the dynamic viscosity of the fluid  $\eta$ :

$$\tau_w = \eta \left[ \frac{du}{dz} \right]_{z=0} \approx \eta \frac{\Delta u}{\Delta z} .$$

[8] separates the near-wall flow in several layers by means of gray-value-thresholding. For each layer, which is characterized by a distinct distance from the wall, the motion of the particle can be determined with a conventional PIV algorithm. This results in a vector field  $u(z)$  for each layer, from which the wall shear stress can be derived.

We present an optical flow-based approach for analyzing image sequences recorded using the technique described above, which leads to following benefits compared to the analysis proposed in [8]:



**Fig. 1.** A monochromatic beam of light penetrates the dyed fluid with the intensity  $g_0$ , and hits the particle with intensity  $g_p$  after covering the distance  $z$ . After reflecting, it passes through the dye again, and hits the camera sensor with the intensity  $g$ . The intensity decrease can be calculated using Beer-Lambert’s law.

- Since PIV is a correlation-based technique, the particle density in image sequences suitable for a PIV-based analysis has to be sufficiently high. For this reason, [8] chooses the width of the distinct layers relatively large, taking into account inaccuracies. We will overcome this by regarding every particle individually.
- Since PIV in its simple form is a 2D technique, there is no possibility of estimating out-of-plane motions. In our method brightness changes, i. e. motions perpendicular to the image plane, will be incorporated in the underlying equations.
- In order to estimate the wall shear stress [8] has to calculate the velocity vector fields first. Our method delivers the wall shear rate directly, without previous estimation of the vector fields.

## 2 Estimation of Depth and Velocity

We estimate the distance of the particle’s surface from the wall by eliminating  $z$  in Beer-Lambert’s law (1):

$$z = z_*(\ln g_0 - \ln g) .$$

In order to estimate the particle’s velocity, we consider two cases. First we assume that the suspended particles move parallel to the wall, so that  $z$  won’t change. The grayvalue then remains constant for all times, and we can apply the brightness change constraint equation (BCCE) to obtain the optical flow:

$$dg/dt = (\nabla g)^T \mathbf{f} + g_t = 0 . \tag{2}$$

The optical flow represents the components of the particle’s velocity parallel to the wall:  $\mathbf{f} = (u, v)^T$ . Secondly if the particles don’t move parallel to the wall, i. e. with  $z$  not constant, the grayvalue will change with time, according to:

$$\frac{dg}{dt} = -\frac{g_0}{z_*} \frac{dz}{dt} \exp -z/z_* = -\frac{1}{z_*} \frac{dz}{dt} = -\frac{w}{z_*} g ,$$

where the component of the particle’s velocity perpendicular to the wall  $w = dz/dt$  is introduced. From this we are able to construct some kind of *generalized brightness change constraint equation* GBCCE, as proposed in [9]:

$$(\nabla g)^T \mathbf{f} + g_t = -(w/z_*)g \quad , \quad (3)$$

which can be written as a scalar product of the data vector  $\mathbf{d}$  and the parameter vector  $\mathbf{p}$ :

$$\mathbf{d} \cdot \mathbf{p}^T = (g_x, g_y, g/z_*, g_t) \cdot (u, v, w, 1)^T = 0 \quad ,$$

where  $g_x, g_y$  and  $g_t$  denote the partial derivatives of the gray values with respect to the spatial and temporal dimensions. To sufficiently constrain the equation system, we assume a constant  $\mathbf{p}$  over a small spatio-temporal neighborhood, surrounding the location of interest containing  $n$  pixels. With the data matrix  $\mathbf{D} = (\mathbf{d}_1, \dots, \mathbf{d}_n)^T$  replacing the data vector, the equation-system can be solved in a total least squares (TLS) sense akin to the structure tensor [7]:

$$\|\mathbf{D}\mathbf{p}\|_2 = \mathbf{p}^T \mathbf{D}^T \mathbf{D} \mathbf{p} \rightarrow \min \quad .$$

with  $\mathbf{p}^T \mathbf{p} = 1$  to avoid the trivial solution  $\mathbf{p} = 0$ . The Eigenvector  $\mathbf{e} = (e_1, e_2, e_3, e_4)^T$  to the smallest eigenvalue of the generalized structure tensor

$$\mathbf{D}^T \mathbf{D} = \begin{pmatrix} \langle g_x \cdot g_x \rangle & \langle g_x \cdot g_y \rangle & \langle g_x \cdot g \rangle / z_* & \langle g_x \cdot g_t \rangle \\ \langle g_x \cdot g_y \rangle & \langle g_y \cdot g_y \rangle & \langle g_y \cdot g \rangle / z_* & \langle g_y \cdot g_t \rangle \\ \langle g_x \cdot g \rangle / z_* & \langle g_y \cdot g \rangle / z_* & \langle g \cdot g \rangle / z_*^2 & \langle g \cdot g_t \rangle / z_* \\ \langle g_x \cdot g_t \rangle & \langle g_y \cdot g_t \rangle & \langle g \cdot g_t \rangle / z_* & \langle g_t \cdot g_t \rangle \end{pmatrix}$$

represents the sought after solution to the problem. In this notation local spatiotemporal averaging using a binomial filter is represented by pointed brackets. In the case of full flow, which means no aperture problem is present, the parameter vector is given by  $\mathbf{p} = 1/e_4(e_1, e_2, e_3)^T$ . The structures (here: particles imaged to circles of diameter smaller than 5 pixels) contain no edges, whose dimensions are greater than the size of the neighborhood which is chosen for velocity estimation (here:  $33 \times 33$  pixels). So the image sequences recorded with the technique described above generally exhibit no aperture problem.

Image sequences recorded with the technique described above generally exhibit no aperture problem, so we consider only full flow.

### 3 Estimation of the Wall Shear Rate

In the introduction we emphasized that knowledge about the spatially distribution of the wall shear stress is essentially for understanding biofluidmechanics near the wall of a blood vessel or an artificial organ. In this chapter we show a method which delivers the wall shear rate directly. The wall shear rate is the wall shear stress divided by the dynamic viscosity of the fluid. We derive the wall shear rate by selecting certain components of the velocity gradient tensor at the wall. This object describes the kinematics of the fluid up to first order completely. The velocity gradient tensor may be regarded as

a generalization of the concept of the affine parameterization of 2D-optical flow fields, which will be recapitulated briefly in the following:

The optical flow  $\mathbf{f}(\mathbf{x}, t)$  may be expanded to a first order Taylor series in the vicinity of  $(\mathbf{x}_0, t_0)$  [10]:

$$\mathbf{f}(\mathbf{x}, t) \approx \mathbf{f}(\mathbf{x}_0, t_0) + \begin{pmatrix} \partial u / \partial x & \partial u / \partial y \\ \partial v / \partial x & \partial v / \partial y \end{pmatrix} \mathbf{x} + \begin{pmatrix} \partial u / \partial t \\ \partial v / \partial t \end{pmatrix} t \equiv \mathbf{t} + \mathbf{A}\mathbf{x} + \mathbf{a}t .$$

The BCCE (2) supplemented by this parameterization yields the *extended brightness change constraint equation (EBCCE)*:

$$(\nabla g)^T (\mathbf{t} + \mathbf{A}\mathbf{x} + \mathbf{a}t) + g_t = 0 . \quad (4)$$

Geometric transformations of the local neighborhood may be computed from the components of the matrix  $\mathbf{A}$ . Examples are divergence or vorticity:

$$\operatorname{div}(\mathbf{f}) = \partial u / \partial x + \partial v / \partial y \quad \text{or} \quad \operatorname{rot}(\mathbf{f}) = \partial u / \partial v - \partial v / \partial x .$$

In the following we consider 3D physical flow fields. We apply the notation  $\mathbf{u} \equiv (u_1, u_2, u_3)^T \equiv (u, v, w)^T$  for the 3D velocity vector at the 3D position  $\mathbf{x} \equiv (x_1, x_2, x_3)^T \equiv (x, y, z)^T$ . A flow field  $\mathbf{u}(\mathbf{x})$  can be extended to a first order Taylor series in the vicinity of  $(\mathbf{x}_0, t_0)$ :

$$u_i(x_j, t) \approx u_i(x_{j,0}, t_0) + \frac{\partial u_i}{\partial x_j} x_j + \frac{\partial u_i}{\partial t} t .$$

We made use of Einstein's summation convention, and  $i$  and  $j$  are defined from 1 to 3. In vector-matrix-notation this reads:

$$\mathbf{u}(\mathbf{x}, t) \approx \mathbf{s} + \Gamma \mathbf{x} + \mathbf{b}t ,$$

where  $\mathbf{s}$  is a 3D-translation,  $\Gamma = (\gamma_{ij}) = (\partial u_i / \partial x_j)$  is the  $3 \times 3$ -velocity gradient tensor which is essentially the Jacobian, and  $\mathbf{b}$  is a 3D-acceleration. By using an alternative formulation of the GBCCE (3)

$$(g_x, g_y)^T \cdot (u, v) + g_t + (w/z_*)g = (g_x, g_y, g/z_*)^T \cdot (u, v, w) + g_t = (\tilde{\nabla} g)^T \mathbf{u} + g_t = 0 ,$$

where  $\tilde{\nabla}$  is an augmented gradient, the 3D-parametrisation can be incorporated into a 3D-EBCCE:

$$(\tilde{\nabla} g)^T \cdot (\mathbf{s} + \Gamma \mathbf{x} + \mathbf{b}t) + g_t = 0 . \quad (5)$$

From the components of the matrix  $\Gamma$  important physical quantities of the local neighborhood in the flow field can be computed, like

- the vorticity vector  $\omega_k = \epsilon_{ijk} \gamma_{ij}$ ,
- the strain rate tensor  $s_{ij} = 1/2(\gamma_{ij} + \gamma_{ji})$  or
- the dissipation rate  $\epsilon = -2\nu \overline{s_{ij} s_{ij}} = -\nu \overline{(\gamma_{ij} + \gamma_{ji}) \gamma_{ij}}$ .



If we assume that we have pure 2D-flow  $u = u(x, y)$ ,  $v = v(x, y)$  and  $w = 0$ , (5) reduces to the optical flow-parametrization case (4).

In the following we address the case of uniform wall parallel shear flow, i. e.  $u = u(z)$ ,  $v = v(z)$  and  $w = 0$ . The only non-vanishing components of the velocity gradient tensor  $\gamma_{ij} = \partial u_i / \partial x_j$  are

$$\frac{\partial u}{\partial z} = \frac{\partial u_1}{\partial x_3} = \gamma_{13} \quad \text{and} \quad \frac{\partial v}{\partial z} = \frac{\partial u_2}{\partial x_3} = \gamma_{23} .$$

Therefore the 3D-EBCCE (4) can be rewritten to

$$\tilde{\nabla} g^T \cdot \left( 0 + \begin{pmatrix} 0 & 0 & \gamma_{13} \\ 0 & 0 & \gamma_{23} \\ 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix} + 0 \right) + g_t = 0 , \tag{6}$$

which can be transformed to a scalar product of the data vector  $\mathbf{d}$  and the parameter vector  $\mathbf{p}$  after some simple algebraic manipulations:

$$\mathbf{d} \cdot \mathbf{p}^T = (g_x, g_y, g_t) \cdot (\gamma_{13}, \gamma_{23}, 1)^T = 0 .$$

Starting from this scalar product, we can construct an expanded structure tensor, similar to the way presented in Section 2:

$$\mathbf{D}^T \mathbf{D} = \begin{pmatrix} \langle g_x z \cdot g_x z \rangle & \langle g_x z \cdot g_y z \rangle & \langle g_x z \cdot g_t \rangle \\ \langle g_x z \cdot g_y z \rangle & \langle g_y z \cdot g_y z \rangle & \langle g_y z \cdot g_t \rangle \\ \langle g_x z \cdot g_t \rangle & \langle g_y z \cdot g_t \rangle & \langle g_t \cdot g_t \rangle \end{pmatrix} .$$

By performing an eigen-decomposition we obtain an estimation for the parameter vector to  $\mathbf{p} = 1/e_3(e_1, e_2)^T = (m_{13}, m_{23})^T$  in the full flow case.

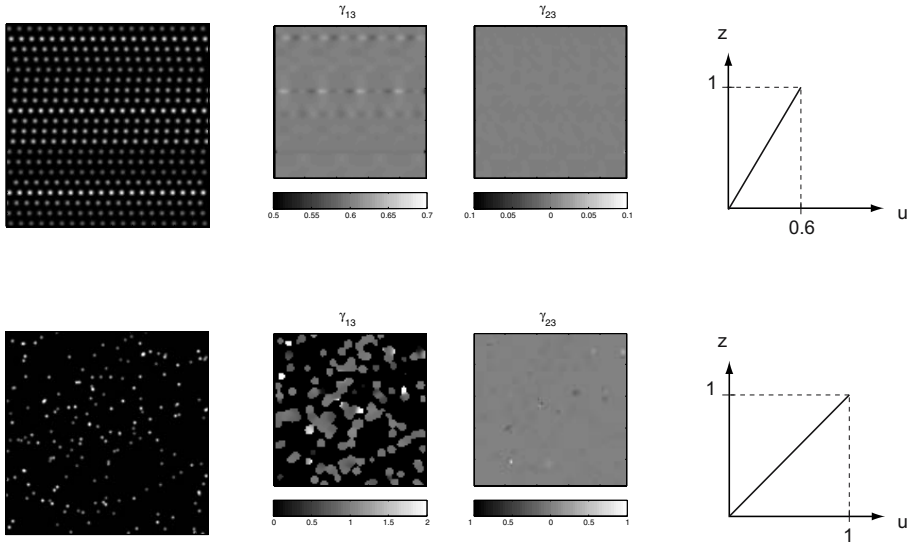
## 4 Results

We apply the analysis presented in Section 3 to synthetically generated and to real acquired image sequences. All sequences are evaluated by using the EBCCE based on the special case of wall parallel shear flow (6).

### 4.1 Synthetic Data

The following image sequences are generated by providing a uniform, wall-parallel 3D-Flow:  $u(z) = \gamma_{13}z$ ,  $v = w = 0$ . The flow is texturized using Gaussian intensity distributions of equal maximum intensity and equal maximum variance, representing the particles. The  $z$ -position of the particles is indicated by attenuating the maximum intensity of the Gaussians according to Beer-Lambert’s law (1).

The first synthetic image sequence contains particles, which are distributed in such a way, that they never will overlap each other: The particles are arranged in rows; each particle in one row having the same depth, and therefore the same brightness and the same speed (Figure 2, top, left). Here the wall shear rate  $m_{13}$  is exactly 0.6, which is



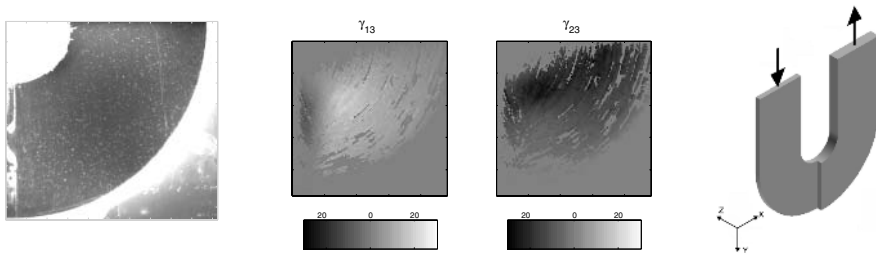
**Fig. 2.** Example images of the synthetic image sequences (left), maps of the wall shear rates estimated with our algorithm (center) and velocity profiles, which deliver the ground truth for the wall shear rates (right). The distinct image sequences are described in the text.

indicated by the profile  $u(z)$  (Figure 2, top, right), and the wall shear rate  $\gamma_{23}$  vanishes. Our algorithm yields wall shear rates, which are displayed in Fig. 2, top, center. The ground truth is reproduced very well. Slight deviations occur, where there are several particle-rows are adjacent, moving with approximately the same speed. The reason for these deviations is the fact that the spatio-temporal neighborhood is of limited size (in this case  $65 \times 65$  pixels).

In the second synthetic image sequence the particles are randomly distributed (Figure 2, bottom, left), moving so that they follows the wall shear rates  $m_{13} = 1$  and  $\gamma_{23} = 0$ . The estimated wall shear rates are mapped in Fig. 2, bottom, middle. As a result of the fact, that overlappings may occur, there are regions, where our algorithm produces significant deviations from the ground truth. This is evidence, that our model fails in the presence of multiple motions.

## 4.2 Real Data

The analyzed image sequence was recorded by [8]. To examine the applicability of the method presented in Section 1 for the investigation of complex flows, a U-shaped channel with a rectangular cross-section and a step was constructed (Figure 3, right). In combination with the bending, the step in the cross-section generates a complex flow which detaches from the wall. Figure 3, left shows a sample image of the recorded sequence. Since the diameter of the spheres is significantly larger than the penetration-depth, only the particles close to the wall are imaged and no overlapping particles are recorded. Therefore, the problem of multiple-motions does not occur in this situation.



**Fig. 3.** Example image of the recorded image sequence (left), maps of the wall shear rates estimated with our algorithm (center) and geometry of the U-shaped channel with the arrows denoting the direction of the flow (right)

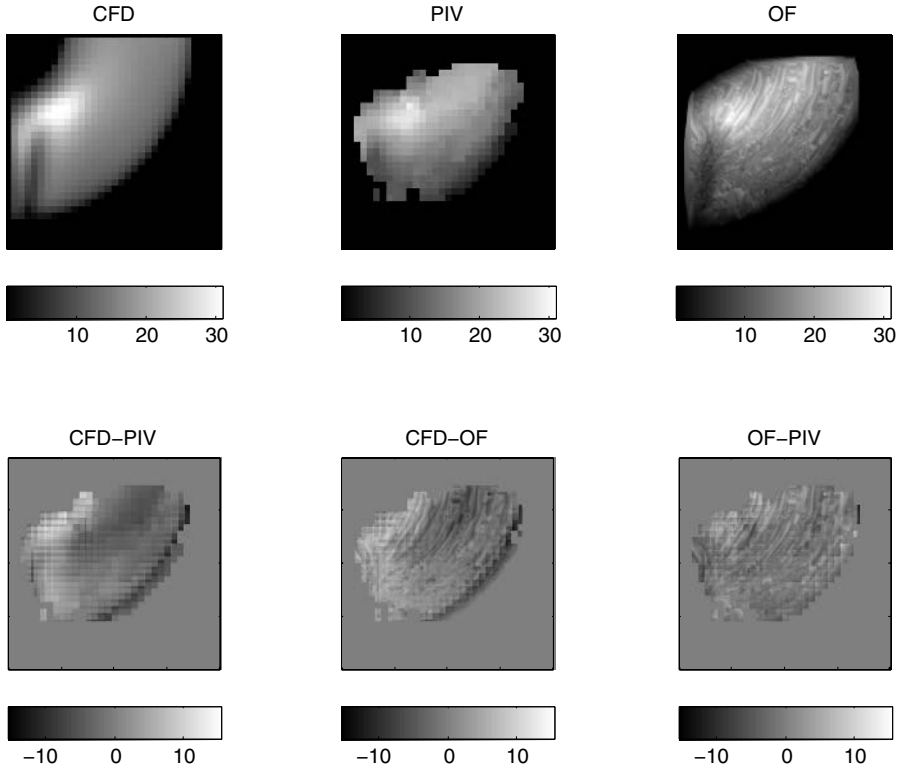
Though not having a uniform flow, we applied (6) since it is evident that the components of the velocity gradient tensor containing the derivatives w. r. t.  $z$  are large compared to the components containing the derivatives w. r. t.  $x$  and  $y$ .

The components  $m_{13}$  and  $m_{23}$  of the wall shear rate are mapped in Fig. 3, center. Since the examined flow is stationary, the shear rate is averaged over 300 frames. The gaps in the otherwise dense flow field indicate the spots, where the confidence measure, which is provided by the structure-tensor-technique, was too low for providing a reliable result.

Figure 4 displays the magnitude of the wall shear rate, obtained with different techniques, and compares the methods to each other. To establish some kind of “ground truth” [8] computed the flow numerically with the solver FLUENT6, which is shown in Fig. 4, top, left. The analysis of the flow using the PIV-technique and subsequent derivation of the wall shear rate, as carried out by [8] is shown in Fig. 4, top, center. Our result is mapped in Fig. 4, top, right. In order to compare the techniques with each other, we filled the gaps by means of interpolation, and afterwards smoothed the result using a 2D-anisotropic diffusion.

Our optical-flow-based method provides a dense, highly resolved vector field of the wall shear rate, which is capable of estimating this value at positions where the PIV-method fails. The flow-detachment in the lower left corner can, for instance, be reproduced very well. Both techniques, optical flow and PIV, show a deficit of the wall shear rate in the upper left corner, and a surplus in the upper right corner, compared to the analysis provided by computational fluid dynamics. These systematical deviations may occur as a result of the fact, that the particles, which are of a comparable large size, cannot follow the fluid ideally, or influence the fluid. To provide a measure of how much the results of the experimental methods are apart from the numerical solution, we added up the magnitudes of the differences on each pixel.

Optical flow resulted in about 10% better results, than PIV, when the CFD-solution was regarded as the “ground truth”. Besides that, the optical flow analysis yielded a much better spatial resolution, and the area, where a reliable estimate of the wall shear rate is possible, is about 30% greater compared to the area, obtained using the PIV analysis.



**Fig. 4.** Top: Wall shear rates determined by computational fluid dynamics (left), PIV-technique (center) and our optical-flow-based method (right). Bottom: Pointwise Differences between CFD and PIV (left), CFD and optical flow (center), optical flow and PIV (right).

## 5 Conclusion

A novel technique is presented for the direct estimation of the wall shear stress from particle-based image sequences. We propose an extension of the BCCE so that estimation can be done of the particle's velocity perpendicular to the image plane, using an exponential brightness change model, and also so that a direct analysis of the components of the strain rate tensor such as the wall shear rate is possible. Both synthetic and real experiments demonstrate the feasibility of the technique in good agreement with the ground truth. Though in this paper we addressed stationary wall-parallel flows only, our method may be extended to instationary, full 3D-flows in principle. In order to solve these challenges we are currently investigating a convection-driven free-surface flow. Furthermore we will solve the problem caused by the restriction of using spheres of exactly the same size, so that smaller and less expensive particles may be used, by means of illuminating with light consisting of two wavelengths [11].

*Acknowledgements.* We gratefully acknowledge the support by the priority program 1147 of the German Research Foundation.

## References

- [1] Horn, B.K.P., Schunk, B.G.: Determining optical flow. *Artificial Intelligence* **17** (1981) 185–204
- [2] Jehle, M., Klar, M., Jähne, B.: Optical-flow based velocity analysis. In Tropea, C., Foss, J., Yarin, A., eds.: *Springer Handbook of Experimental Fluid Dynamics*. (in preparation)
- [3] Corpetti, T., Memin, E., Perez, P.: Dense estimation of fluid flows. *IEEE Trans. on Pattern Analysis and Machine Intelligence* **24** (2002) 365–380
- [4] Garbe, C., Spies, H., Jähne, B.: Estimation of surface flow and net heat flux from infrared image sequences. *Journal of Mathematical Imaging and Vision* **19** (2003) 159–174
- [5] Ruhnau, P., Kohlberger, T., Schnörr, C., Nobach, H.: Variational optical flow estimation for particle image velocimetry. *Experiments in Fluids* **38** (2005) 21–32
- [6] Adrian, R.J.: Particle-imaging techniques for experimental fluid mechanics. *Annu. Rev. Fluid Mech.* **23** (1991) 261–304
- [7] Jähne, B., Haussecker, H., Geissler, P., eds.: *Handbook of Computer Vision and Applications*. Academic Press, San Diego, CA, USA (1999)
- [8] Debaene, P.: Neuartige Messmethode zur zeitlichen und örtlichen Erfassung der wandnahen Strömung in der Biofluidmechanik. Phd thesis, TU Berlin (2005)
- [9] Haussecker, H.W., Fleet, D.J.: Computing optical flow with physical models of brightness variation. *PAMI* **23** (2001) 661–673
- [10] Fleet, D.J.: *Measurement of Image Velocity*. Kluwer Academic Publishers, Dordrecht, Netherlands (1992)
- [11] Jehle, M., Jähne, B.: A novel method for spatiotemporal analysis of flows within the water-side viscous boundary layer. In: *12th International Symposium of Flow Visualisation*, Göttingen, Germany (2006)

# On-Line Variational Estimation of Dynamical Fluid Flows with Physics-Based Spatio-temporal Regularization

Paul Ruhnau, Annette Stahl, and Christoph Schnörr

Computer Vision, Graphics, and Pattern Recognition Group  
Department of Mathematics and Computer Science;  
University of Mannheim, D-68131 Mannheim, Germany  
{ruhnau, astahl, schnoerr}@uni-mannheim.de  
<http://www.cvgpr.uni-mannheim.de>

**Abstract.** We present a variational approach to motion estimation of instationary fluid flows. Our approach extends prior work along two directions: (i) The full incompressible Navier-Stokes equation is employed in order to obtain a physically consistent regularization which does not suppress turbulent flow variations. (ii) Regularization along the time-axis is employed as well, but formulated in a receding horizon manner contrary to previous approaches to spatio-temporal regularization. This allows for a recursive on-line (non-batch) implementation of our estimation framework.

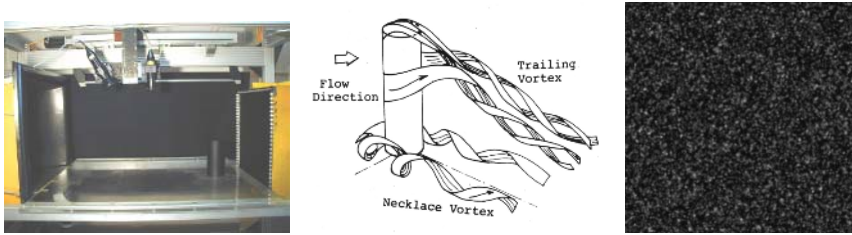
Ground-truth evaluations for simulated turbulent flows demonstrate that due to imposing both physical consistency and temporal coherency, the accuracy of flow estimation compares favourably even with optical flow approaches based on higher-order div-curl regularization.

## 1 Introduction

Image sequence analysis of fluid flows constitutes an active research field with a high industrial impact. Corresponding real-world measurements in concrete scenarios complement numerical results from direct simulations of the Navier-Stokes equation, particularly in the case of turbulent flows, and for the understanding of the complex spatio-temporal evolution of instationary flow phenomena. More and more advanced imaging devices (lasers, high-speed cameras, control logic, etc.) are currently developed that allow to record fully time-resolved image sequences of fluid flows at high resolutions. As a consequence, there is a need for advanced algorithms for the analysis of such data, to provide the basis for a subsequent pattern analysis, and with abundant applications across various areas.

The image measurement process proceeds as follows: First, the flow medium is seeded with small particles that are designed such that they accurately follow the fluid's motion. Next, entire velocity fields are measured by taking two or more images of the flow within short time intervals, and by estimating and interpolating the displacements of individual particles from frame to frame. This experimental method is known as *Particle Image Velocimetry (PIV)* [12]. Figure 1 shows a typical experimental setup in

a wind tunnel. To avoid blurred images when the flow is fast, laser pulses are used. As these are only 6-10 ns long, they are capable of freezing any motion. Note that at present the described experimental setup is only capable of yielding 2D velocity fields. Therefore, we have to confine ourselves to 2D image analysis, for the time being.



**Fig. 1. Left:** Experimental setting to study the flow around a cylinder. This setting results in an unsteady three-dimensional flow which can only be investigated using advanced imaging measuring techniques. **Middle:** Schematic illustration of typical flow phenomena [8]. **Right:** Typical PIV image.

A basic requirement for any motion estimation scheme is physical consistency. Otherwise, the information provided by a subsequent motion analysis is limited. Current approaches to PIV [12] do not address this issue *as part of* the motion estimation scheme. As a consequence, this calls for a novel combination of motion estimation and the Navier-Stokes equation which governs the real unknown flow in all applications. Our contribution in this paper is a variational approach to the estimation of motion fields constrained by the Navier-Stokes equation.

## 1.1 Related Work

Recently, variational optical flow techniques from the field of computer vision have been adopted and extended for the purpose of PIV [14,9,13,4,15]. Besides combining a carefully designed data term and coarse-to-fine estimation schemes with a standard first-order regularizer [14], a physically more plausible regularization has been suggested recently [15]. Because this approach is based on the Stokes-equation, however, it is based on related assumptions which are valid only for low Reynold numbers, i.e. *non-turbulent* flow. Another competitive research direction concerns the design and use of higher-order regularizers [9,4,19]. By separately penalizing the *gradient* of the divergence and the curl of flows, the major disadvantage of first-order regularization that penalize flow variations too much, are alleviated. Issues like well-posedness, accurated discretization and numerical stability, on the other hand, become more involved.

## 1.2 Contribution

We present a framework for fluid motion estimation that utilizes as prior knowledge that fact that flows have to satisfy the incompressible vorticity transport equation. This equation relates to the full (incompressible) Navier-Stokes equations and therefore is also valid in *turbulent* scenarios. Furthermore, rather than considering image pairs, our

estimation scheme takes into account the whole image sequence. As a result, it takes into account previous estimation results so as to enforce spatio-temporal coherency and regularization, however, *without* penalizing flow structures that are characteristic for instationary turbulent flows. Finally, analogously to the corresponding concept from control theory, our overall algorithm works in a receding horizon manner, that is flow velocities can be computed as soon as their respective frames have been recorded. In principle, this sets the stage for the real-time extraction of instationary flow phenomena from particle image sequences.

### 1.3 Organization

We present the vorticity transport equation, which embodies our prior knowledge we use for flow estimation, in section 2.1. Section 2.2 motivates and describes our variational approach and details the resulting constrained optimization problem. Corresponding numerical issues are dealt with in section 3. Numerical experiments for evaluating the approach are presented in section 4. We conclude in section 5.

## 2 Approach

### 2.1 The Vorticity Transport Equation

Let  $u = (u_1, u_2)^\top$ ,  $u = u(x, t)$ ,  $x = (x_1(t), x_2(t))^\top$ , denote a two-dimensional velocity field.

The incompressible *vorticity transport equation* is a special form of the Navier-Stokes equation for homogeneous flow and can be expressed as follows

$$\frac{D\omega}{Dt} = \frac{\partial}{\partial t}\omega + u \cdot \nabla\omega = \nu\Delta\omega, \quad \omega(x, 0) = \omega_0. \quad (1)$$

It describes the evolution of the fluid's vorticity over time. Note that in the absence of external forces acting on the fluid, this equation describes the flow completely.

### 2.2 Variational Model

Let  $I(x_1, x_2, t)$  denote the gray value of an image sequence recorded at location  $x = (x_1, x_2)^\top$  within some rectangular image domain  $\Omega$  and time  $t \in [0, T]$ . We adopt the basic assumption underlying most approaches to motion estimation that  $I$  is conserved. Thus, the total (material) derivative of  $I$  vanishes:

$$\frac{DI}{Dt} = u \cdot \nabla I + I_t = 0. \quad (2)$$

The spatial and temporal derivatives of  $I$  of the optical flow constraint (2) are estimated locally by using FIR filters. As the focus of this paper is on physically consistent regularization and not on design of the data term, we refer the interested reader to [14] for a detailed description.

As is well known, eqn. (2) alone cannot be used to reconstruct the velocity field  $u$ , because *any* vector field with components  $u \cdot \nabla I = -I_t$  at each location  $x$  satisfies (2).



The standard approach is to minimize the squared residual of (2) over the entire image domain  $\Omega$  and to add a variational term that either enforces smoothness of the flow (*first-order* regularization) [17]<sup>1</sup>

$$\int_{\Omega} \left\{ (u \cdot \nabla I + I_t)^2 + \alpha |\nabla \cdot u|^2 + \beta |\nabla \times u|^2 \right\} dx, \tag{3}$$

or smoothness of the divergence and vorticity (*second-order* regularization) [18]

$$\int_{\Omega} \left\{ (u \cdot \nabla I + I_t)^2 + \alpha |\nabla(\nabla \cdot u)|^2 + \beta |\nabla(\nabla \times u)|^2 \right\} dx. \tag{4}$$

We emphasize that both approaches (3) and (4) take only into account *spatial* context and determine a vector field for a *fixed* point in time  $t \in [0, T]$ .

Therefore, following the ideas of [16], our present work is an attempt to elaborate a *dynamic* representation of fluid flow. To this end, we solve eqn. (1) for the time interval  $[0, T]$  between a subsequent pair of image frames, where  $\omega_0$  denotes our current vorticity estimate. As a result, we obtain a *transported* vorticity field  $\omega_T := \omega(x, T)$ , which can be regarded as a *predicted* vorticity based on the assumption that our fluid is governed by the Navier-Stokes equation. The regularization term that we employ penalizes derivations from the predicted vorticity values and forces incompressibility:

$$\begin{aligned} \frac{1}{2} \int_{\Omega} \left\{ (u \cdot \nabla I + I_t)^2 + \lambda (\nabla \times u - \omega_T)^2 \right\} dx, \\ \text{s.t. } \nabla \cdot u = 0. \end{aligned} \tag{5}$$

We apply Neumann boundary conditions (i.e.  $\partial u / \partial n = 0$  on  $\partial \Omega$ ). Note that, while the regularization term of (5) penalizes deviations between the current vorticity estimate  $\omega$  and the propagated vorticity estimate of the preceding frame  $\omega_T$ , it does *not* enforce smoothness of the current vorticity. In practice, an implementation of (5) therefore leads to increasingly noisy vorticity estimates. Increasing the parameter  $\nu$  reduces the problem only slightly:  $\omega_T$  becomes smoother, but smoothness of  $\omega$  is still not enforced directly.

To overcome this problem, we add a term that mimics the small viscous term (Laplacian) on the right-hand side of eqn. (1). Expressing the new second-order regularization term equivalently through a first-order regularizer and an additional linear constraint, we finally obtain:

$$\begin{aligned} E = \frac{1}{2} \int_{\Omega} \left\{ (u \cdot \nabla I + I_t)^2 + \lambda (\omega - \omega_T)^2 + \kappa |\nabla \omega|^2 \right\} dx, \\ \text{s.t. } \nabla \cdot u = 0, \\ \nabla \times u = \omega. \end{aligned} \tag{6}$$

As we usually do not have a vorticity estimate at the very first frame of an image sequence, the overall estimation process is initialized with a vorticity estimate  $\omega_0 = 0$ .

---

<sup>1</sup> It can be shown easily that the Horn&Schunck approach [7] is just the special case of this regularization where  $\alpha = \beta$ .

The novel vorticity transport regularizer in (6), in connection with (1), can be perceived as a *special second-order div-curl regularizer*: Estimated flows from a given image sequence have vanishing divergence and a curl field (vorticity) that should be smooth and as close as possible to the transported vorticity.

### 3 Discretization and Optimization

#### 3.1 Discretisation of the Vorticity Transport Equation

We solve the time-dependent vorticity transport equation (1) with a second-order conservative finite difference algorithm. The method is upwind and two-dimensional in that the numerical fluxes are obtained by solving the characteristic form at cell edges (i.e. edges between adjacent pixels), and all fluxes are evaluated and differenced at the same time. The finite difference method that we employ is the Fromm-Van-Leer scheme [11].

The basic idea is to satisfy Godunov's theorem in a "natural" way. Roughly speaking, Godunov's theorem says that all methods of accuracy greater than order one will produce spurious oscillations in the vicinity of large gradients, while being second-order accurate in regions where the solution is smooth. Accordingly, Fromm-Van-Leer's scheme detects discontinuities and adapts its behavior such that the high-order accuracy of Fromm's scheme is preserved for smooth parts of the solution, while spurious oscillations are avoided through first-order accuracy at detected discontinuities.

#### 3.2 Variational Approach

For every image pair (two consecutive frames of the image sequence), we have to solve optimization problem (6) which comprises a convex functional and two linear constraint equations. We transform this constrained optimization problem into a saddle point problem. Accordingly, the unique vector field  $u(x)$  minimizing (6), along with the vorticity  $\omega$  and multipliers  $p, q$ , are determined by the variational system

$$\begin{aligned} a((u, \omega)^\top, (\tilde{u}, \tilde{\omega})^\top) + b((p, q)^\top, (\tilde{u}, \tilde{\omega})^\top) &= ((f, g)^\top, (\tilde{u}, \tilde{\omega})^\top), \quad \forall \tilde{u}, \tilde{\omega} \\ b((\tilde{p}, \tilde{q})^\top, (u, \omega)^\top) &= 0, \quad \forall \tilde{p}, \tilde{q}. \end{aligned} \quad (7)$$

The bilinear and linear forms read:

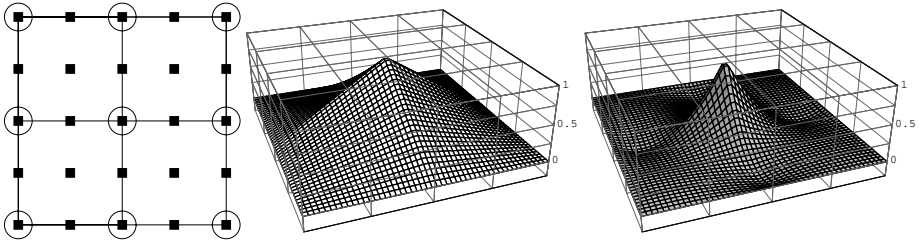
$$a((u, \omega)^\top, (\tilde{u}, \tilde{\omega})^\top) := \int_{\Omega} \left\{ u \cdot \nabla I \nabla I \cdot \tilde{u} + \lambda \omega \tilde{\omega} + \kappa \nabla \omega \cdot \nabla \tilde{\omega} \right\} dx, \quad (8)$$

$$b((p, q)^\top, (\tilde{u}, \tilde{\omega})^\top) := - \int_{\Omega} \left\{ p \nabla \cdot \tilde{u} + q (\nabla \times \tilde{u} - \tilde{\omega}) \right\} dx. \quad (9)$$

The right-hand side reads:

$$((f, g)^\top, (\tilde{u}, \tilde{\omega})^\top) := \int_{\Omega} \left\{ -I_t \nabla I \cdot \tilde{u} + \lambda \omega_T \tilde{\omega} \right\} dx. \quad (10)$$

We choose a regular tessellation of the image domain  $\Omega$  and discretize (7) using finite elements. It is well-known from computational fluid dynamics (cf. Stokes equation) that standard first-order finite element discretizations of saddle point problems may result



**Fig. 2.** **Left:** Sketch of 2D Taylor-Hood elements: biquadratic velocity elements (squares) and bilinear pressure elements (circles). **Middle:** Basis function  $\phi$  of a bilinear finite element. **Right:** Basis function  $\psi$  of a biquadratic finite element.

in instabilities or even in so-called locking effects, where the zero velocity field is the only one satisfying the incompressibility condition.

Therefore, when solving saddle point problems, mixed finite elements are traditionally used [2]. An admissible choice is the so-called Taylor-Hood element based on a square reference element with nine nodes (fig. 2). Each component of the velocity field is defined in terms of piecewise quadratic basis functions  $\psi_i$  located at each node (the solid squares in fig. 2), whereas the Lagrange multipliers  $p$  and  $q$  and the vorticity  $\omega$  are represented by linear basis functions  $\phi_i$  attached to each corner node (indicated by circles in fig. 2). It can be shown that Taylor-Hood elements fulfill the so-called Babuska-Brezzi condition [2], making the discretized problem well-posed.

Indexing the velocity nodes (squares in fig. 2) by  $1, 2, \dots, N$ , we obtain

$$u_1(x) = \sum_{i=1}^N u_i \psi_i(x) \tag{11}$$

and similarly for  $u_2(x)$  (where  $u = (u_1, u_2)^\top$ ) and the components of  $\tilde{u}$ .

By analogy, we obtain for the  $M$  Lagrange multiplier nodes (circles in fig. 2)

$$p(x) = \sum_{i=1}^M p_i \phi_i(x) \tag{12}$$

and similarly expressions for  $q, \omega, \tilde{p}, \tilde{q}, \tilde{\omega}$ . Hence, each function  $u, \tilde{u}$  is represented by  $2N$  real variables, and each function  $p, q, \omega, \tilde{p}, \tilde{q}, \tilde{\omega}$  is represented by  $M$  real variables. For the sake of simplicity, we will use the same symbols to denote the corresponding vectors. The discretized system (7) then reads

$$\begin{aligned} A(u, \omega)^\top \cdot (\tilde{u}, \tilde{\omega})^\top + B^\top(p, q)^\top \cdot (\tilde{u}, \tilde{\omega})^\top &= (f, g)^\top \cdot (\tilde{u}, \tilde{\omega})^\top, \quad \forall \tilde{u}, \tilde{\omega} \\ B(u, \omega)^\top \cdot (\tilde{p}, \tilde{q})^\top &= 0, \quad \forall \tilde{p}, \tilde{q}. \end{aligned} \tag{13}$$

These equations have to be satisfied for arbitrary  $\tilde{u}, \tilde{p}, \tilde{q}, \tilde{\omega}$ , thus we obtain:

$$A \begin{pmatrix} u \\ \omega \end{pmatrix} + B^\top \begin{pmatrix} p \\ q \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix}, \quad B \begin{pmatrix} u \\ \omega \end{pmatrix} = 0. \tag{14}$$

In order to numerically solve the saddle point problem (14), we want to employ the Uzawa algorithm (cf., e.g. [1]). However, this requires  $A$  to be positive definite which is not the case here, because the relations  $u$  and  $\omega$  defining  $A$  in (8) are mutually independent and  $u$  is only involved through a degenerate quadratic form. This problem can be removed by (a) including a penalty term related to the divergence constraint into our Lagrange multiplier formulation to obtain an Augmented Lagrangian formulation [5], and by (b) splitting the vorticity matching term into two equivalent terms, one containing  $\nabla \times u$  and the other one containing  $\omega$ . This yields the following modification of the bilinear form (8):

$$a_p((u, \omega)^\top, (\tilde{u}, \tilde{\omega})^\top) := \int_{\Omega} \left\{ u \cdot \nabla I \nabla I \cdot \tilde{u} + \frac{\lambda}{2} (\omega \tilde{\omega} + (\nabla \times u)(\nabla \times \tilde{u})) + \mu (\nabla \cdot u)(\nabla \cdot \tilde{u}) + \kappa \nabla \omega \cdot \nabla \tilde{\omega} \right\} dx. \quad (15)$$

We point out that this modification is done for numerical reasons only. It does not change the optimization problem (6). Matrix  $A_p$  resulting from the discretization of (15) is positive definite and, because  $u$  and  $\omega$  do *not explicitly* depend on each other, can be split into two systems:

- The system containing  $u$  is the linear system with a simple first-order div-curl regularization (cf., e.g. [17], and (3)).
- The system containing  $\omega$  corresponds to a simple first-order quadratic functional.

Because  $A_p$  is invertible and well-conditioned, we solve the first equation of the system (14), with  $A$  replaced by  $A_p$ , for the unknown  $u$

$$\begin{pmatrix} u \\ \omega \end{pmatrix} = A_p^{-1} \left[ \begin{pmatrix} f \\ g \end{pmatrix} - B^\top \begin{pmatrix} p \\ q \end{pmatrix} \right], \quad (16)$$

and insert the result into the second equation:

$$BA_p^{-1} \left[ \begin{pmatrix} f \\ g \end{pmatrix} - B^\top \begin{pmatrix} p \\ q \end{pmatrix} \right] = 0. \quad (17)$$

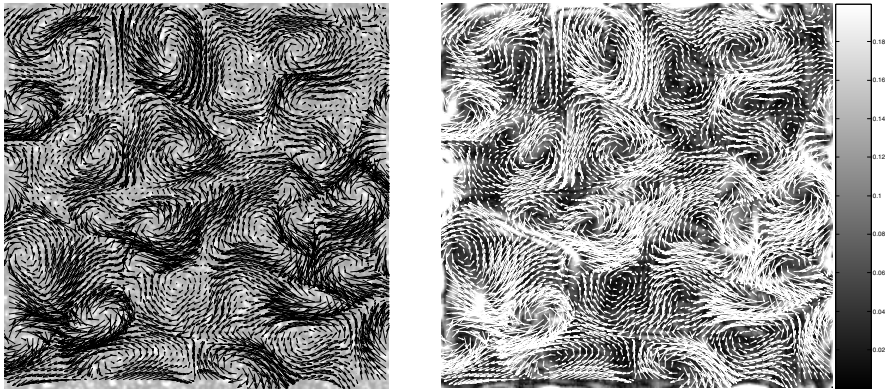
This problem only involves the adjoint variables  $p, q$ :

$$(BA_p^{-1}B^\top) \begin{pmatrix} p \\ q \end{pmatrix} = BA_p^{-1} \begin{pmatrix} f \\ g \end{pmatrix}. \quad (18)$$

The matrix  $(BA_p^{-1}B^\top)$  is symmetric and positive definite. Therefore, we apply the conjugate gradient iteration to (18). This requires a single matrix inversion in every iteration step. This is efficiently accomplished using multi grid iteration (cf. [6]).

## 4 Experimental Evaluation

This section shows numerical results on ground truth fluid image sequences obtained with our approach in comparison with first-order regularization and with second-order div-curl regularization.



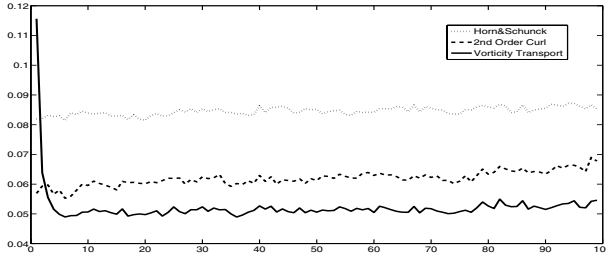
**Fig. 3. Left:** 100th frame of the synthetic image sequence with ground truth velocity field. **Right:** Estimated velocity field for the 100th frame. The background intensity shows the absolute RMS error (brighter = larger error), which is about 0.055 px. on average (cf. fig. 4).

The evaluation of our approach from the viewpoint of fluid mechanics (real data, without ground-truth) is beyond the scope of this paper.

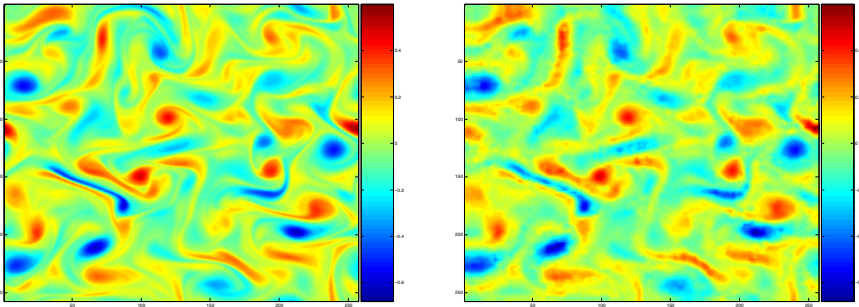
The synthetic PIV image sequence that we used for testing was provided by [3]. The underlying velocity field was computed by a so-called pseudo-spectral code that solves the vorticity transport equation in Fourier space and evaluates a subgrid model for simulating small-scale turbulent effects on the larger scales of the flow. These latter effects, of course, are *not* known in practice, nor was anything related to that used while evaluating our approach.

In order to simulate the intensity function of real PIV images, the computed velocity fields are used to transport collections of (images of) particles that are typically used for the seeding of flows so as to make them visible (cf. section 1). The scheme resembles the one described in [10]. We used the first 100 frames of the synthesized PIV image sequence and compared the following three approaches:

- *Horn&Schunck* [7]: First-order regularization, temporal coherency is not exploited, no incompressibility constraint is imposed. The smoothness parameter  $\lambda = 0.005$  was manually selected for best performance.
- *2nd Order Regularization* [19]: These authors used higher-order regularization with an additional incompressibility constraint. Instead of mixed finite elements (as we do), the authors used the so-called mimetic finite differencing scheme. Temporal coherency is not exploited. Parameters:  $\lambda_1 = 0.5$ ,  $\lambda_2 = 0.05$ , manually selected for best performance.
- *Vorticity Transport Approach (this paper)*: As described above, higher-order regularization is used, the incompressibility constraint is imposed, and temporal coherency is exploited in an on-line manner. Parameters  $\lambda = 0.005$ ,  $\mu = 0.005$ ,  $\nu = 0.1$ ,  $\kappa = 0.0005$ . As for the other approaches, we selected the regularization parameters  $\lambda, \mu, \kappa$  by hand. Note that the viscosity coefficient  $\nu$  is not a free user parameter but characterizes the physical nature of the fluid flow.



**Fig. 4.** Average absolute RMS error (in pixels) for frames 1-100, using three different methods. First-order regularization performs worse than second-order regularization. Both error curves are constant because temporal coherency is not exploited. The approach based on vorticity transport starts with a rather low accuracy (assumption of  $\omega = 0$ , which is not valid) but then becomes significantly more accurate than the two other techniques due to the physically consistent regularization over time. This novel spatio-temporal regularization is achieved with an on-line computational scheme and fixed storage requirements, irrespective of the length of the image sequence. The decay of the error curve within the first 10 frames clearly displays the usage of this implicitly encoded “memory”.



**Fig. 5. Left:** True vorticity of frame 100. **Lower Right:** Estimated vorticity  $\omega$  for frame 100. For the first frame, the estimation process was initialized with  $\omega = 0$ , corresponding to “nothing is known in advance”. The result on the right shows that not only the vorticity transport equation has been successfully adapted to the observed image sequence, but that it improves the accuracy of flow estimation in terms of  $u$ , too (cf. fig. 4). As a consequence, flow *derivatives* can be estimated fairly accurate, as shown in the right panel. Such quantitative information is very important in connection with imaging-based experimental fluid mechanics.

Figure 4 compares the errors of all three approaches over time. The first-order regularization approach yields the highest errors, while the second-order approach is much more accurate. The errors of both approaches stay constant over time because each subsequent image pair is independently evaluated and temporal coherency is ignored.

For the first frame, the approach presented in this paper, utilizing the vorticity transport equation, shows worse performance than the other two algorithms. During the subsequent period of time, however, the error of the vorticity transport approach de-

creases considerably, because not only higher-order regularization is used but temporal coherency is successfully exploited as well.

We emphasize that temporal coherency does *not* mean smoothness. Rather, the flow exhibits high spatio-temporal gradients as turbulent fluids do. Temporal coherency relates to a physically consistent transport mechanism interacting with flow estimation from an image sequence. Due to the on-line computational scheme, fixed computational resources are needed no matter how long the image sequence is. The decay of the error curve over several frames in figure 4 shows, however, that the approach is able to memorize the history longer than just the previous frame.

Figure 3 displays the estimated velocity for the for the 100th frame, along with the respective RMS errors. The reconstructed velocity field is surprisingly exact, in view of the highly non-rigid motion we are dealing with. Figure 5 shows that even the vorticity related to flow *derivatives* is reconstructed quite well under these difficult conditions. We expect such quantitative data to be valuable information in connection with imaging-based fluid mechanics.

## 5 Conclusion

We presented an approach to fluid motion estimation that uses the vorticity transport equation for physically consistent spatio-temporal regularization. The approach combines variational motion estimation with higher-order regularization and motion prediction through a transport process. For motions that conform to our assumption (i.e. fluids that are governed by the incompressible 2D Navier-Stokes equation), a temporal regularization effect, computed in a recursive manner, was demonstrated. In these scenarios, our approach outperforms advanced variational approaches for optical flow estimation.

## Acknowledgment

Support by the Deutsche Forschungsgemeinschaft (DFG, SCHN 457/6) within the priority programme “Bildgebende Messverfahren in der Strömungsmechanik” ([www.spp1147.tu-berlin.de](http://www.spp1147.tu-berlin.de)) and by the EU-project “Fluid Image Analysis and Description” (<http://fluid.irisa.fr/>) is gratefully acknowledged.

## References

1. D. Braess. *Finite elements. Theory, fast solver & appl. in solid mechanics*. Springer, 1997.
2. F. Brezzi and M. Fortin. *Mixed and hybrid finite element methods*. Springer-Verlag New York, Inc., New York, NY, USA, 1991.
3. J. Carlier and D. Heitz. 2D turbulence sequence provided by Cemagref within the European Project ‘Fluid Image Analysis and Description’.
4. Th. Corpetti, D. Heitz, G. Arroyo, E. Mémin, and A. Santa-Cruz. Fluid experimental flow estimation based on an optical-flow scheme. *Exp. Fluids*, 40(1):80–97, 2005.
5. M. Fortin and R. Glowinski. *Augmented Lagrangian Methods: Applications to the Numerical Solution of Boundary-value Problems*. North-Holland, Amsterdam, 1983.

6. W. Hackbusch. *Iterative Solution of Large Sparse Systems of Equations*, volume 95 of *AMS*. Springer, 1993.
7. B. Horn and B. Schunck. Determining optical flow. *Art. Intelligence*, 17:185–203, 1981.
8. T. Kawamura, M. Hiwada, T. Hibino, I. Mabuchi, and M. Kumada. Flow around a finite circular cylinder on a flat plate. *Bulletin of the JSME*, 27(232):2142–2151, 1984.
9. T. Kohlberger, E. Mémin, and C. Schnörr. Variational dense motion estimation using the helmholtz decomposition. In L.D. Griffin and M. Lillholm, editors, *Scale Space Methods in Computer Vision*, volume 2695 of *LNCS*, pages 432–448. Springer, 2003.
10. K. Okamoto, S. Nishio, and T. Kobayashi. Standard images for particle-image velocimetry. *Meas. Sci. Technol.*, 11:685–691, 2000.
11. E. G. Puckett and P. Colella. *Finite Difference Methods for Computational Fluid Dynamics (Cambridge Texts in Applied Mathematics)*. Cambridge University Press, 2005.
12. M. Raffel, C. Willert, and J. Kompenhans. *Particle Image Velocimetry*. Springer, 2001.
13. P. Ruhnau, C. Gütter, and C. Schnörr. A variational approach for particle tracking velocimetry. *Meas. Sci. Technol.*, 16(7):1449–1458, 2005.
14. P. Ruhnau, T. Kohlberger, H. Nobach, and C. Schnörr. Variational optical flow estimation for particle image velocimetry. *Exp. Fluids*, 38:21–32, 2005.
15. P. Ruhnau and C. Schnörr. Optical stokes flow: An imaging based control approach. *Exp. Fluids*, 2006. submitted.
16. A. Stahl, P. Ruhnau, and C. Schnörr. A distributed-parameter approach to dynamic image motion. In *Int. Workshop on The Repres. and Use of Prior Knowl. in Vision, LNCS, Springer, ECCV 2006*. in press.
17. D. Suter. Mixed finite elements and whitney forms in visual reconstruction. In *Geometric Methods in Computer Vision II*, pages 51–62, 1993.
18. D. Suter. Mixed-finite element based motion est. *Innov. Tech. Biol. Med.*, 15(3), 1994.
19. J. Yuan, P. Ruhnau, E. Mémin, and C. Schnörr. Discrete orthogonal decomposition and variational fluid flow estimation. In *Scale-Space 2005*, volume 3459 of *Lect. Not. Comp. Sci.*, pages 267–278. Springer, 2005.



# Near Real-Time Motion Segmentation Using Graph Cuts

Thomas Schoenemann and Daniel Cremers

CVPR Group, University of Bonn  
Römerstr. 164, 53117 Bonn, Germany  
{schoenemann, dcremers}@iai.uni-bonn.de

**Abstract.** We present a new approach to integrated motion estimation and segmentation by combining methods from discrete and continuous optimization. The velocity of each of a set of regions is modeled as a Gaussian-distributed random variable and motion models and segmentation are obtained by alternated maximization of a Bayesian a-posteriori probability. We show that for fixed segmentation the model parameters are given by a closed-form solution. Given the velocities, the segmentation is in turn determined using graph cuts which allows a globally optimal solution in the case of two regions. Consequently, there is no contour evolution based on differential increments as for example in level set methods. Experimental results on synthetic and real data show that good segmentations are obtained at speeds close to real-time.

## 1 Introduction

Since the seminal works of Lucas and Kanade [17] and Horn and Schunck [13], motion estimation has become one of the major problems addressed in Computer Vision. Motion estimation techniques can be employed in numerous Computer Vision tasks such as the study of dynamical processes [14].

A closely related problem is motion *segmentation*, namely the grouping of image regions which are similar in their motion. Early approaches worked by first estimating the flow field, then segmenting it (cf. [19]). More recently, particularly since the work of [18] and [9], approaches to address the problems of motion estimation and segmentation by minimizing a single energy functional have become popular. Minimization is done by alternately updating the flow field and the segmentation boundary.

Most present approaches handle the case of piecewise affine motion [9,2,15] or allow non-parametric variation of the flow field [8,1]. In this work we present models with both piecewise constant and piecewise affine velocities.

Current approaches to motion segmentation are typically based on pde evolution such as the Level Set Method [9,8,1]. While there exist motion segmentation methods using graph cuts [2,15], they are based on non-linear flow-errors, resulting in run-times far from real-time. We use linearized flow errors, allowing accurate segmentations at speeds close to real-time at the costs of less accurate

motion estimates. Given the velocity in each region, the globally optimal bilayer segmentation can be obtained in effectively linear time.

This paper is organized as follows: In Section 2 we derive an energy functional for motion segmentation with piecewise constant velocities by modeling the velocity of each region as a Gaussian-distributed random variable. Section 3 extends this framework to piecewise affine velocities. In Section 4 we propose an efficient optimization scheme, combining graph cuts and differential methods. Finally, we present in Section 5 experimental results on synthetic and real data which demonstrate that high-quality purely motion based segmentations can be obtained with run-times close to real time.

## 2 Statistical Formulation of Motion Segmentation

Given two frames  $I_1$  and  $I_2$  of a video sequence, we are interested in determining for each pixel  $\mathbf{p} = (x_p, y_p)$  in the first image where it is to be found in the second, i.e. with what velocity it moved. In this work we deal with motion segmentation and for the sake of efficiency fix the number of regions to 2. Notice that an extension to multiple regions is easily possible using the expansion moves of [7]. In this case, however, we can no longer guarantee globally optimal segmentations as the multilabel problem is NP-hard. For the moment, each region  $i$  is associated a constant velocity  $\mathbf{v}_i, i \in \{0, 1\}$ . The next section will extend this to more elaborate parametric motion models. The problem then is to assign each pixel  $\mathbf{p}$  a region  $l(\mathbf{p}) \in \{0, 1\}$  and determine the optimal velocity for each region. We denote by  $R_i$  the set of all pixels labeled  $i$ . We address this problem by the Bayesian method of minimizing the negative logarithm of the posterior probability

$$\begin{aligned} & \arg \min_{l, \bar{\mathbf{v}}} -\log(\text{pr}(l, \bar{\mathbf{v}}|I_1, I_2)) \\ &= \arg \min_{l, \bar{\mathbf{v}}} \left[ \underbrace{-\log(\text{pr}(l))}_{E_{smooth}(l)} \underbrace{-\log(\text{pr}(I_2|\bar{\mathbf{v}}, l, I_1))}_{E_{data}(l, \bar{\mathbf{v}})} - \log(\underbrace{\text{pr}(\bar{\mathbf{v}}, I_1|l)}_{uniform}) + \text{const} \right] \quad (1) \end{aligned}$$

where  $\bar{\mathbf{v}}$  contains all velocities. In this work, we assume the third probability to be uniform within a reasonable range and assume that the first only depends on the length of the boundary. Based on the Cauchy-Crofton formula from integral geometry this length can be approximated by [6]

$$E_{smooth}(l) = \frac{\nu}{2} \sum_{(\mathbf{p}, \mathbf{q}) \in \mathcal{N}} \frac{(1 - \delta(l(\mathbf{p}), l(\mathbf{q})))}{\|\mathbf{p} - \mathbf{q}\|} \quad (2)$$

with Kronecker- $\delta$ , the free parameter  $\nu$  and a neighborhood system  $\mathcal{N}$  we choose to be of size 8. Assuming the intensities of the moving objects to stay constant,  $\text{pr}(\bar{\mathbf{v}}, I_2|l, I_1)$  might be restricted to be non-zero only if

$$0 = I_2(\mathbf{p} + \mathbf{v}_{l(\mathbf{p})}) - I_1(\mathbf{p}) \approx \nabla I(\mathbf{p})^T \mathbf{v}_{l(\mathbf{p})} + I_t(\mathbf{p}) \quad \forall \mathbf{p} \quad (3)$$

The last quantity is known as *linearized flow-error*. However, there are several reasons for this assumption not to be true: For one, there is the camera noise. Then there are changes in lighting and reflections. Lastly, we assume that each region has a constant velocity, but desire to segment objects with partially different depths (think of a mirror on a car, or simply a sphere) as one. However, parts with different depths will move with different velocities in the image plane.

Nevertheless, we desire the constraint in (3) to be fulfilled as good as possible and assume the probability  $pr(I_2|\bar{\mathbf{v}}, l, I_1)$  to be only dependent on its deviation from 0 for all pixels.

It can be shown that the likelihood proposed by [17] is equivalent to the assumption of Gaussian noise on the image data, where the noise is independent of the region. This results in a Gaussian-distributed flow-error

$$E_{data}(l, \bar{\mathbf{v}}) = \frac{1}{2} \sum_{i=0}^1 \sum_{\mathbf{p} \in R_i} (\nabla \mathbf{I}(\mathbf{p})^T \mathbf{v}_i + I_t(\mathbf{p}))^2 \quad (4)$$

In this work, following [10] we assume that the velocity at each pixel  $\mathbf{p}$  of region  $i$  is a Gaussian-distributed random variable, that is  $\tilde{\mathbf{v}}_i(\mathbf{p}) = \mathbf{v}_i + \boldsymbol{\eta}(\mathbf{p})$  for  $\mathbf{p} \in R_i$ , where  $\boldsymbol{\eta}(\mathbf{p}) \sim N(\mathbf{0}, \sigma_i^2 \mathbf{I}_2)$  with identity matrix  $\mathbf{I}_2$ . We require that  $\tilde{\mathbf{v}}_i(\mathbf{p})$  exactly fulfill the constraint in (3), resulting in

$$\nabla \mathbf{I}(\mathbf{p})^T \mathbf{v}_i + I_t(\mathbf{p}) = \nabla \mathbf{I}(\mathbf{p})^T \boldsymbol{\eta}(\mathbf{p})$$

The flow-error for  $\mathbf{v}_i$  is now  $N(0, \sigma_i^2 \|\nabla \mathbf{I}(\mathbf{p})\|^2)$ -distributed. This leads to

$$E_{data}(l, \bar{\mathbf{v}}, \boldsymbol{\sigma}) = \frac{1}{2} \sum_{i=0}^1 \sum_{\mathbf{p} \in R_i} \left[ \log(2\pi \|\nabla \mathbf{I}(\mathbf{p})\|^2 \sigma_i^2) + \frac{(\nabla \mathbf{I}(\mathbf{p})^T \mathbf{v}_i + I_t(\mathbf{p}))^2}{\|\nabla \mathbf{I}(\mathbf{p})\|^2 \sigma_i^2} \right] \quad (5)$$

where  $\boldsymbol{\sigma}$  contains all variances. In contrast to [10] we allow a separate variance for each region. Consequently the normalization term cannot be neglected. Lacking suitable estimates for the variances, we optimize it in each region. To avoid numerical instabilities, we replace  $\|\nabla \mathbf{I}(\mathbf{p})\|^2$  by  $\max\{\|\nabla \mathbf{I}(\mathbf{p})\|^2, 1\}$  in the denominator.

### 3 Extension to Parametric Motion Models

In this section we extend the data term in (5) to piecewise affine motion. A pixel  $\mathbf{p} = (x_p, y_p)$  belonging to  $R_i$  is now no longer assigned the constant velocity  $\mathbf{v}_i$ , but an affine velocity  $\mathbf{S}(\mathbf{p})\boldsymbol{\vartheta}_i$  where

$$\mathbf{S}(\mathbf{p}) = \begin{pmatrix} x_p & y_p & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_p & y_p & 1 \end{pmatrix}$$

and  $\boldsymbol{\vartheta}_i$  is the vector containing all parameters of the affine model for region  $i$ :

$$\boldsymbol{\vartheta}_i = (a_i \ b_i \ c_i \ d_i \ e_i \ f_i)^T$$

We differ from the common notation  $\mathbf{A}\mathbf{p} + \mathbf{b}$  as this simplifies the equations for the update of parameters greatly. The task is now to minimize the energy functional (1) with data term

$$E_{data}(l, \bar{\mathbf{v}}, \boldsymbol{\sigma}) = \frac{1}{2} \sum_{i=0}^1 \sum_{\mathbf{p} \in R_i} \left[ \log(2\pi \|\nabla \mathbf{I}(\mathbf{p})\|^2 \sigma_i^2) + \frac{(\nabla \mathbf{I}(\mathbf{p})^T \mathbf{S}(\mathbf{p}) \boldsymbol{\vartheta}_i + I_t(\mathbf{p}))^2}{\|\nabla \mathbf{I}(\mathbf{p})\|^2 \sigma_i^2} \right] \quad (6)$$

with respect to the vectors  $\boldsymbol{\vartheta}_i$ , the variances  $\boldsymbol{\sigma}$  and the segmentation  $l$ .

## 4 An Efficient Semi-discrete Optimization Scheme

Minimization of the energy functional in (1) is done by alternately updating the segmentation, the velocities and the variances. To this end we propose a semi-discrete optimization scheme, combining fast discrete global optimization methods for the segmentation step with continuous optimization for the motion parameters. In the following we first state how the segmentation is updated using graph cuts. We then give closed form global solutions for the velocities and variances. Each quantity is set to the globally optimal one given the others. In all cases we show the update for the affine motion model.

### 4.1 Fast Global Segmentation Via Graph Cuts

Given the velocities of each region the segmentation step requires the minimization of a cost functional with binary-valued variables. To solve this problem, we revert to the graph cut method, which will be detailed in the following.

Greig et al. [12] were the first to show how to exploit the graph cut technique for problems of Computer Vision. They were concerned with the problem of binary image restoration. In [16] the minimization of submodular functions of binary variables with at most ternary terms is discussed<sup>1</sup>.

The complexity of the general problem is low-order polynomial, but using the fast algorithm of [5] for most Computer Vision problems (including ours) it is effectively linear. This algorithm makes use of the theorem of Ford and Fulkerson [11] by solving the related problem to compute the maximum flow in a graph.

To give the reader an intuition of how the method works, we explain it in the following. We state here the problem for undirected graphs as this suffices for our application. A graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  consists of a set of nodes  $\mathcal{V}$  and a set of edges  $\mathcal{E}$ . An edge  $e = \{p, q\}$  links two nodes  $p$  and  $q$ . For the problem of the minimum 0/1-cut there are two distinguished nodes 0 and 1, i.e.  $\mathcal{V} = \{0, 1\} \cup \mathcal{V}_0$ . In our case  $\mathcal{V}_0$  will correspond to the set of pixels. Furthermore, each edge  $\{p, q\}$  is assigned a weight  $w_{\{p,q\}}$ .

<sup>1</sup> Code is available at <http://www.adastral.ucl.ac.uk/~vladkolm/software.html>

A cut on  $\mathcal{G}$  is a labeling  $l$  of all nodes such that  $l(v) \in \{0', 1'\}$  for  $v \in \mathcal{V}$  and  $l(0) = 0'$  and  $l(1) = 1'$ . The costs  $|l|$  of a cut is the sum of the weights of all edges between a node labeled '0' and one labeled '1':

$$|l| = \sum_{\{p,q\} \in \mathcal{E}: l(p)=0', l(q)=1'} w_{\{p,q\}}$$

The min-cut problem is to find a cut  $l$  with minimal costs. As it is inherently related to finding a binary labeling, many binary optimization problems of Computer Vision can be reduced to it, including ours.

In the following we show how the graph looks like for motion segmentation with two regions 0 and 1. In our case  $\mathcal{V}_0$  is the set of all pixels. Each pixel  $p$  is linked to 0 by an edge  $\{0, p\}$  and to 1 by an edge  $\{p, 1\}$ . These links are called *t-links*. Additionally there are *n-links* connecting pixels  $p$  and  $q$  for any  $\{p, q\} \in \mathcal{N}$ . Their weight is set to  $w_{\{p,q\}} = \nu$ . Setting

$$w_{\{0,p\}} = \log(2\pi\sigma_1^2 \|\nabla \mathbf{I}(\mathbf{p})\|^2) + \frac{(\nabla \mathbf{I}(\mathbf{p})^T \mathbf{S}(\mathbf{p}) \boldsymbol{\vartheta}_1 + I_t(\mathbf{p}))^2}{\sigma_1^2 \|\nabla \mathbf{I}(\mathbf{p})\|^2}$$

$$w_{\{p,1\}} = \log(2\pi\sigma_0^2 \|\nabla \mathbf{I}(\mathbf{p})\|^2) + \frac{(\nabla \mathbf{I}(\mathbf{p})^T \mathbf{S}(\mathbf{p}) \boldsymbol{\vartheta}_0 + I_t(\mathbf{p}))^2}{\sigma_0^2 \|\nabla \mathbf{I}(\mathbf{p})\|^2}$$

the reader may verify that the costs of any cut  $l$  correspond to the costs of a segmentation  $l'$  where  $l'(p) = 0$  if  $l(p) = 0'$  and  $l'(p) = 1$  if  $l(p) = 1'$  (see figure 1 for an example on a one-dimensional image). Hence, using graph cuts the globally optimal segmentation can be computed in one step. Our efficient implementation uses flow-recycling [4] where previously computed flows are reused and the t-links are updated in each iteration.

### 4.2 Update of Continuous Parameters

As suggested in [17,10] minimization with respect to the motion parameters  $\mathbf{v}_i, \sigma_i$  can be done by setting the respective derivatives of (5) to zero. This leads to  $\mathbf{v}_i = \mathbf{M}^{-1} \mathbf{b}$  with

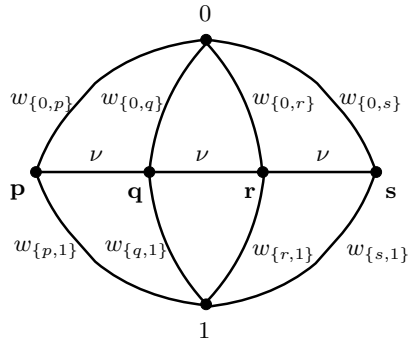
$$\mathbf{M} = \sum_{\mathbf{p} \in R_i} \frac{\mathbf{S}(\mathbf{p})^T \nabla \mathbf{I}(\mathbf{p}) \nabla \mathbf{I}(\mathbf{p})^T \mathbf{S}(\mathbf{p})}{\|\nabla \mathbf{I}(\mathbf{p})\|^2}$$

$$\mathbf{b} = - \sum_{\mathbf{p} \in R_i} \frac{\mathbf{S}(\mathbf{p})^T \nabla \mathbf{I}(\mathbf{p}) \cdot I_t(\mathbf{p})}{\|\nabla \mathbf{I}(\mathbf{p})\|^2}$$

The variance is given by

$$\sigma_i^2 = \frac{1}{|R_i|} \cdot \sum_{\mathbf{p} \in R_i} \frac{(\nabla \mathbf{I}(\mathbf{p})^T \mathbf{S}(\mathbf{p}) \boldsymbol{\vartheta}_i + I_t(\mathbf{p}))^2}{\|\nabla \mathbf{I}(\mathbf{p})\|^2}$$

So all quantities (segmentation, velocities and variances) are set to the globally optimal solution given the other quantities. Notice that convergence is guaranteed as the energy never increases and is always strictly positive.



**Fig. 1.** An example of a graph for the segmentation problem on a 1-D image

## 5 Experimental Results

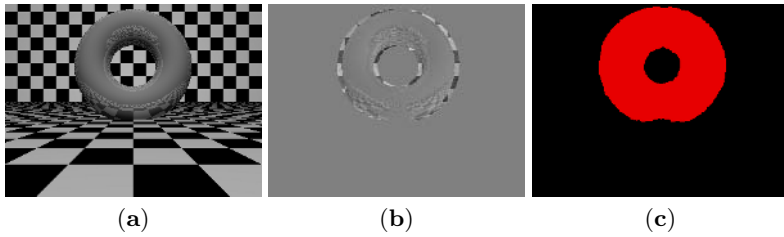
To show the abilities of the proposed model, we deal first with synthetic data, showing that surfaces with non-constant depth can be handled. Afterwards we present results on real-life data. All images were pre-smoothed. To initialize the segmentation process, we applied block-matching. Therefore the image was segmented into non-overlapping blocks, each block was tested against a set of integer-valued velocities and assigned the best one. Afterwards the two most frequent velocities were taken.

The only free parameter is the length penalty  $\nu$ . For all experiments we chose the  $\nu$  that gave the best performance. Notice that in contrast to existing pde-based methods, all minimization processes have been run until convergence.

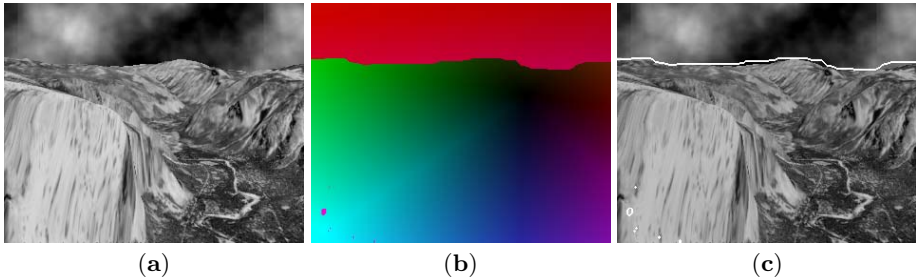
### 5.1 Experiments with Synthetic Data

Figure 2 (a) shows an artificial scene with reflections (but without shadows). The difference image in part (b) shows that only the torus moved (to the right). The flow field depicted in part (c) shows that the proposed motion segmentation model is able to handle reflections and surfaces of non-constant depth. The affine model is not needed here. For all images used to visualize flow fields the color hue indicates the direction of movement whereas the intensity indicates the strength.

Additionally we tested the proposed model on the well-known Yosemite Sequence as shown in figure 3. As is common, we measure the motion between frame 8 (displayed in part a) and frame 9. The difference image gives little information here. Parts (b) and (c) display the segmentation result and the flow field, respectively. As desired, the clouds are separated from the mountains and the valley (except for some small regions in the lower left corner). The average angular error amounts to 11.27. While this error is larger than those reported for non-parametric motion models [8,1], it is still a good value considering the simplicity of the method.



**Fig. 2.** (a) first frame of an artificial scene (b) difference image to second frame (c) motion segmentation by piecewise constant velocity,  $\nu = 9.5$



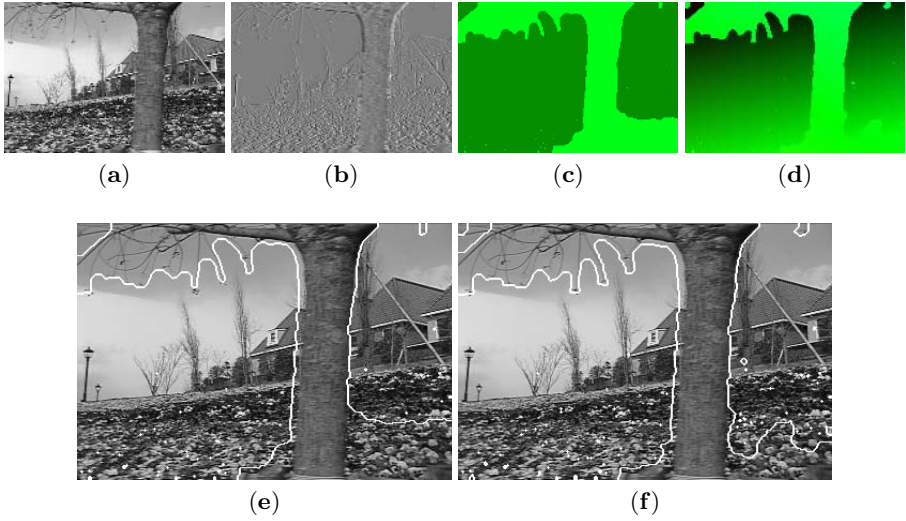
**Fig. 3.** Results for motion segmentation on the Yosemite Sequence between frame 8 (a) and frame 9 (not shown) of the sequence. (b) segmentation obtained with the affine model,  $\nu = 9.5$ . The white line indicates the boundary of the two regions (c) visualization of the flow field.

## 5.2 Experiments with Real Data

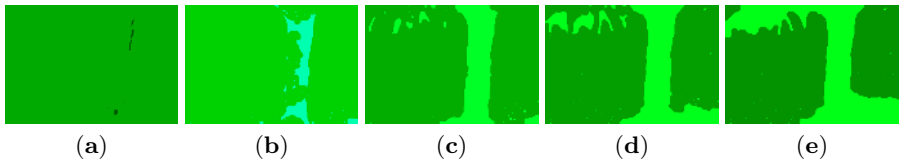
Here we present results on real data. We start with the well-known Flower Garden Sequence shown in figure 4 (a). The motion is estimated between frames 1 and 2. Parts (c-f) show the segmentations and flow fields obtained with constant and affine velocities. The affine model gives the better flow field as the background decreases smoothly in depth. For the segmentation this does not matter much. Figure 5 depicts how the flow field evolves at different stages of the minimization process. We only show this for constant velocities.

When applying the model with constant velocities to every pair of consecutive frames, we obtain an average run-time of 825 msec per frame pair on the full resolution of  $360 \times 240$  on a 3.4 GHz machine. When reducing the resolution to  $180 \times 120$ , this reduces to 180 msec per frame<sup>2</sup>. If the number of iterations is reduced to 2 for each frame pair (excepting the first), 80 msec or 12.5 fps are achieved, which is good enough for real time. Actually, even 1 iteration per frame pair gives sufficient quality, yielding 16.5 fps. Admittedly, the quality reduces slightly compared to full resolution.

<sup>2</sup> The length penalty is then divided by 2 as that is the relative decrease of the length of a line compared to the number of pixels.



**Fig. 4.** Results on the Flower Garden Sequence (a) frame 1 of the sequence (b) difference image to second frame (c) flow field obtained with constant velocities,  $\nu = 4$  (d) flow field obtained with affine velocities,  $\nu = 4$ . Note that the velocity of the background gradually decays toward the top of the image. (e+f) segmentations obtained in (c) and (d). The white lines indicate segmentation boundaries.



**Fig. 5.** Intermediate steps for minimizing the energy functional with constant velocities. 14 iterations were needed until convergence. Here the flow fields after iteration 1 (a), 3 (b), 5 (c), 7 (d) and 12 (e) are shown. Compare figure 4 (c) for the final result.



**Fig. 6.** Results for the Pickup Sequence (a) first frame (b) difference image to second frame (c) segmentation for the model with constant velocities,  $\nu = 3$  (d) segmentation for the affine model,  $\nu = 3$  (e) flow field for the affine model



About 40% of the run-time are used for the computation of the minimum cut. The other major sources are the computation of edge costs and the update of velocities. The affine model more than doubles the run-time.

Lastly, we present results on the Pickup Sequence from [3], which we modified by introducing artificial motion of the whole image (originally only the hand with the can moved). The first frame and the difference image are shown in figure 6 (a) and (b). Part (c) shows that the model with constant velocities is able to separate the hand and the can from the background. As can be seen in part (d) the affine model produces a similar segmentation, but part (e) reveals that the lower parts of the arm are assigned lower velocities as their depth is greater.

## 6 Conclusion

We proposed an efficient semi-discrete optimization method for motion segmentation. Based on the assumption that the velocity in each region can be modeled as a Gaussian distributed random variable, we derived a cost functional for the joint estimation and segmentation of piecewise constant or piecewise affine motion. For the case of two motion models, we developed a fast minimization scheme which alternates a globally optimal segmentation via graph cuts with a globally optimal motion estimation. Experiments show that for moderate resolutions accurate purely motion-based segmentations can be obtained in real-time.

**Acknowledgments.** This work was supported by the German Research Foundation, grant #CR-250/1-1. We thank Thomas Brox for constant support and many helpful discussions and Kalin Kolev for helpful comments on the code.

## References

1. T. Amiaz and N. Kiryati. Dense discontinuous optical flow via contour-based segmentation. In *Int. Conf. on Image Processing*, volume 3, pages 1264–1267, Genova, Italy, Sept. 2005.
2. S. Birchfield and C. Tomasi. Multiway cut for stereo and motion with slanted surfaces. In *IEEE Int. Conf. on Computer Vision*, pages 489–495, 1999.
3. M. Black and A. Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. In *Europ. Conf. on Computer Vision*, LNCS, pages 329–342, Cambridge, England, Apr. 1996. Springer.
4. Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images. In *IEEE Int. Conf. on Computer Vision*, volume 1, pages 105–112, 2001.
5. Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in computer vision. In A. J. M. Figueiredo, J. Zerubia, editor, *Energy Minimization Methods in Computer Vision and Pattern Recognition*, volume 2134 of LNCS, pages 359–374. Springer, 2001.
6. Y. Boykov and V. Kolmogorov. Computing geodesics and minimal surfaces via graph cuts. In *IEEE Int. Conf. on Computer Vision*, 2003.
7. Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 23(11):1222–1239, 2001.

8. T. Brox, A. Bruhn, and J. Weickert. Variational motion segmentation with level sets. In *Europ. Conf. on Computer Vision*, Graz, Austria, May 2006. Springer.
9. D. Cremers and S. Soatto. Motion Competition: A Variational Framework for Piecewise Parametric Motion Segmentation. *Int. J. of Computer Vision*, 62(3):249–265, May 2005.
10. D. Cremers and A. L. Yuille. A Generative Model Based Approach to Motion Segmentation. In B. Michaelis and G. Krell, editors, *Pattern Recognition (Proc. DAGM)*, volume 2781 of *LNCS*, pages 313–320, Magdeburg, Sept. 2003. Springer.
11. L. Ford and D. Fulkerson. *Flows in Networks*. Princeton University Press, Princeton, New Jersey, 1962.
12. D. M. Greig, B. T. Porteous, and A. H. Seheult. Exact maximum *a posteriori* estimation for binary images. *J. Roy. Statist. Soc., Ser. B.*, 51(2):271–279, 1989.
13. B. Horn and B. Schunck. Determining optical flow. *A.I.*, 17:185–203, 1981.
14. B. Jähne, H. Haußecker, H. Scharr, H. Spies, D. Schmundt, and U. Schurr. Study of dynamical processes with tensor-based spatiotemporal image processing techniques. In H. Burkhardt and B. Neumann, editors, *Europ. Conf. on Computer Vision*, volume 1407 of *Lect. Not. Comp. Sci.*, pages 322–336. Springer, 1998.
15. O. Juan. *On Some Extensions of Level Sets and Graph Cuts & Their Applications to Image and Video Segmentation*. PhD thesis, École Nationale des Ponts et Chaussées, May 2006.
16. V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 24(5):657–673, 2004.
17. B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. 7th International Joint Conference on Artificial Intelligence*, pages 674–679, Vancouver, 1981.
18. C. Schnörr. Determining optical flow for irregular domains by minimizing quadratic functionals of a certain class. *Int. J. of Computer Vision*, 6(1):25–38, 1991.
19. J. Wang and E. Adelson. Representating moving images with layers. *IEEE Trans. on Image Processing*, 3(5):625–638, 1994.

# Segmentation-Based Motion with Occlusions Using Graph-Cut Optimization

Michael Bleyer\*, Christoph Rhemann, and Margrit Gelautz

Institute for Software Technology and Interactive Systems  
Vienna University of Technology  
Favoritenstrasse 9-11/188/2, A-1040 Vienna, Austria  
{bleyer, gelautz}@ims.tuwien.ac.at

**Abstract.** We propose to tackle the optical flow problem by a combination of two recent advances in the computation of dense correspondences, namely the incorporation of image segmentation and robust global optimization via graph-cuts. In the first step, each segment (extracted by colour segmentation) is assigned to an affine motion model from a set of sparse correspondences. Using a layered model, we then identify those motion models that represent the dominant image motion. This layer extraction task is accomplished by optimizing a simple energy function that operates in the domain of segments via graph-cuts. We then estimate the spatial extent that is covered by each layer and identify occlusions. Since treatment of occlusions is hardly possible when using entire segments as matching primitives, we propose to use the pixel level in addition. We therefore define an energy function that measures the quality of an assignment of segments *and* pixels to layers. This energy function is then extended to work on multiple input frames and minimized via graph-cuts. In the experimental results, we show that our method produces good-quality results, especially in regions of low texture and close to motion boundaries, which are challenging tasks in optical flow computation.

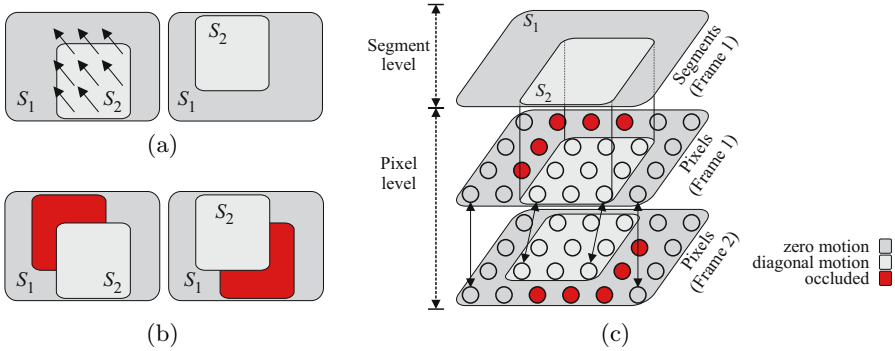
## 1 Introduction

The estimation of optical flow is one of the oldest, but still most active research topics in computer vision. Major challenges are twofold. Firstly, matching often fails in the absence of discriminative image features that can be uniquely matched in the other view. This is the case in *untextured regions* and in the presence of texture with only a single orientation (*aperture problem*). Secondly, a pixel's matching point can be *occluded* in the other view. Those occlusions often occur at motion discontinuities, which makes it specifically challenging to precisely outline object boundaries. In spite of its obvious importance, standard optical flow approaches still tend to ignore the occlusion problem (e.g., [1,2,3]).

This paper proposes an algorithm that explicitly addresses these problems by taking advantage of *colour segmentation* and robust optimization via *graph-cuts*. Our contribution lies in that we show how to set up an energy function

---

\* This work was funded by the Austrian Science Fund (FWF) under project P15663.



**Fig. 1.** The occlusion problem in segmentation-based matching and our proposed solution. Explanation is given in the text.

that formulates segmentation-based motion with treatment of occlusions. The advantage of this energy function is that it can be optimized via robust graph-cut-based optimization. The motivation for using colour segmentation is that energy minimization approaches often bias towards the reconstruction of simple object shapes and consequently fail in the presence of complex motion discontinuities. To explain the idea behind our energy function, let us consider the two views illustrated in Fig. 1a. The images show two segments  $S_1$  and  $S_2$  at different instances of time with segment  $S_2$  undergoing motion as indicated by the arrows. As a consequence of the moving foreground object, occlusions occur in both frames (coloured red in Fig. 1b).  $S_1$  is partially affected by occlusions, which is problematic in the following sense. When using segments as matching primitives, we can only state that the *complete* segment  $S_1$  has zero motion. However, we cannot express the fact that some pixels of  $S_1$  are affected by occlusion. In other words, *occlusions cannot be dealt with in the domain of segments*.

In order to correctly model occlusions, we propose an energy function that operates on two levels, one representing the extracted segments and the other representing pixels. In addition to all segments (top layer of Fig. 1c), we as well assign every pixel of the reference image to a motion model (middle layer of Fig. 1c). The basic idea is to enforce that every (visible) pixel is assigned to the same motion model as the segment to which it belongs. However, and this is the important point, a pixel is also allowed to be occluded. Finally, we as well include every pixel of the second image into our problem formulation (bottom layer of Fig. 1c). We enforce that a (visible) pixel and its matching point in the other image must both have identical motion model assignments. This constraint serves to implement the uniqueness assumption [4]. This assumption is used to identify occlusions symmetrically in both images.

In relation to prior work, using colour segmentation for the dense correspondence problem does not represent a novel idea. Black and Jepson [5] propose a colour segmentation-based motion algorithm that fits a variable order parametric model to each individual segment using a precomputed flow field. Analogous to

our approach, the basic idea behind this procedure is that the flow field is likely to vary smoothly inside such a segment. However, the authors do not account for the occlusion problem and miss to model smoothness across segments. Recently, segmentation-based techniques have also gained attention in the stereo community (e.g., [6,7]). Although quite different from each other, segmentation-based stereo methods take benefit from increased robustness in untextured regions and in areas close to disparity discontinuities. This is well reflected by the good experimental results of those algorithms. For the motion layer extraction problem, segmentation-based techniques using clustering methods are proposed in [8,9].

In the context of energy minimization approaches, our technique is most closely related to various motion segmentation algorithms. Ayer and Sawhney [10] employ the minimum description length (MDL) encoding principle in order to derive the smallest set of layers necessary to describe the image motion. They formulate statistical cost functions that are optimized by an expectation maximization algorithm. Willis et al. [11] present a graph-cut-based approach to achieve a dense and piecewise smooth assignment of pixels to layers. They do, however, not explicitly model the occlusion problem. In contrast to this, Xiao and Shah [12] embed occlusion detection into a graph-cut-based method in a very recent work. They claim to be the first ones to deal with the explicit identification of occluded pixels for the motion segmentation task. The most obvious difference to those approaches is that none of them uses image segmentation.

Among prior work, the closest related one originates from literature on the simpler stereo correspondence problem. Hong and Chen [7] combine colour segmentation-based matching with graph-cut optimization. They heuristically identify occlusions in a preprocessing step, which then allows them to model the correspondence problem on the segment level only. However, the results of this method depend on the success of this preprocessing step, and it is not clear how well an a-priori identification of occlusions can work, especially in the presence of large motion. In contrast to this, our energy function knows about the existence of occlusions. Flow vectors and occlusions are computed simultaneously, which we believe results in a more accurate reconstruction of both.

## 2 Our Approach

### 2.1 Colour Segmentation and Initial Models

In the first step, we apply colour segmentation to the reference image. Since our basic assumption states that the flow values inside a colour segment vary smoothly, it is important that a segment does not overlap a motion discontinuity. It is therefore safer to use oversegmentation (Fig. 2b). In the current implementation, we apply the mean-shift-based segmentation algorithm described in [13].

The optical flow inside each segment is modelled by affine motion, which is

$$\begin{aligned} V_x(x, y) &= a_{x0} + a_{xx}x + a_{xy}y \\ V_y(x, y) &= a_{y0} + a_{yx}x + a_{yy}y \end{aligned} \quad (1)$$

with  $V_x$  and  $V_y$  being the x- and y-components of the flow vector at image coordinates  $x$  and  $y$  and the  $a$ 's denoting the six parameters of the model. However, our approach could easily be extended to a more sophisticated model. To initialize the motion of each segment, a set of sparse correspondences is computed using the KLT-tracker [14]. A segment's affine parameters are then derived by least squared error fitting to all correspondences found inside this segment. We apply the iterative plane fitting algorithm described by Tao et al. [6] to reduce the sensitivity of the least squared error solution to outliers.

## 2.2 Layer Extraction

When using a layered representation [15], the first questions one has to answer are: How many layers are present in the sequence and what are their motion parameters? Initially, the set of our layers  $\mathcal{L}$  is built by all motion models found in the previous step. In order to extract a small set of layers out of  $\mathcal{L}$ , we minimize a simple energy function  $E(f)$ , which measures the optimality of an assignment  $f$  of segments to layers, in the form of

$$E(f) = E_{data}(f) + E_{smooth}(f). \quad (2)$$

The data term  $E_{data}$  calculates how well  $f$  agrees with the input images and is defined by

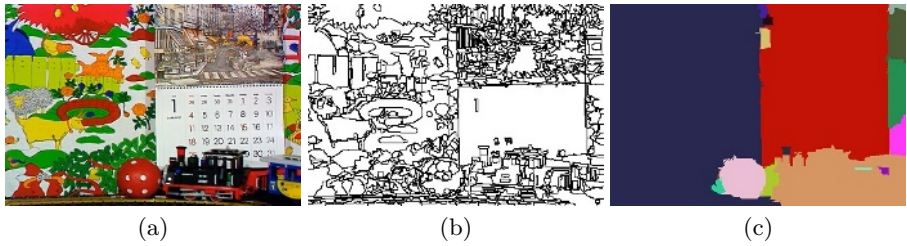
$$E_{data}(f) = \sum_{S \in \mathcal{R}} \sum_{p \in S} d(p, m[f(S)](p)) \quad (3)$$

with  $\mathcal{R}$  being the set of all segments of the reference view and  $f(S)$  being the index of the layer to which segment  $S$  is assigned. We write  $m[k](p)$  to denote the matching point of a pixel  $p$  in the other view according to the  $k$ th layer. More precisely,  $m[k](p)$  is derived by computing the displacement vector at  $p$  using the affine parameters of the layer at index  $k$  (equation (1)) and adding it to the coordinates of  $p$ . The function  $d(\cdot, \cdot)$  measures the dissimilarity of two pixels, which is the sum-of-absolute-differences of RGB values in our implementation. The second term  $E_{smooth}$  of the energy function measures to which extent the current assignment  $f$  is spatially smooth.  $E_{smooth}$  is defined by

$$E_{smooth}(f) = \sum_{(S, S') \in \mathcal{N}} \begin{cases} \lambda_{smooth} \cdot b(S, S') & : f(S) \neq f(S') \\ 0 & : \text{otherwise} \end{cases} \quad (4)$$

with  $\mathcal{N}$  being all pairs of neighbouring segments,  $b(\cdot, \cdot)$  computing the border length between such and  $\lambda_{smooth}$  being a constant user-defined penalty.

We approximate the minimum of the energy function in equation (2) using the  $\alpha$ -expansion algorithm of Boykov et al. [16]. Starting from an arbitrary configuration, we iteratively change this configuration by computing the optimal  $\alpha$ -expansion move for each layer until convergence. The graph built for calculating the optimal  $\alpha$ -expansion consists of nodes that correspond to segments. Since the number of segments is significantly lower than the number of pixels, minimization of equation (2) via graph-cuts is quite efficient.



**Fig. 2.** Colour segmentation and layer extraction. (a) Original image. (b) Result of the colour segmentation step. Segment borders are shown. (c) Result of the layer extraction step. Pixels of the same colour belong to the same layer.

Those layers that are not present in the computed configuration  $f^*$  are removed from the set of layers  $\mathcal{L}$ , which drastically decreases the number of layers. However, it is quite likely that the correct layer was not contained in our initial set, due to the small spatial extent over which the motion parameters were initially computed. We therefore refit the layers over their new spatial extents according to the assignment of segments to layers in  $f^*$  to derive a set of refined layers  $\mathcal{L}'$ . We then update  $\mathcal{L}$  by  $\mathcal{L} := \mathcal{L} \cup \mathcal{L}'$ . Starting from the configuration  $f^*$ , we apply the  $\alpha$ -expansion algorithm using our refined layer set  $\mathcal{L}$  to obtain the new configuration  $f^{**}$ . We again remove those layers from  $\mathcal{L}$  that do not occur in  $f^{**}$ . If the costs of  $f^{**}$  are not lower than those of  $f^*$ ,  $\mathcal{L}$  represents our final set of layers. Otherwise, this procedure is iterated.

We show results of the layer extraction step in Fig. 2c. Since the proposed algorithm operates on the segment level only, it is not capable of handling occlusions. It therefore produces artefacts in regions close to motion boundaries. Although there are only small occluded areas in the sequence shown in Fig. 2 such artefacts are visible in the proximity of the rotating ball.<sup>1</sup> However, this strategy works well enough to deliver the dominant image motion and it is computationally efficient.

### 2.3 Layer Assignment

Knowing the set of layers, the task of the assignment step is to estimate which parts of the images are covered by which layers as well as to identify occlusions. As stated in the introduction, the segment level alone is not sufficient for treatment of occlusions. In the following, we therefore design an energy function involving both, the segment and the pixel level. Minimization of the derived objective function via the  $\alpha$ -expansion algorithm is not discussed in this paper for space limitations, but is thoroughly described in [17].

**Energy Function.** In contrast to the previous section, a configuration  $f$  is no longer an assignment of segments to layers, but an assignment of segments

<sup>1</sup> We will present an example where this effect is more severe in the experimental results.

and pixels to layers. Moreover, a pixel can be assigned to a dedicated label 0 expressing the fact that the pixel's matching point is occluded in the other view. We define the energy function  $E'(f)$  measuring the quality of a configuration  $f$ , which assigns segments and pixels to layers, by

$$E'(f) = E'_{data}(f) + E'_{segment}(f) + E'_{mismatch}(f) + E'_{smooth}(f). \quad (5)$$

The individual terms of  $E'(f)$  are described one after the other in the following.

The first term  $E'_{data}$  measures the agreement of  $f$  with the input data and is defined by

$$E'_{data}(f) = \sum_{p \in \mathcal{I}} \begin{cases} d(p, m[f(p)](p)) & : f(p) \neq 0 \\ \lambda_{occ} & : \text{otherwise} \end{cases} \quad (6)$$

with  $\mathcal{I}$  being the set of all pixels of the reference image  $\mathcal{I}_R$  as well as of the second view  $\mathcal{I}_S$  and  $\lambda_{occ}$  denoting a constant predefined penalty. While  $E'_{data}$  measures the pixel dissimilarity for visible pixels, it imposes a penalty on occluded ones. This penalty is necessary, since otherwise declaring all pixels as occluded would result in a trivial minimum of  $E'(f)$ . To allow for a symmetrical identification of occlusions,  $E'_{data}$  operates on both images. The matching point  $m[k](p) \in \mathcal{I}_R$  of a pixel  $p \in \mathcal{I}_S$  is thereby computed using the inverse motion model of the  $k$ th layer. The second term  $E'_{segment}(f)$  of the energy function enforces the segmentation information on the pixel level and is defined by

$$E'_{segment}(f) = \sum_{p \in \mathcal{I}_R} \begin{cases} \infty & : f(p) \neq 0 \wedge f(p) \neq f(seg(p)) \\ 0 & : \text{otherwise} \end{cases} \quad (7)$$

with  $seg(p)$  being a function that returns the segment to which pixel  $p$  belongs. The basic idea is that a pixel is either occluded or assigned to the same layer as all other visible pixels of the same segment. Solutions that violate this constraint generate infinite costs. The third term  $E'_{mismatch}$  accounts for a consistent layer assignment across the reference and the second images. It is defined by

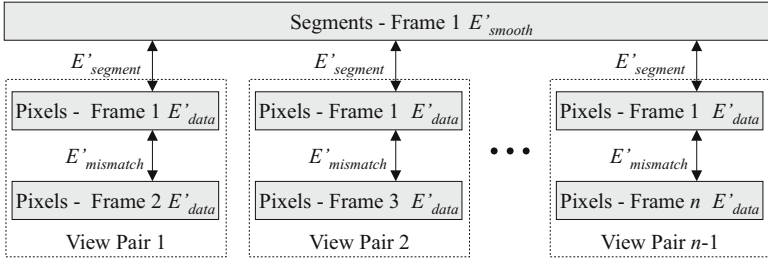
$$E'_{mismatch}(f) = \sum_{p \in \mathcal{I}} \begin{cases} \lambda_{mismatch} & : f(p) \neq 0 \wedge f(p) \neq f(m[f(p)](p)) \\ 0 & : \text{otherwise} \end{cases} \quad (8)$$

with  $\lambda_{mismatch}$  being a user-set penalty. This penalty is imposed for each pixel  $p$  whose matching point is assigned to a different layer than that of  $p$ . Finally, we apply the smoothness assumption on the segment level.  $E'_{smooth}$  is identical to the smoothness term of the previous section. For completeness, we write:

$$E'_{smooth}(f) = E_{smooth}(f). \quad (9)$$

**Extension to Multiple Input Frames.** The energy function of equation (5) is designed to be used with only two input images. However, oftentimes frames in between these two images are available as well and can be used to improve the matching results. Let  $I_1$  and  $I_n$  be the first and last views of a short video





**Fig. 3.** Conceptual view of the energy function  $E'(f)$

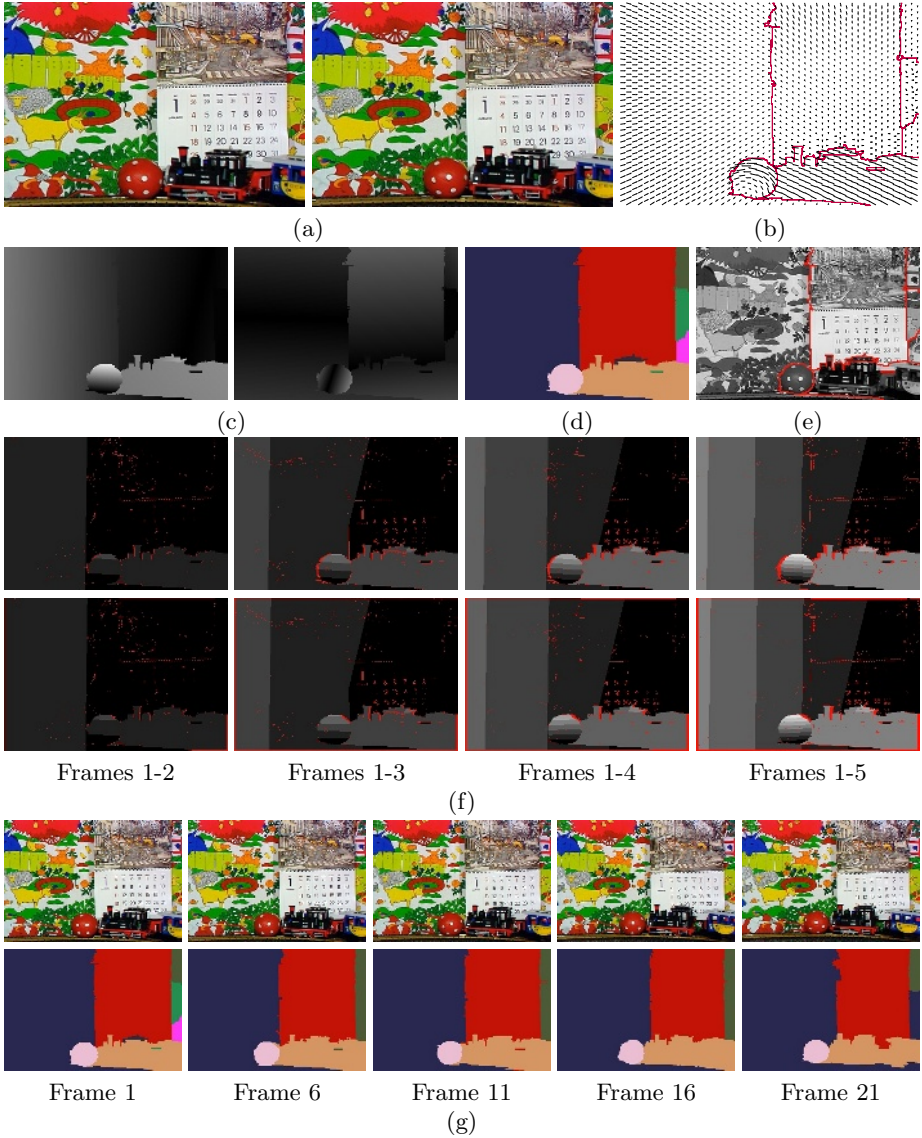
clip of  $n$  frames. For computing the optical flow between  $I_1$  and  $I_n$ , we do not only match  $I_1$  against  $I_n$ , but also match  $I_1$  against any intermediate view  $I_k$  with  $1 < k < n$ . The basic idea behind this is that a pixel of the reference frame  $I_1$ , which is occluded when matching  $I_1$  and  $I_n$ , might be visible (and therefore matchable) when computing the correspondences between  $I_1$  and  $I_k$ . This concept was originally used by Xiao and Shah [12,18].

To implement this idea, we split up a sequence of  $n$  images into  $n - 1$  view pairs. Each view pair consists of the reference frame  $I_1$ , on which we apply colour segmentation, and a second image  $I_k \neq I_1$ , i.e. we derive the view pairs  $I_1 - I_2$ ,  $I_1 - I_3, \dots, I_1 - I_n$ . From the layer extraction step, we have the dominant motion models of the view pair  $I_1 - I_n$ . For simplicity, we assume that within a very short image sequence the motion is linear, so that the motion models for the other view pairs can be linearly interpolated from those. To propagate the layer assignments of the individual view pairs between each other, we connect the reference frame  $I_1$  of each view pair to the segment level using the term  $E'_{segment}$  (Fig. 3). From its definition in equation (7),  $E'_{segment}$  enforces a pixel of the reference view to have the same layer assignment as its corresponding segment, unless the pixel is occluded. Since the reference frames of all view pairs are now connected to the segment level, a pixel  $p$  of  $I_1$  in view pair  $VP$  that is assigned to layer  $l$  has to be assigned to  $l$  in any other view pair  $VP'$  or carry the occlusion label. This is what Xiao and Shah refer to as the *General Occlusion Constraint* [18], which is integrated into our energy function without additional effort.

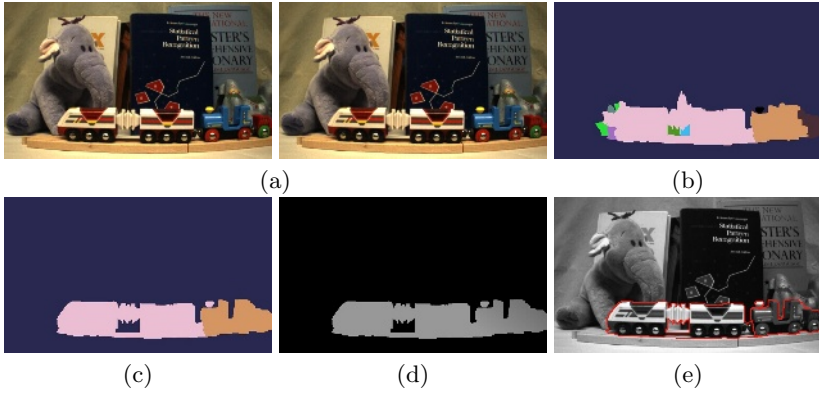
### 3 Experimental Results

We have tested our algorithm on a standard test set (Fig. 4) as well as on a self-recorded one (Fig. 5). Throughout our test runs, we set  $\lambda_{occ} := \lambda_{mismatch} - 1$ . The effect of this is that every view inconsistent pixel is labelled as occluded on the pixel level. More precisely, if two pixels assigned to different layers project to the same pixel of the other view, one of them is view inconsistent and has to be declared as occluded. Therefore, the uniqueness constraint is enforced.

As a first test sequence, we have picked five frames from the *Mobile & Calendar* sequence (Fig. 4a). Within this sequence, there is translational motion



**Fig. 4.** Results for the *Mobile & Calendar* sequence. (a) Frames 1 and 5 of five input frames. (b) Flow vectors with layer boundaries outlined. (c) Absolute x- and y-components of the computed flow vectors. (d) Assignment of segments to layers. (e) Layer boundaries coloured in red superimposed on input frame 1. (f) Absolute x-components of the flow vectors on the pixel level. The top row shows the reference view (frame 1), while the match images (frames 2 – 5) are presented at the bottom. Pixels carrying the occlusion label are coloured in red. (g) Motion segmentation for each fifth frame of the complete sequence.



**Fig. 5.** Results for a self-recorded sequence. (a) Frames 1 and 3 of three input frames. (b) Results of the layer extraction step. (c) Assignments of segments to layers. (d) Absolute x-components. (e) Layer borders superimposed on view 1.

on the train and the poster, while rotational motion originates from the ball. Furthermore, the camera zooms out. Results computed by our method (Figs. 4b–g) indicate that the algorithm is well suited to precisely delineate motion discontinuities. Moreover, our technique can equivalently be regarded as a motion segmentation method, since the layer assignment result (Fig. 4d) divides the image into homogeneously moving regions. In Fig. 4g, we apply our algorithm to segment the complete sequence into video objects that undergo homogeneous motion. A more detailed explanation of this process is, however, found in [17].

In addition to the standard test set, we tested the proposed method on a self-recorded sequence (Fig. 5a). The sequence shows a train moving from right to left in front of a static background. Although the motion is relatively simple, the scene contains complex motion boundaries (e.g., the link connecting the wagons) and large occluded areas. These occlusions are the reason why the layer extraction step delivers poor results in the proximity of the motion discontinuities (Fig. 5b). In contrast to this, the assignment step that explicitly models occlusions seems to be able to outline the motion boundaries correctly (Fig. 5c).

## 4 Discussion

We have presented a layered segmentation-based algorithm for the estimation of dense motion correspondences. In the layer extraction step, we optimize a simple energy function on the segment level. Since the segment level alone is not sufficient to handle occlusions, we define an energy function that operates on the segment and on the pixel level in the assignment step. This energy function is extended to allow for the computation of the motion between multiple images. Our method determines correct flow information in traditionally challenging regions such as areas of low texture and close to motion discontinuities.

Further research will concentrate on overcoming two limitations of our approach. The algorithm currently describes the image motion using the affine model. This may result in an oversimplification of the real motion, especially in the presence of large motion. However, the affine model could easily be replaced by a more sophisticated one without major changes in our implementation. A more severe problem is that the segmentation assumption is not guaranteed to hold true. Our current remedy to this is to apply a strong oversegmentation. However, since this does not completely overcome this problem, our algorithm could take benefit from an operation that allows splitting segments.

## References

1. Horn, B., Schunck, B.: Determining optical flow. *Artificial Intelligence* **17** (1981) 185–203
2. Lucas, B., Kanade, T.: An iterative image registration technique with an application to stereo. *IJCAI* (1981) 121–130
3. Brox, T., Bruhn, A., Papenber, N., Weickert, J.: High accuracy optical flow estimation based on a theory for warping. In: *ECCV*. Volume 4. (2004) 25–36
4. Zitnick, C., Kanade, T.: A cooperative algorithm for stereo matching and occlusion detection. *TPAMI* **22**(7) (2000) 675–684
5. Black, M., Jepson, A.: Estimating optical flow in segmented images using variable-order parametric models with local deformations. *TPAMI* **18**(10) (1996) 972–986
6. Tao, H., Sawhney, H., Kumar, R.: A global matching framework for stereo computation. In: *ICCV*. (2001) 532–539
7. Hong, L., Chen, G.: Segment-based stereo matching using graph cuts. In: *CVPR*. Volume 1. (2004) 74–81
8. Ke, Q., Kanade, T.: A subspace approach to layer extraction. In: *CVPR*. (2001) 255–262
9. Altunbasak, Y., Eren, P., Tekalp, A.: Region-based parametric motion segmentation using color information. *GMIP* **60**(1) (1998) 13–23
10. Ayer, S., Sawhney, H.: Layered representation of motion video using robust maximum-likelihood estimation of mixture models and mdl encoding. In: *ICCV*. (1995) 777–784
11. Willis, J., Agarwal, S., Belongie, S.: What went where. In: *CVPR*. (2003) 37–44
12. Xiao, J., Shah, M.: Motion layer extraction in the presence of occlusion using graph cuts. *TPAMI* **27**(10) (2005) 1644–1659
13. Christoudias, C., Georgescu, B., Meer, P.: Synergism in low-level vision. In: *ICPR*. Volume 4. (2002) 150–155
14. Shi, J., Tomasi, C.: Good features to track. In: *CVPR*. (1994) 593–600
15. Wang, J., Adelson, E.: Representing moving images with layers. *Transactions on Image Processing* **3**(5) (1994) 625–638
16. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. *TPAMI* **23**(11) (2001) 1222–1239
17. Bleyer, M.: Segmentation-based Stereo and Motion with Occlusions. PhD thesis, Vienna University of Technology (2006)
18. Xiao, J., Shah, M.: Accurate motion layer segmentation and matting. In: *CVPR*. Volume 2. (2005) 698–703

# Realtime Depth Estimation and Obstacle Detection from Monocular Video

Andreas Wedel<sup>1,2</sup>, Uwe Franke<sup>1</sup>, Jens Klappstein<sup>1</sup>,  
Thomas Brox<sup>2</sup>, and Daniel Cremers<sup>2</sup>

<sup>1</sup> DaimlerChrysler Research and Technology, REI/AI,  
71059 Sindelfingen, Germany

<sup>2</sup> Computer Vision and Pattern Recognition Group,  
Rheinische Friedrich-Wilhelms Univeristät, 53117 Bonn, Germany

**Abstract.** This paper deals with the detection of arbitrary static objects in traffic scenes from monocular video using structure from motion. A camera in a moving vehicle observes the road course ahead. The camera translation in depth is known. Many structure from motion algorithms were proposed for detecting moving or nearby objects. However, detecting stationary distant obstacles in the focus of expansion remains quite challenging due to very small subpixel motion between frames. In this work the scene depth is estimated from the scaling of supervised image regions. We generate obstacle hypotheses from these depth estimates in image space. A second step then performs testing of these by comparing with the counter hypothesis of a free driveway. The approach can detect obstacles already at distances of 50m and more with a standard focal length. This early detection allows driver warning and safety precaution in good time.

## 1 Introduction

Automatic detection and verification of objects in images is a central challenge in computer vision and pattern analysis research. An important application is robustly hypothesizing and verifying obstacles for safety applications in intelligent vehicles. The practical value of such systems becomes evident as obstacle



**Fig. 1.** Six out of ten front-end crashes could be prevented if safety systems reacted a split second earlier than the driver. Detecting arbitrary obstacles from monocular video in the road course ahead, however, is quite challenging.

detection is a prerequisite to warn the driver of approaching hazards (see Fig. 1). Commonly used radar sensors lack detecting static objects therefore we tackle this problem using computer vision.

For a camera mounted in a vehicle with a given camera translation in depth, detecting obstacles in traffic scenes has three major challenges:

1. The algorithm has to run in real time with minimum delay in reaction time.
2. Obstacles have to be detected at large distances to route the vehicle or warn the driver as early as possible.
3. The position and horizontal dimension of an obstacle have to be estimated precisely to safely guide the vehicle in case an emergency brake is insufficient.

The first two challenges demand an algorithm able to detect obstacles in the focus of expansion where optical flow displacement vectors between consecutive frames are extremely small. Overcoming this by skipping frames violates the first constraint. The last challenge requires robust verification of obstacle boundaries.

Traditional vision based obstacle detection relies on depth estimation from stereo systems [7]. Such systems work well. However, single cameras are already available in series production performing numerous vision based driver assistance algorithms such as intelligent headlight control and night view. Obstacle detection from a single camera is, hence, a desirable alternative.

According to [1,14] obstacle detection in monocular vision can be split into methods employing a-priori knowledge and others based on the relative image motion. Former algorithms need to employ strict assumptions regarding the appearance of observed objects. Since we are interested in a model free approach, we have to use latter methods. Proposed realtime optical flow algorithms [3,11,13] and obstacle detection based on those [12] calculate the displacement between consecutive frames of an image sequence. In such a basic approach, integrating flow vectors over successive image pairs is subject to drifts and therefore these algorithms are not suitable for the posed problem. Moreover, these methods detect obstacles in two steps firstly calculating flow vectors for every pixel and secondly analyzing those flow vectors. Working directly in image space is more desirable as all the information available is accessed directly.

We propose an obstacle detection algorithm in the two standard steps:

1. **Hypothesis generation from estimating scene depth** in image space.
2. **Candidate testing by analyzing perspective transformation** over time.

In the first step, conclusions about scene depth are drawn from the scaling factor of image regions, which is determined using region tracking. We use the tracking algorithm described in [8] which is consistent over multiple frames of an image sequence and directly estimates scale and translation in image space. For an evaluation of different tracking algorithms we refer to [5]. If distance measurements fall below a given threshold, obstacle hypotheses are generated. Due to the restricted reliability of depth from region scaling, such an approach can result in false hypotheses which have to be dismissed.

The testing of generated hypotheses is performed in the second step. We check whether the observed perspective distortion over time corresponds to an obstacle

with distance given from hypothesis generation or to a free driveway. In such an approach we are able to detect arbitrary obstacles directly in image space. The two steps will be investigated separately in Sects. 2 and 3. Experimental results on real image data can be found in Sect. 4. A final conclusion and motivation for further work will be given in Sect. 5.

## 2 Depth Tracking from Monocular Video

This section investigates the mathematical foundation for reconstructing scene depth from monocular vision. First we describe the underlying perspective projection and the model used for depth computation. Then we describe how depth information can be computed from scaling of image regions and how this fact can be used to efficiently detect stationary obstacles. Finally we investigate the error in depth estimation.

We use a monocular camera mounted on a vehicle such that the camera’s optical axis  $e_3$  coincides with the vehicle translation direction. The reference system is the left-handed camera coordinate system  $(0, e_1, e_2, e_3)$  with the  $e_2$  unit vector being the ground plane normal. In particular, we assume a flat ground and a straight road to travel on. The image plane has equation  $Z = f$ , where  $f$  is the focal length of the camera. The ground plane is  $Y = -Y_0$  with the camera height  $Y_0$ . For a point  $\mathbf{X} = (X, Y, Z)^T$  in 3-D space we obtain the corresponding image point  $\mathbf{x} = (x, y)^T$  by a perspective projection:

$$\mathbf{x} = \frac{f}{Z} \begin{pmatrix} X \\ -Y \end{pmatrix}. \tag{1}$$

In practice the camera coordinate system  $e_3$  axis usually is not parallel to the ground plane. Camera rotation can be compensated transforming the camera to a virtual forward looking camera in a similar way as described in [9]. The camera translation in depth between consecutive frames is known from inertial sensors.

Obstacles are assumed to be axis parallel bounded boxes. This states that the  $Z$  coordinate of the obstacle plane facing the camera is constant. In practice the relative depths on obstacle surfaces are small compared to the distance between obstacle and camera such that this assumption is a good approximation.

Let  $\mathbf{X}(t) = (X(t), Y(t), Z(t))^T$  be a point at time  $t$  and  $\mathbf{x}(t)$  its projected image point. The camera translation in depth between time  $t$  and  $t + \tau$  is  $\mathbf{T}(t, \tau)$  leading to  $\mathbf{X}(t + \tau) = \mathbf{X}(t) + \mathbf{T}(t, \tau)$ . The camera translational and rotational velocity is  $\dot{\mathbf{T}}(t)$  and  $\dot{\mathbf{\Omega}}(t)$  respectively. Particular coordinates are represented by subscripted characters. Traditional structure from motion algorithms based on optical flow involve using the image velocity field mentioned by Longuet-Higgins and Prazdny in [10] (the time argument is dropped due to better readability):

$$\dot{\mathbf{x}} = \frac{1}{Z} \begin{pmatrix} x\dot{T}_Z - f\dot{T}_X \\ y\dot{T}_Z - f\dot{T}_X \end{pmatrix} - \begin{pmatrix} \dot{\Omega}_X \frac{xy}{f} + \dot{\Omega}_Y \left( f + \frac{x^2}{f} \right) + \dot{\Omega}_Z y \\ \dot{\Omega}_X \left( f + \frac{y^2}{f} \right) + \dot{\Omega}_Y \frac{xy}{f} + \dot{\Omega}_Z x \end{pmatrix}. \tag{2}$$

Such algorithms are exact for time instances where image velocities are measurable. However, flow vectors measure the displacement of image points between frames. Therefore resolving (2) using an explicit or implicit integration method induces drifts by adding up errors in inter-frame motions. We divide the motion of image regions into two parts and show that under the given conditions the scene depth can be estimated solely by estimating the scaling factor of image regions. The transformation of an image point for a pure translation using (1) becomes

$$\begin{aligned} \mathbf{x}(t + \tau) &= \frac{f}{Z(t + \tau)} \begin{pmatrix} X(t + \tau) \\ Y(t + \tau) \end{pmatrix} = \frac{f}{Z(t) + T_Z(t, \tau)} \begin{pmatrix} X(t) + T_X(t, \tau) \\ Y(t) + T_Y(t, \tau) \end{pmatrix} \quad (3) \\ &= \underbrace{\frac{Z(t)}{Z(t) + T_Z(t, \tau)}}_{s(t, \tau)} \underbrace{\frac{f}{Z(t)} \begin{pmatrix} X(t) \\ Y(t) \end{pmatrix}}_{\mathbf{x}(t)} + \frac{f}{Z(t) + T_Z(t, \tau)} \begin{pmatrix} T_X(t, \tau) \\ T_Y(t, \tau) \end{pmatrix}. \quad (4) \end{aligned}$$

It should be pointed out, that we use absolute image coordinates and not velocities for computation. With a correctly given vehicle translation and displacement of image points, scene depth can be directly calculated over large time scales. As only the translation in depth  $T_Z(t, \tau)$  is known, a single observation is not sufficient to determine scene depth. With the assumed model though, front faces of obstacles have equal  $Z$  coordinates and therefore multiple observations in an image region can be used to solve an over-determined equation system for the scaling factor and the translation.

The key for depth reconstruction is to use the scale  $s(t, \tau)$  directly obtained by the used region tracking over multiple frames to calculate scene depth:

$$d \equiv Z(t) = \frac{s(t, \tau)}{1 - s(t, \tau)} T_Z(t, \tau). \quad (5)$$

**Distance histogram.** For reconstructing scene depth we observe the image region to which obstacles in 30m distance with 0.9m height are mapped (compare Fig. 4). This region is divided up into  $n$  overlapping image regions  $\{R_i\}_{i=0}^n$ , which are individually tracked until their correlation coefficient surpasses a fixed threshold. The estimated distances of the tracked regions are projected onto the  $x$ -axis in image space (which can be regarded as a discrete resolution of the viewing angle) to receive a distance histogram. Projected obstacle distances are weighted by distance from region center. An appropriate weighting function is the triangular hat function  $\Delta_{R_i}$  defined on the region width. With the image region distance  $d(R_i)$  and the characteristic function  $\chi_{R_i}(x) = 1 \Leftrightarrow x \in R$  this results in the following distance histogram (see Fig. 5):

$$d(x) = \frac{1}{\sum_i \chi_{R_i}(x) \Delta_{R_i}(x)} \sum_i \chi_{R_i}(x) \Delta_{R_i}(x) d(R_i). \quad (6)$$



**Error in depth estimation.** In this paragraph we will show how depth variance can be calculated by error propagation taking into account errors due to rotation as well. In real case scenarios rotational effects occur as steer angle, shocks and vibrations experienced by the vehicle can introduce rapid and large transients in image space.

Recalling that (2) involves velocities, we use explicit Euler integration for modeling the incremental rotational transformation. For the analysis of rotational errors the translational part can be set to zero leading to:

$$\mathbf{x}(t + \tau) = \mathbf{x}(t) + \tau \begin{pmatrix} \dot{\Omega}_X(t) \frac{x(t)y(t)}{f} + \dot{\Omega}_Y(t) \left( f + \frac{x(t)^2}{f} \right) + \dot{\Omega}_Z(t)y(t) \\ \dot{\Omega}_X(t) \left( f + \frac{y(t)^2}{f} \right) + \dot{\Omega}_Y(t) \frac{x(t)y(t)}{f} + \dot{\Omega}_Z(t)x(t) \end{pmatrix} . \quad (7)$$

The constant terms in (7) will influence only the translation estimation and therefore keep the scaling factor unchanged. The influence of the roll rate ( $\dot{\Omega}_Z$ ) when looking at each image coordinate equation by itself is constant, too. The yaw rate ( $\dot{\Omega}_Y$ ) and pitch rate ( $\dot{\Omega}_X$ ) are linear and quadratic in the image coordinates and therefore will influence the scale factor. Let  $s$  be the estimated scale factor,  $e_s$  the error in scale estimation, and  $\hat{s}$  the true scale with zero rotation. From (7) it follows that

$$s = \hat{s} + e_s + cx + cy . \quad (8)$$

However, assuming the yaw and pitch angle to be bounded by  $\pm 10^\circ$  and the focal length to be greater than 800px leads to

$$c \equiv \frac{\tau \dot{\Omega}}{f} \in [-2.2 \cdot 10^{-4}, 2.2 \cdot 10^{-4}] . \quad (9)$$

The limited image size (of  $640 \times 480$  pixel) and the bounded values of the rotation parameters therefore limit the effect on estimation of region scale. With known scale variance from tracking  $\sigma_s^2$  and variance in translation  $\sigma_{T_Z}^2$  the depth variance can be calculated by error propagation from (5) and (8) via:

$$\sigma_d^2 = \frac{1}{(1-s)^2} T_Z (\sigma_s + x\sigma_c + y\sigma_c)^2 + \frac{s}{1-s} \sigma_{T_Z}^2 . \quad (10)$$

It has to be pointed out that the relative error in scale estimation becomes smaller as the scale factor increases, such that the influence of rotation on the scaling factor becomes negligible over large time scales (see Fig. 3).

The next section deals with obstacle detection based on the distance histogram from (6). The separation between depth estimation and obstacle detection allows for usage of distance histograms generated by alternative sensors (e.g. a scanning radar) for a sensor-fusion. Results obtained from depth estimation can be found in Sect. 4.

### 3 Obstacle Detection by Hypothesis Verification

The distance histogram from the previous section can serve to find potential obstacles. If any entry in the image distance histogram falls below a fixed distance threshold, an obstacle hypothesis is created. As pointed out in [6], robust computer vision algorithms should provide not only parameter estimates but also quantify their accuracy. Although we get the distance accuracy of an obstacle hypothesis by error propagation from tracking, this does not evaluate the probability of an obstacle's pure existence. This section describes obstacle detection by hypothesis testing resulting in a quality specified output.

Let  $d$  be the distance of an obstacle hypothesis drawn from the distance histogram. With the known camera translation in depth  $T_Z$  the transformation of the obstacle in image space using (5) is

$$\mathbf{x}' = V(\mathbf{x}) = \begin{pmatrix} \frac{d-T_Z}{d} & 0 \\ 0 & \frac{d-T_Z}{d} \end{pmatrix} \mathbf{x} . \quad (11)$$

The counter hypothesis of a free driveway with plane equation  $e_2 = -Y_0$  will be transformed in image space using homogeneous coordinates according to

$$\mathbf{x}' = Q(\mathbf{x}) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \frac{T_Z}{Y_0} & 1 \end{bmatrix} \mathbf{x} . \quad (12)$$

Obviously this is only true for the ground plane up to the projected horizon. The hypothesis *no obstacle* above the horizon is set to be the identity (as this is equivalent with obstacles being infinitely distant).

**Hypothesis testing.** Let  $F(\mathbf{x})$  and  $G(\mathbf{x})$  be the intensity value for the initial image and the image after vehicle translation respectively. We assume a Gaussian distribution of the intensity values and fixed standard deviation, thus for an image transformation function  $f$  corresponding to an image region  $R$  we get

$$p_R(f) \propto e^{-|G-F|^2} \quad (13)$$

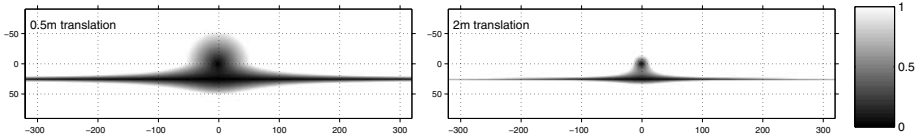
with  $|G - F|^2$  being the *sum of squared differences* defined as

$$-\log(p_R(f)) = \sum_{\mathbf{x} \in R} (G(\mathbf{x}') - F(\mathbf{x}))^2 . \quad (14)$$

$p$  is maximal if the intensity value differences are minimal and vice versa. The scope of hypotheses verification is finding the transformation with higher probability. Let  $p_1 = p_R(V)$  and  $p_2 = p_R(Q)$ , it then follows

$$p_1 > p_2 \quad \Leftrightarrow \quad \log p_1 > \log p_2 \quad \Leftrightarrow \quad \log p_1 - \log p_2 > 0 . \quad (15)$$

Therefore hypothesis testing boils down to calculating the SSD-difference for the two transformation assumptions. The absolute distance from zero represents the reliability of the result.



**Fig. 2.** Difference between flow field from planar ground (no flow above horizon) and obstacle in 30 m distance (planar motion parallax) for different camera translation in depth. The brightness corresponds to the length of the flow difference vector. Clearly, the challenge to distinguish between an obstacle and the planar ground near the focus of expansion by relative image motion becomes visible (camera focal length 689 pixel).

In practice, vehicle translation is not solely restricted to translation in  $T_z$ . However, the motion parameters not included in the model can be compensated for the most part by estimating an extra region shift. Nevertheless, over larger time scales, hypothesis verification becomes more and more prone to errors due to lighting changes and the unmodelled motion.

Therefore, in the verification case, we restrict ourselves to time scales of 20 frames (in practice this corresponds to camera translations of more than 2 m). As indicated in Fig. 2 and by our experimental results, such a translation provides a sufficient difference between the two transformation assumptions and allows for reliable hypothesis testing.

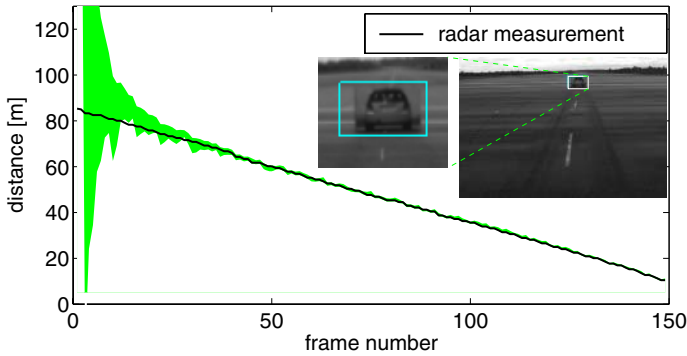
## 4 Experimental Results

The proposed algorithm has been tested on real roads. The results are given in the following.

**Comparison with distance from radar.** A textured wall with a corner reflector behind the wall represents the obstacle. Due to the breadboard construction the distance measurement from radar is taken as the reference value and compared to distance from depth tracking. The results in Fig. 3 show, that distance measurement by scale is error-prone around the initial frame. This is not surprising as the scale factor is close to 1 and therefore division by  $1 - s$  in (5) for distance computation leads to high inaccuracies. However, distance computation becomes quickly stable with greater vehicle translation. This clearly shows that distance estimation over large time scales is indispensable.

**Obstacle detection performance.** In the remaining part of this section we show three exemplary sequences from our test series on real roads to demonstrate hypotheses generation and testing. Figure 4 shows the first frame for each of these sequences. Notice that obstacle edges are present close to the focus of expansion what makes detection quite challenging.

The sequences are taken from a camera with 8.4 mm focal length (8.4 mm corresponds to 840 pixel) and 1.1 m camera height. The correlation threshold for replacing a depth tracker is set to 0.8. The threshold for hypothesis verification in the distance histogram is set to 70 m and restricted to the driving corridor.



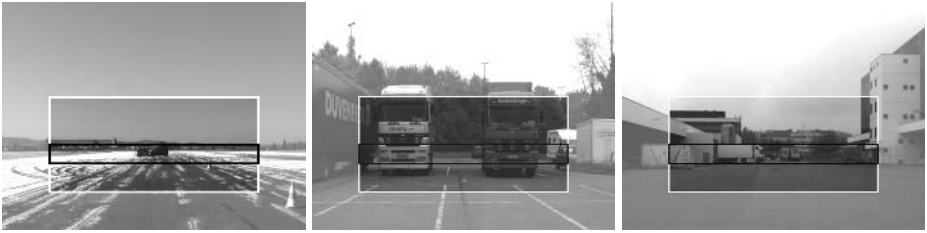
**Fig. 3. Distance from radar compared to distance from region scale.** Distance from scale plus and minus its standard deviation is represented by the gray area. The *thumbnail image* was taken at 50 m obstacle distance. The plot shows, that depth tracking allows to accurately estimate depth for distant obstacles.

These settings have been fixed in all three experiments showing the robustness of such parameters.

The first sequence shows an approach to a crash scene more than 100 m away. The vehicle speed is approximately 10 m/sec. The algorithm detects the stationary obstacle already at 69 m distance. Obstacle verification is error prone at such distances leading to a low value for the SSD difference. At 45 m distance (see Fig. 5) the obstacle is verified and horizontal obstacle boundaries are successfully detected such that a driver assistance system can safely evade this hazardous situation.

The middle set of images proves the reliable testing of obstacle hypotheses. The two trucks influence depth tracking and generate an obstacle hypothesis in the distance histogram for the free region amongst them (see Fig. 5 black line). Obstacle verification clearly rejects this hypothesis verifying a free corridor. As the vehicle approaches closer to the hazardous scene, the distance histogram adopts to the true observations picking up the bushes in the background as obstacles.

The right example deals with an obstacle boundary close to the focus of expansion. Note that the truck trailer has no texture making it hard for structure from motion algorithms to detect the vehicle in general. Nevertheless, the truck is detected and verified successfully at 67 m. Obstacle boundaries are close to ground truth. At such a large distance, the two trucks on the right influence hypothesis verification and lead to obstacle assumptions. Obviously the verification is correct but not for the given hypothesis distance. As the vehicle approaches the hazardous object in Fig. 5, the obstacle boundary is estimated precisely although it runs next to the focus of expansion. The image shows the truck still 50 m away. Experiments on several test sequences show, that robust object detection and verification can be reached with the proposed basic approach. Further quantitative studies on larger test data bases are the focus of ongoing research.



**Fig. 4. Initial frames.** The **white box** indicates the cropped image size shown in Fig. 5. The **black box** marks the area used for depth tracking.



**Fig. 5. Obstacle detection with distance histogram** (black, scale on the left) and **hypotheses verification** (white, logarithmic scale, obstacle verified if above dashed line). The images show, that robust obstacle detection and verification is reached.

## 5 Conclusions

We have presented an algorithm for static obstacle detection in monocular image sequences. The scene depth is estimated by the change of region scale in image space; obstacle hypotheses are generated if depth estimation falls below a fixed threshold. To verify these hypotheses we check whether the observed transformation in image space is more likely to be generated by a static object or by the flat ground.

We implemented the algorithm on a Pentium IV with 3.2GHz and achieved a framerate of 23 frames per second for the distance histogram calculation. The distance histogram and verification computation together run at approximately 13frames per second. To the authors’ knowledge, this is the fastest monocular motion–base obstacle detection algorithm in literature for obstacles close to the focus of expansion. The approach is easily applicable to other motion based distance measurements for obstacle detection and verification.

Further research will concentrate on speed gain. A wide range of algorithms in literature was proposed to speed up and stabilize tracking in image space. To name one possibility, pixel selection can be used to reduce computation time in region tracking. It is in the focus of ongoing studies to intelligently distribute the single regions used for depth tracking in image space. Although the described system works well in unknown environments we believe that optimizing the distribution and number of the tracked regions with respect to the currently observed scene will lead to even better results and less computation time. Moreover, we will investigate means to improve obstacle detection by method

of segmentation [4] and globally optimized optic flow estimation [2] forced into distinction of vertical and horizontal planes.

It also remains an open problem to detect moving obstacles in a monocular scenario. However, to pick up the threads given in the introduction, moving objects are well detected by common radar sensors therefore a sensor fusion combining measurements from an active radar and passive visual sensor is a promising field for further research.

## References

1. M. Bertozzi, A. Broggi, M. Cellario, A. Fascioli, P. Lombardi, and M. Porta. Artificial vision in road vehicles. In *Proceedings of the IEEE*, volume 90, pages 1258–1271, 2002.
2. T. Brox, A. Bruhn, N. Papenbergh, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In T. Pajdla and J. Matas, editors, *Proc. 8th European Conference on Computer Vision*, volume 3024 of *LNCS*, pages 25–36. Springer, May 2004.
3. A. Bruhn, J. Weickert, and C. Schnörr. Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods. *International Journal of Computer Vision*, 63(3):211–231, 2005.
4. D. Cremers and S. Soatto. Motion competition: A variational framework for piecewise parametric motion segmentation. *International Journal of Computer Vision*, 62(3):249–265, May 2005.
5. B. Deutsch, C. Gräßl, F. Bajramovic, and J. Denzler. A comparative evaluation of template and histogram based 2D tracking algorithms. In *DAGM-Symposium 2005*, pages 269–276, 2005.
6. W. Förstner. 10 pros and cons against performance characterization of vision algorithms. *Performance Characteristics of Vision Algorithms*, 1996.
7. U. Franke and A. Joos. Real-time stereo vision for urban traffic scene understanding. In *Proc. IEEE Conference on Intelligent Vehicles*, pages 273–278, 2000.
8. G. Hager and P. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1025–1039, 1998.
9. Q. Ke and T. Kanade. Transforming camera geometry to a virtual downward-looking camera: Robust ego-motion estimation and ground-layer detection. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 390–397, 2003.
10. H. Longuet-Higgins and K. Prazdny. The interpretation of a moving retinal image. In *Proceedings of the Royal Society of London*, volume 208 of *Series B, Biological Sciences*, pages 385–397, July 1980.
11. B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.
12. C. Rabe, C. Volmer, and U. Franke. Kalman filter based detection of obstacles and lane boundary. In *Autonome Mobile Systeme*, volume 19, pages 51–58, 2005.
13. F. Stein. Efficient computation of optical flow using the census transform. In *DAGM04*, pages 79–86, 2004.
14. Z. Sun, G. Bebis, and R. Miller. On-road vehicle detection using optical sensors: A review. In *IEEE International Conference on Intelligent Transportation Systems*, volume 6, pages 125 – 137, 2004.

# 3D Human Motion Sequences Synchronization Using Dense Matching Algorithm

Mikhail Mozerov<sup>1</sup>, Ignasi Rius<sup>1</sup>, Xavier Roca<sup>1</sup>, and Jordi González<sup>2</sup>

<sup>1</sup> Computer Vision Center and Departament d'Informàtica  
Universitat Autònoma de Barcelona, 08193 Cerdanyola, Spain  
mozerov@cvc.uab.es

<sup>2</sup> Institut de Robòtica i Informàtica Industrial (UPC-CSIC), Edifici U  
Parc Tecnològic de Barcelona. 08028, Spain

**Abstract.** This work solves the problem of synchronizing pre-recorded human motion sequences, which show different speeds and accelerations, by using a novel dense matching algorithm. The approach is based on the dynamic programming principle that allows finding an optimal solution very fast. Additionally, an optimal sequence is automatically selected from the input data set to be a time scale pattern for all other sequences. The synchronized motion sequences are used to learn a model of human motion for action recognition and full-body tracking purposes.

## 1 Introduction

Visual motion analysis of human motion remains one of the most challenging open problems from computer vision [4,10]. The number of related difficulties is wide ranging from shape and appearance changes, 2D-3D projection ambiguities and self and non-self occlusions among others. Many applications, such as action recognition or full-body 3D tracking, use high dimensional space models, and only a reduced number of the considered space components are directly observable from 2D images. As a result, incorporating *a priori* information on human motion into these applications is essential. Many action recognition and 3D body tracking works rely on proper models of human motion, which constrain the search space using a training data set of pre-recorded motions [3,6,8,9]. Consequently, it is highly desirable to extract useful information from the training set of motion. However, training sequences may be acquired under very different conditions, showing different durations, velocities and accelerations during the performance of a particular action. As a result, it is difficult to put in correspondence postures from different sequences of the same action in order to perform useful statistical analysis to the raw training data. Therefore, a method for synchronizing the whole training set is required so that we can establish a mapping between postures from different sequences. Ning et al. proposed a method for normalizing the length of cyclic walking sequences using a self-correlation measure [6]. As a result, the training walking cycles are rescaled to last the same period of time and are aligned to the same phase. Then, a walking motion model is learnt as Gaussian distributions per each joint, which include constraints on human motion. The model is used

to track a walking sequence of a 12 DOF body model using a particle filtering framework. However, unlike our approach, self-correlation is only suitable for cyclic motion sequences.

Similarly to our work, in [5] a variation of Dynamic Programming (DP) is used to match motion sequences acquired from a motion capture system. However, the overall approach is aimed at the optimization of a posterior key-frame search algorithm. Then, the output from this process is used for synthesizing realistic human motion by blending the training set. They divided the body in 4 portions, and similarities are evaluated independently for each part. In contrast, our approach synchronizes motion sequences considering the whole body in the matching process. We also use a representation based on relative joint angles which is more suitable for human motion representation.

The DP approach has been widely used in the literature for stereo matching and image processing applications [1,7]. Such applications often demand fast calculations in real-time, robustness against image discontinuities and unambiguous matching. Likewise, we present a dense matching algorithm based on DP, which is used to synchronize human motion sequences of the same action class in the presence of different speeds and accelerations. The algorithm finds an optimal solution in real-time. Additionally, we automatically select from the training data the best pattern for time synchronization following a minimum global distance criterion.

The synchronized version of the training set is utilized to learn an action-specific model of human motion. The observed variances from the synchronized postures of the training set are computed to determine which human postures can be feasible during the performance of a particular action. This knowledge is subsequently used in a particle filter tracking framework to prune those predictions which are not likely to be found in that action.

The remainder of this paper is organized as follows: Section 2 explains the principles of human action modeling. In Section 3 we introduce a new dense matching algorithm for human motion sequences synchronization. Experimental results with real 3D human motion data are presented and discussed in Section 4. Section 5 summarizes our conclusions.

## 2 Human Action Model

The motion sequences we want to synchronize have been acquired using a commercial Motion Capture system. A set of 19 reflective markers were placed on several characteristic points of the subject's body to obtain its absolute 3D positions. The body model employed is composed of twelve rigid body parts (hip, torso, shoulder, neck, two thighs, two legs, two arms and two forearms) and fifteen joints. These joints are structured in a hierarchical manner, constituting a kinematic tree, where the root is located at the hip. We use directional cosines to represent relative orientations of the limbs within the kinematic tree. The height of the pelvis is also modeled since it provides useful information for characterizing actions such as jumping or sitting. As a result, we represent a human body posture  $\psi$  using 37 parameters, i.e.

$$\psi = \{u, \theta_1^x, \theta_1^y, \theta_1^z, \dots, \theta_{12}^x, \theta_{12}^y, \theta_{12}^z\}, \quad (1)$$



where  $u$  is the normalized height of the pelvis, and  $\theta_l^x, \theta_l^y, \theta_l^z$  are the relative directional cosines for limb  $l$ , i.e. the cosine of the angle between a limb  $l$  and each axis  $x$ ,  $y$ , and  $z$  respectively. Directional cosines constitute a good representation method for body modeling, since it doesn't lead to discontinuities, in contrast to other methods such as Euler angles or spherical coordinates. Additionally, unlike quaternion, they have a direct geometric interpretation. However, such representation generates a considerable redundancy of the vector space components. Indeed, we are using 3 parameters to determine only 2 DOF for each limb.

Let us introduce a particular performance of an action. A performance  $\Psi_i$  consists of a time-ordered sequence of postures

$$\Psi_i = \{\psi_i^1, \dots, \psi_i^{F_i}\}, \tag{2}$$

where  $i$  is an index indicating the number of performance, and  $F_i$  is the total number of postures that constitute the performance  $\Psi_i$ . We assume that each two consecutive postures are separated by a time interval  $\delta f$ , which depends on the frame rate of the pre-recorded input sequences, thus the duration of a particular performance is  $T_i = \delta f F_i$ . Finally, an action  $A_k$  is defined by all the  $I_k$  performances that belong to that action  $A_k = \{\Psi_1, \dots, \Psi_{I_k}\}$ .

As we mentioned above, the original vector space is redundant. Additionally, the human body motion is intrinsically constrained, and these natural constraints lead to highly correlated data in the original space. Therefore, we aim to find a more compact representation of the original data to avoid redundancy. To do this, we consider a set of performances corresponding to a particular action  $A_k$ , and perform Principal Component Analysis to all the postures that belong to that action. Eventually, the following eigenvector decomposition equation has to be solved

$$\lambda_j \mathbf{e}_j = \Sigma_k \mathbf{e}_j, \tag{3}$$

where  $\Sigma_k$  stands for the  $37 \times 37$  covariance matrix calculated with all the postures of action  $A_k$ . As a result, each eigenvector  $\mathbf{e}_j$  corresponds to a mode of variation of human motion, and its corresponding eigenvalue  $\lambda_j$  is related to the variance specified by the eigenvector. In our case, each eigenvector reflects a natural mode of variation of human gait. To perform dimensionality reduction over the original data, we consider only the first  $b$  eigenvectors that span the new representation space for this action, hereafter *aSpace* [2]. We assume that the overall variance of a new space approximately equals to the overall variance of the unreduced space

$$\lambda_s = \sum_{j=1}^b \lambda_j \approx \sum_{j=1}^b \lambda_j + \varepsilon_b = \sum_{j=1}^{37} \lambda_j, \tag{4}$$

where  $\varepsilon_b$  is the *aSpace* approximation error.

Consequently, we use Eq. (4) to find the smallest number  $b$  of eigenvalues, which provide an appropriate approximation of the original data, and human postures are projected into the *aSpace* by

$$\tilde{\Psi} = [\mathbf{e}_1, \dots, \mathbf{e}_b]^T (\Psi - \bar{\Psi}), \tag{5}$$

where  $\Psi$  refers to the original posture,  $\tilde{\Psi}$  denotes the lower-dimensional version of the posture represented using the *aSpace*,  $[\mathbf{e}_1, \dots, \mathbf{e}_b]$  is the *aSpace* transformation matrix that correspond to the first  $b$  selected eigenvectors, and  $\bar{\Psi}$  is the posture mean value that is formed by averaging all postures, which are assumed to be transformed into the *aSpace*. As a result, we obtain a lower-dimensional representation of human postures more suitable to describe human motion since we found that each dimension on the *aSpace* describes a natural mode of variation of human motion.

The projection of the training sequences into the *aSpace* will constitute the input for our sequence synchronization algorithm. Hereafter, we consider a multidimensional signal  $\mathbf{x}_i(t)$  as an interpolated expansion of each training sequence  $\tilde{\Psi}_i = \{\tilde{\Psi}_i^1, \dots, \tilde{\Psi}_i^{F_i}\}$  such as

$$\tilde{\Psi}_i^f = \mathbf{x}_i(t) \text{ if } t = (f - 1)\delta f; f = 1, \dots, F_i; \tag{6}$$

where the time domain of each action performance  $\mathbf{x}_i(t)$  is  $[0, T_i)$ .

### 3 Synchronization Algorithm

Let us assume that any two considered signals correspond to the identical action, but one runs faster than another (e.g. Fig. 1. (a)). Under the assumption that the rates ratio of the compared actions is a constant, the two signals might be easily linearly synchronized in the following way

$$\mathbf{x}_n(t) \approx \mathbf{x}_{n,m}(t) = \mathbf{x}_m(\alpha t); \quad \alpha = \frac{T_m}{T_n}; \tag{7}$$

where  $\mathbf{x}_n$  and  $\mathbf{x}_m$  are the two compared multidimensional signals,  $T_n$  and  $T_m$  are the periods of the action performances  $n$  and  $m$ ,  $\tilde{\mathbf{x}}_{n,m}$  is linearly normalized version of  $\mathbf{x}_m$  hence  $T_n = T_{m,n}$ .

Unfortunately, in our research we rarely if ever have a constant rate ratio  $\alpha$ . An example, which is illustrated in Fig. 1. (b), shows that a simple normalization using Eq. (7) does not give us the needed signal fitting, and a nonlinear data synchronization method is needed. Further in the text we shall assume that the linear synchronization is done and all the periods  $T_n$  possess the same value  $T$ .

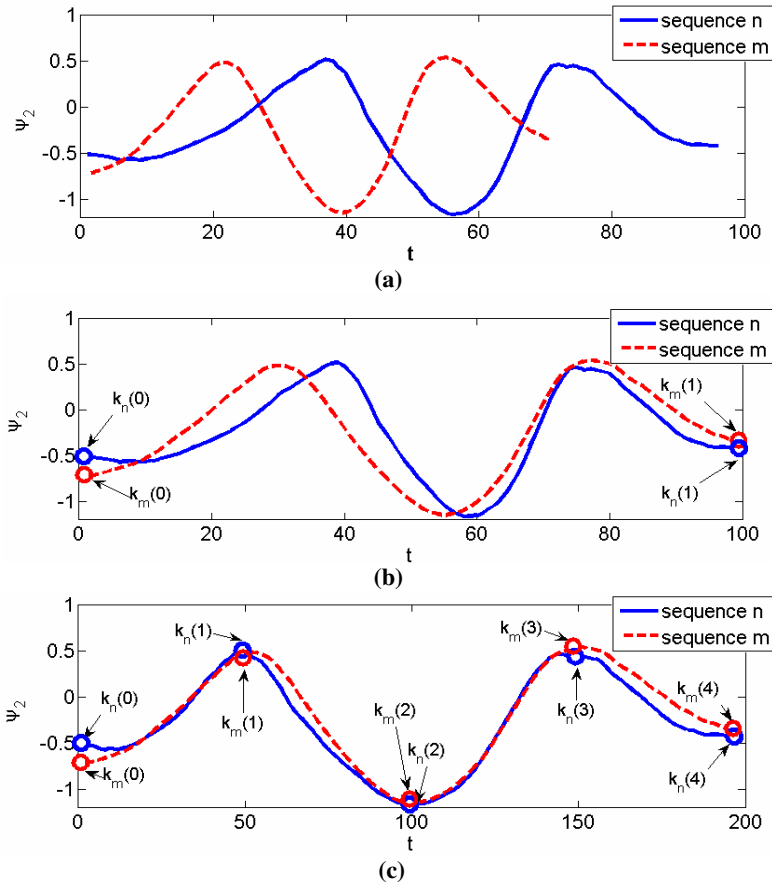
The nonlinear data synchronization should be done by

$$\mathbf{x}_n(t) \approx \mathbf{x}_{n,m}(t) = \mathbf{x}_m(\tau(t)); \quad \tau(t) = \int_0^t \alpha(t) dt; \tag{8}$$

where  $\mathbf{x}_{n,m}(t)$  is the best synchronized version of the action  $\mathbf{x}_m(t)$  to the action  $\mathbf{x}_n(t)$ . In the literature the function  $\tau(t)$  is usually referred to as the distance-time function. It is not an apt turn of phrase indeed, and we suggest naming it as the rate-to-rate synchronization function instead.

The rate-to-rate synchronization function  $\tau(t)$  satisfies several useful constraints, that are

$$\tau(0)=0; \quad \tau(T)=T; \quad \tau(t_k) \geq \tau(t_l) \text{ if } t_k > t_l. \tag{9}$$



**Fig. 1.** (a) Non synchronized one-dimensional sequences. (b) Linearly synchronized sequences. (c) Synchronized sequences using a set of key-frames.

One common approach for building the function  $\tau(t)$  is based on a key-frame model. This model assumes that the compared signals  $\mathbf{x}_n$  and  $\mathbf{x}_m$  have similar sets of singular points, that are  $\{t_n(0), \dots, t_n(p), \dots, t_n(P-1)\}$  and  $\{t_m(0), \dots, t_m(p), \dots, t_m(P-1)\}$  with the matching condition  $t_n(p) = t_m(p)$ . The aim is to detect and match these singular points, thus the signals  $\mathbf{x}_n$  and  $\mathbf{x}_m$  are synchronized. However, the singularity detection is an intricate problem itself, and to avoid the singularity detection stage we propose a dense matching. In this case a time interval  $t_n(p+1) - t_n(p)$  is constant, and in general  $t_n(p) \neq t_m(p)$ .

The function  $\tau(t)$  can be represented as  $\tau(t) = t(1 + \Delta_{n,m}(t))$ . In this case, the sought function  $\Delta_{n,m}(t)$  might synchronize two signals  $\mathbf{x}_n$  and  $\mathbf{x}_m$  by

$$\mathbf{x}_n(t) \approx \mathbf{x}_m(t + \Delta_{n,m}(t)t); \tag{10}$$

Let us introduce a formal measure of synchronization of two signals by

$$D_{n,m} = \int_0^T \|\mathbf{x}_n(t) - \mathbf{x}_m(t + \Delta_{n,m}(t))\| dt + \mu \int_0^T \left\| \frac{d\Delta_{n,m}(t)}{dt} \right\| dt. \tag{11}$$

where  $\|\bullet\|$  denotes one of possible vector distances,  $D_{n,m}$  is referred to as the synchronization distance that consists of two parts, where the first integral represents the functional distance between the two signals, and the second integral is a regularization term, which expresses desirable smoothness constraints of the solution. The proposed distance function is simple and makes intuitive sense. It is natural to assume that the compared signals are synchronized better when the synchronization distance between them is minimal. Thus, the sought function  $\Delta_{n,m}(t)$  should minimize the synchronization distance between matched signals.

In the case of a discrete time representation, Eq.(11) can be rewritten as

$$D_{n,m} = \sum_{i=0}^{<P} \left| \mathbf{x}_n(i\delta t) - \mathbf{x}_m(i\delta t + \Delta_{n,m}(i)\delta t) \right|^2 + \mu \sum_{i=0}^{<P-1} \left| \Delta_{n,m}(i+1)\delta t - \Delta_{n,m}(i) \right|, \tag{12}$$

where  $\delta t$  is a time sampling interval. Eq. (9) implies

$$\left| \Delta_{n,m}(p+1) - \Delta_{n,m}(p) \right| \leq 1, \tag{13}$$

where index  $p = \{0, \dots, P-1\}$  satisfies  $\delta t P = T$ .

The synchronization problem is similar to the matching problem of two epipolar lines in a stereo image. In the case of the stereo image processing the parameter  $\Delta(t)$  is called disparity. For stereo matching a disparity space image (DSI) representation is used [1,7]. The DSI approach assumes that 2D DSI matrix has dimensions time  $0 \leq p < P$ , and disparity  $-D \leq d \leq D$ . Let  $E(d, p)$  denote the DSI cost value assigned to matrix element  $(d, p)$  and calculated by

$$E_{n,m}(p, d) = \left| \mathbf{x}_n(p\delta t) - \mathbf{x}_m(p\delta t + d\delta t) \right|^2. \tag{14}$$

Now we formulate an optimization problem as follows: find the time-disparity function  $\Delta_{n,m}(p)$ , which minimizes the synchronization distance between the compared signals  $\mathbf{x}_n$  and  $\mathbf{x}_m$  i.e.

$$\Delta_{n,m}(p) = \arg \min_d \sum_{i=0}^{<P} E_{n,m}(i, d(i)) + \mu \sum_{i=0}^{<P-1} |d(i+1) - d(i)|. \tag{15}$$

The discrete function  $\Delta(p)$  coincides with the optimal path through the DSI trellis as it is shown in Fig. 2. Here term “optimal” means that the sum of the cost values along this path plus the weighted length of the path is minimal among all other possible paths.

The optimal path problem can be easily solved by using the method of dynamic programming. The method consists of step-by-step control and optimization that is given by a recurrence relation

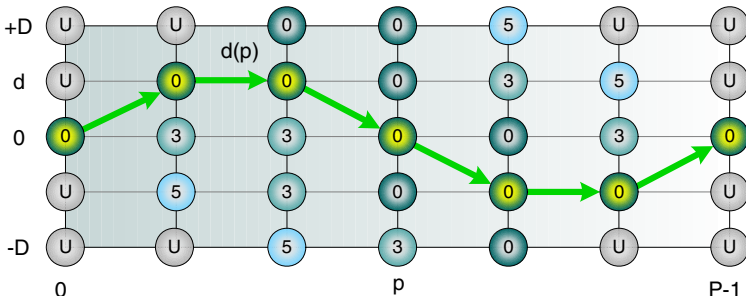


Fig. 2. The optimal path trough the DSI trellis

$$S(p, d) = E(p, d) + \min_{k \in \{0, \pm 1\}} \{S(p-1, d+k) + \mu|d+k|\}, \tag{16}$$

$$S(0, d) = E(0, d),$$

where the scope of the minimization parameter  $k \in \{0, \pm 1\}$  is chosen in accordance with Eq. (13). By using the recurrence relation the minimal value of the objective function in Eq.(15) can be found at the last step of optimization. Next, the algorithm works in reverse order and recovers a sequence of optimal steps (using the lookup table  $K(p, d)$  of the stored values of the index  $k$  in the recurrence relation (16)) and eventually the optimal path by

$$d(p-1) = d(p) + K(p, d(p)),$$

$$d(P-1) = 0, \tag{17}$$

$$\Delta(p) = d(p).$$

Now the synchronized version of  $\mathbf{x}_m(t)$  might be easily calculated by

$$\mathbf{x}_{n,m}(p\delta t) = \mathbf{x}_m(p\delta t + \Delta_{n,m}(p)\delta t). \tag{18}$$

Here we assume that  $n$  is the number of the base rate sequences and  $m$  is the number of sequences to be synchronized.

The dense matching algorithm that synchronize two arbitrary  $\mathbf{x}_n(t)$  and  $\mathbf{x}_m(t)$  pre-recorded human motion sequences  $\mathbf{x}_n(t)$  and  $\mathbf{x}_m(t)$  is now summarized as follows:

- Prepare a 2D DSI matrix, and set initial cost values  $E_0$  using Eq. (14).
- Find the optimal path trough the DSI using recurrence Eqs. (16-17).
- Synchronize  $\mathbf{x}_m(t)$  to the rate of  $\mathbf{x}_n(t)$  using Eq.(18).

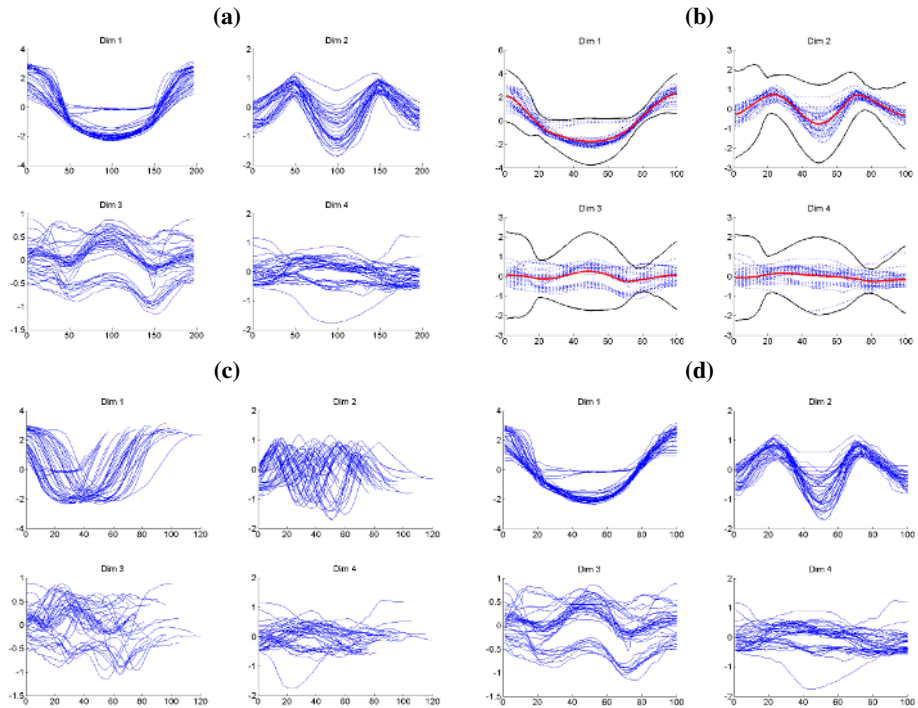
Our algorithm assumes that a particular sequence is chosen to be a time scale pattern for all other sequences. It is obvious that an arbitrary choice among the training set is not a reasonable solution, and now we aim to find a statistically proven rule that is able to make an optimal choice according to some appropriate criterion. Note that each synchronized pair of sequences  $(n, m)$  has its own synchronization distance calculated by Eq. (12). Then the full synchronization of all the sequences relative to the pattern sequences  $n$  has its own global distance

$$C_n = \sum_{m \in A_k} C_{n,m}. \quad (19)$$

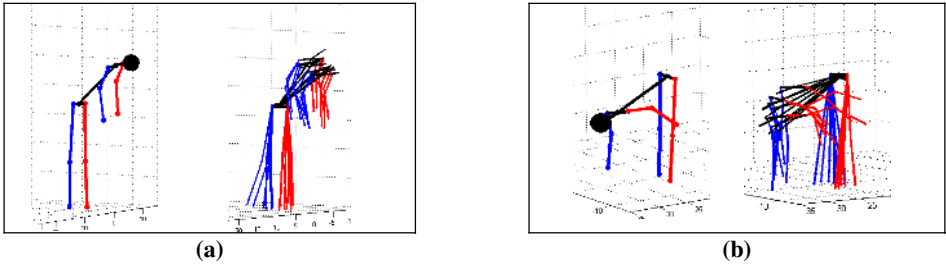
We propose to choose the synchronizing pattern sequence with minimal global distance. In statistical sense such signal can be considered as a median value over all the performances that belong to the set of  $A_k$  or can be referred to as “median” sequence.

## 4 Computer Experiments

The synchronization method has been tested with a training set consisting of 40 performances of a bending action. To build the *aSpace* representation, we choose the first 16 eigenvectors that captured 95% of the original data. The first 4 dimensions within the *aSpace* of the training sequences are illustrated in Fig.3.(a). All the performances have different durations with 100 frames on average. The observed initial data shows different durations, speeds and accelerations between the sequences. Such a mistiming makes very difficult to learn any common pattern from the data. The proposed synchronization algorithm was coded in C++ and run with a 3 GHz Pentium D processor. The time needed for synchronizing two arbitrary sequences taken from our



**Fig. 3.** (a) Non-synchronized training set. (b) Automatically-synchronized training set with the proposed approach. (c) Manually-synchronized training set with key-frames. (d) Learnt motion model for the bending action.



**Fig. 4.** (a) and (b) Mean learnt postures from the action corresponding to frames 10 and 40 (left). Sampled postures using the learnt corresponding variances (right).

database is  $1.5 \cdot 10^{-2}$  seconds and 0.6 seconds to synchronize the whole training set, which is illustrated in Fig.3.(b). To prove the correctness of our approach, we manually synchronized the same training set by selecting a set of 5 key-frames in each sequence by hand following a maximum curvature subjective criterion. Then, the training set was resampled so each sequence had the same number of frames between each key-frame. In Fig.3.(c), the first 4 dimensions within the *aSpace* of the resulting manually synchronized sequences are shown. We might observe that the results are very similar to the ones obtained with the proposed automatic synchronization method. The synchronized training set from Fig.3.(b) has been used to learn an action-specific model of human motion for the bending action. The model learns a mean-performance for the synchronized training set, and its observed variance at each posture. In Fig.3.(d) the learnt action model for the bending action is plotted. The mean-performance corresponds to the solid red line while the black solid line depicts  $\pm 3$  times the learnt standard deviation at each synchronized posture. The input training sequence set is depicted as dashed blue lines.

This motion model can be used in a particle filter framework as *a priori* knowledge on human motion. The learnt model would predict for the next time step only those postures which are feasible during the performance of a particular action. In other words, only those human postures which lie within the learnt variance boundaries from the mean performance are accepted by the motion model. In Fig.4 we show two postures corresponding to frames 10 and 40 from the learnt mean performance, and a random set of accepted postures by the action model. We might observe that for each selected mean posture, only similar and meaningful postures are generated.

## 5 Conclusion

In this paper, a novel dense matching algorithm for human motion sequences synchronization has been proposed. The technique utilizes dynamic programming, and can be used in real-time applications. We also introduce the definition of the median sequence that is used to choose a time scale pattern for all other sequences. The synchronized motion sequences are utilized to learn a model of human motion and to extract signal statistics.

## Acknowledgements

This work has been supported by EC grant IST-027110 for the HERMES project and by the Spanish MEC under projects TIC2003-08865 and DPI-2004-5414. M. Mozerov acknowledges the support of the Ramon y Cajal research program, MEC, Spain, and J. González also acknowledges the support of a Juan de la Cierva Postdoctoral fellowship from the Spanish MEC.

## References

1. Brown, M. Z., Burschka, D., and Hager, G. D.: Advances in computational stereo. *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 25 (2003) 993–1008
2. González, J., Varona, J., Roca, X., and Villanueva, J.J.: Analysis of human walking based on aSpaces. *Lecture Notes in Computer Science*, Vol. 3179. Springer-Verlag, Berlin Heidelberg New York (2004) 177–188
3. Grochow, K., Martin, S.L., Hertzmann, A., and Popovic, Z.: Style-based inverse kinematics. *ACM Transactions on Graphics (Proc. of SIGGRAPH 2004)*, Vol. 23 (2004) 522–531
4. Moeslund, T.B., and Granum, E.: A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding*, Vol. 81 (2001) 231–268
5. Nakazawa, A., Nakaoka, S., and Ikeuchi, K.: Matching and blending human motions using temporal scaleable dynamic programming. *Proc. of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, (2004) 287–294
6. Ning, H., Tan, T., Wang, L., and Hu, W.: Kinematics-based tracking of human walking in monocular video sequences. *Image and Vision Computing*, Vol. 22 (2004) 429–441
7. Scharstein, D., and Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, Vol. 47 (2002) 7–42
8. Sidenbladh, H., Black, M.J., and Sigal, L.: Implicit probabilistic models of human motion for synthesis and tracking. *Lecture Notes in Computer Science*, Vol. 2350. Springer-Verlag, Berlin Heidelberg New York (2002) 784–800
9. Urtasun, R., Fleet, D.J., Hertzmann, A. and Fua, P.: Priors for people tracking from small training sets. *Proc. IEEE International Conference on Computer Vision (ICCV05)*, Vol. 1 (2005) 403–410
10. Wang, L., Hu, W., and Tan, T.: Recent developments in human motion analysis. *Pattern Recognition*, Vol. 36 (2003) 585–601



# Cloth X-Ray: MoCap of People Wearing Textiles<sup>\*</sup>

Bodo Rosenhahn<sup>1</sup>, Uwe G. Kersting<sup>2</sup>, Katie Powell<sup>2</sup>, and Hans-Peter Seidel<sup>1</sup>

<sup>1</sup> Max Planck Institute for Informatics, Stuhlsatzhausenweg 85,  
D-66123 Saarbrücken, Germany  
rosenhahn@mpi-inf.mpg.de

<sup>2</sup> Department of Sport and Exercise Science  
The University of Auckland, New Zealand

**Abstract.** The contribution presents an approach for motion capturing (MoCap) of dressed people. A cloth draping method is embedded in a silhouette based MoCap system and an error functional is formalized to minimize image errors with respect to silhouettes, pose and kinematic chain parameters, the cloth draping components and external wind forces. We report on various experiments with two types of clothes, namely a skirt and a pair of shorts. Finally we compare the angles of the MoCap system with results from a commercially available marker based tracking system. The experiments show, that we are basically within the error range of marker based tracking systems, though body parts are occluded with cloth.

## 1 Introduction

Marker-less motion capturing is a highly challenging topic of research and many promising approaches exist to tackle the problem [12,5,1,10,4,7]. In most setups it is required that the subjects have to wear either body suits, to be naked or at least to wear clothing which stresses the body contours (e.g. swim suits). Such clothing is often uncomfortable to wear in contrast to loose clothing (shirts or shorts). The analysis of outdoor sport events also requires to take clothing into account. On the other hand, cloth draping is a well established field of research in computer graphics and virtual clothing can be moved and rendered so that it blends seamlessly with motion and appearance in movie scenes [6,8,9,17]. Existing approaches can be roughly divided in geometrically or physically based ones. Physical approaches model cloth behavior by using potential and kinetic energies. The cloth itself is often represented as a particle grid in a spring-mass scheme or by using finite elements [9]. Geometric approaches [17] model clothes by using other mechanics theories which are often determined empirically. These methods can be very fast computationally but are often criticized as being visually unappealing.

The motivation of this work is to combine a cloth draping algorithm with a marker-less MoCap system. The key idea is to use the appearance of the cloth and the visible parts of the human being to determine the underlying kinematic structure, though it might be heavily occluded.

---

<sup>\*</sup> We gratefully acknowledge funding by the Max-Planck Center for visual computing and communication.

## 2 Foundations: Silhouette Based MoCap

This work is an extension of a previously developed marker-less MoCap system [16]. In this system, the human being is represented in terms of free-form surface patches, joint indices are added to each surface node and the joint positions are assumed. This allows to generate arbitrary body configurations, steered through joint angles. The assumed corresponding counterparts in the images are 2D silhouettes: These are used to reconstruct 3D ray bundles and a spatial distance constraint is minimized to determine the position and orientation of the surface mesh and the joint angles. In this section we will give a brief summary of the MoCap system. These foundations are needed later to explain concisely, where and how the cloth draping approach is incorporated.

### 2.1 Silhouette Extraction

Image segmentation usually means to estimate boundaries of objects in an image. It is an important step for data abstraction, but the task can become very difficult due to noise, shading, occlusion or texture transitions between the object and the background. Our approach is based on image segmentation based on level sets [3,14,2].



**Fig. 1.** Silhouette extraction based on level set functions. Left: Initial segmentation. Right: Segmentation result.

A level set function  $\Phi \in \Omega \mapsto \mathbb{R}$  splits the image domain  $\Omega$  into two regions  $\Omega_1$  and  $\Omega_2$  with  $\Phi(x) > 0$  if  $x \in \Omega_1$  and  $\Phi(x) < 0$  if  $x \in \Omega_2$ . The zero-level line thus marks the boundary between both regions. On a discrete image, the level set functions are modeled through a distance transform from the contour line to the inner and outer region with negative and positive distance values, respectively. Both regions are analyzed with respect to the probabilities of image features (e.g. gray value distributions, color or texture channels). Now the key idea is to evolve the contour line, to maximize the probability density functions with respect to each other. Furthermore, the boundary between both regions should be as small as possible. This can be expressed by adding a

smoothness term. Both parts lead to the following energy functional that is sought to be minimized:

$$E(\Phi, p_1, p_2) = - \int_{\Omega} (H(\Phi(x)) \log p_1 + (1 - H(\Phi(x))) \log p_2 + v |\nabla H(\Phi(x))|) dx$$

where  $v > 0$  is a weighting parameter and  $H(s)$  is a regularized version of the Heaviside function, e.g. the error function. The probability densities  $p_i$  are estimated according to the *expectation-maximization principle*. Having the level set function initialized with some contour, the probability densities within the two regions are estimated by the gray value histograms smoothed with a Gaussian kernel  $K_{\sigma}$  and its standard deviation  $\sigma$ . Figure 1 shows on the left an example image with an initialization of the region as rectangle. The right image shows the estimated (stationary) contour after 50 iterations. As can be seen, the legs and the skirt are well extracted, but there are some deviations in the feet region, due to shadows. Such inaccuracies can be compensated through the pose estimation procedure.

### 2.2 Registration, Pose Estimation

Assuming an extracted image contour and the silhouette of the projected surface mesh, the closest point correspondences between both contours are used to define a set of corresponding 3D lines and 3D points. Then a 3D point-line based pose estimation algorithm for kinematic chains is applied to minimize the spatial distance between both contours: For point based pose estimation each line is modeled as a 3D Plücker line  $L_i = (n_i, m_i)$ , with a (unit) direction  $n_i$  and moment  $m_i$  [13]. The 3D rigid motion is expressed as exponential form

$$M = \exp(\theta \hat{\xi}) = \exp \begin{pmatrix} \hat{\omega} & v \\ 0_{3 \times 1} & 0 \end{pmatrix} \tag{1}$$

where  $\theta \hat{\xi}$  is the matrix representation of a twist  $\xi \in se(3) = \{(v, \hat{\omega}) | v \in \mathbb{R}^3, \hat{\omega} \in so(3)\}$ , with  $so(3) = \{A \in \mathbb{R}^{3 \times 3} | A = -A^T\}$ . The Lie algebra  $so(3)$  is the tangential space of the 3D rotations. Its elements are (scaled) rotation axes, which can either be represented as a 3D vector or screw symmetric matrix,

$$\theta \omega = \theta \begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix}, \text{ with } \|\omega\|_2 = 1 \quad \text{or} \quad \theta \hat{\omega} = \theta \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix}. \tag{2}$$

A twist  $\xi$  contains six parameters and can be scaled to  $\theta \xi$  for a unit vector  $\omega$ . The parameter  $\theta \in \mathbb{R}$  corresponds to the motion velocity (i.e., the rotation velocity and pitch). For varying  $\theta$ , the motion can be identified as screw motion around an axis in space. The six twist components can either be represented as a 6D vector or as a  $4 \times 4$  matrix,

$$\theta \xi = \theta (\omega_1, \omega_2, \omega_3, v_1, v_2, v_3)^T, \|\omega\|_2 = 1, \quad \theta \hat{\xi} = \theta \begin{pmatrix} 0 & -\omega_3 & \omega_2 & v_1 \\ \omega_3 & 0 & -\omega_1 & v_2 \\ -\omega_2 & \omega_1 & 0 & v_3 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \tag{3}$$

To reconstruct a group action  $M \in SE(3)$  from a given twist, the exponential function  $\exp(\theta \hat{\xi}) = \sum_{k=0}^{\infty} \frac{(\theta \hat{\xi})^k}{k!} = M \in SE(3)$  must be computed. This can be done efficiently by using the Rodriguez formula [13].

For pose estimation the reconstructed Plücker lines are combined with the screw representation for rigid motions: Incidence of the transformed 3D point  $X_i$  with the 3D ray  $L_i = (n_i, m_i)$  can be expressed as

$$(\exp(\theta \hat{\xi})X_i)_{3 \times 1} \times n_i - m_i = 0. \tag{4}$$

Since  $\exp(\theta \hat{\xi})X_i$  is a 4D vector, the homogeneous component (which is 1) is neglected to evaluate the cross product with  $n_i$ . Then the equation is linearized and iterated, see [16].

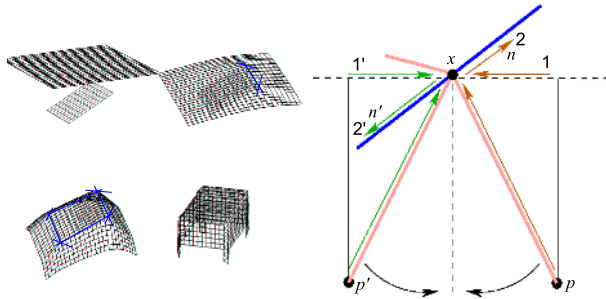
Joints are expressed as special screws with no pitch of the form  $\theta_j \hat{\xi}_j$  with known  $\hat{\xi}_j$  (the location of the rotation axes is part of the model) and unknown joint angle  $\theta_j$ . The constraint equation of an  $i$ th point on a  $j$ th joint has the form

$$(\exp(\theta_j \hat{\xi}_j) \dots \exp(\theta_1 \hat{\xi}_1) \exp(\theta \hat{\xi})X_i)_{3 \times 1} \times n_i - m_i = 0 \tag{5}$$

which is linearized in the same way as the rigid body motion itself. It leads to three linear equations with the six unknown pose parameters and  $j$  unknown joint angles.

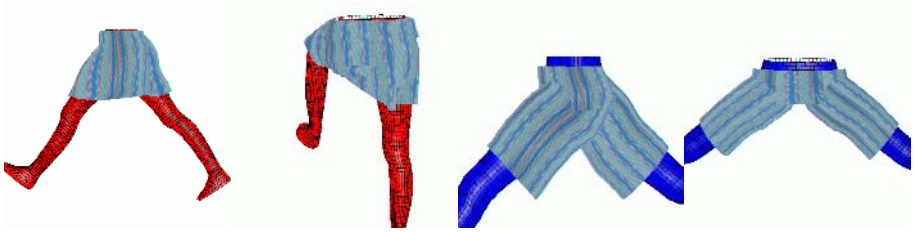
### 3 Kinematic Cloth Draping

For our set-up we decided to use a geometric approach to model cloth behavior. The main reason is that cloth draping is needed in one of the innermost loops for pose estimation and segmentation. Therefore it must be very fast. In our case we need around 400 iterations for each frame to converge to a solution. A cloth draping algorithm in the area of seconds would require hours to calculate the pose of one frame and weeks for a whole sequence. We decided to model the skirt as a string-system with underlined

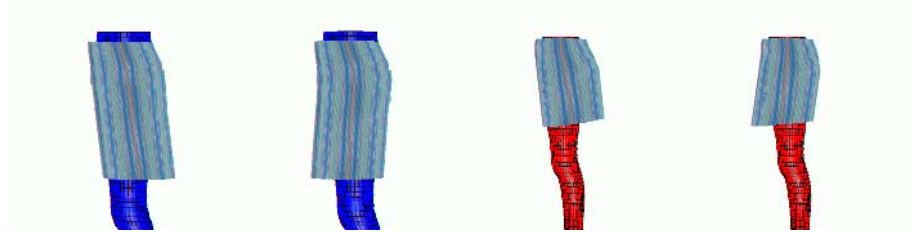


**Fig. 2.** The cloth draping principle. Joints are used to deform the cloth while draping on the surface mesh.

kinematic chains: The main principle is visualized on the left in Figure 2 for a piece of cloth falling on a plane. The piece of cloth is represented as a particle grid, a set of points with known topology. While lowering the cloth, the distance of each cloth point to the ground plane is determined. If the distance between one point on the cloth to the surface is below a threshold, the point is set as a fixed-point, see the top right image on the left of Figure 2. Now the remaining points are not allowed to *fall* downwards any more.



**Fig. 3.** Cloth draping of a skirt and shorts in a simulation environment



**Fig. 4.** Wind model on the shorts (left) and the skirt (right). Visualized is frontal and backward wind.

Instead, for each point, the nearest fixed-point is determined and a joint (perpendicular to the particle point) is used to rotate the free point along the joint axis through the fixed point. The used joint axes are marked as blue lines in Figure 2. The image on the right in Figure 2 shows the geometric principle to determine the twist for rotation around a fixed point: The blue line represents a mesh of the rigid body,  $x$  is the fixed point and the (right) pink line segment connects  $x$  to a particle  $p$  of the cloth. The direction between both points is projected onto the  $y$ -plane of the fixed point (1). The direction is then rotated around 90 degrees (2), leading to the rotation axis  $n$ . The point pair  $(n, x \times n)$  are the components of the twist, see equation (3). While lowering the cloth, free particles not touching a second rigid point, will swing below the fixed point (e.g.  $p'$ ). This leads to an opposite rotation (indicated with  $(1')$ ,  $(2')$  and  $n'$ ) and the particle swings back again, resulting in a natural swinging draping pattern. The draping velocity is steered through a rotation velocity  $\theta$ , which is set to 2 degrees during iteration. Since all points either become fixed points, or result in a stationary configuration while swinging backwards and forwards, we constantly use 50 iterations to drape the cloth. The remaining images on the left in Figure 2 show the ongoing draping and the final result.

Figure 3 shows example images of a skirt and a pair of shorts falling on the leg model. The skirt is modeled as a 2-parametric mesh model. Due to the use of general rotations, the internal distances in the particle mesh cannot change with respect to one of these dimensions, since a rotation maintains the distance between the involved points. However, this is not the case for the second sampling dimension. For this reason, the skirt needs to be re-constrained after draping. If a stretching parameter is exceeded, the particles are re-constrained to minimal distance to each other. This is only done for the non-fixed points (i.e. for those which are not touching the skin). It results in a

better appearance, especially for certain leg configurations. Figure 3 shows that even the creases are maintained. In this case, shorts are simpler since they are modeled as cylinders, transformed together with the legs and then draped.

To improve the dynamic behavior of clothing during movements, we also add a wind-model to the cloth draping. We continue with the cloth-draping in the following way: dependent on the direction of wind we determine a joint on the nearest fixed point for each free point on the surface mesh with the joint direction being perpendicular to the wind direction. Now we rotate the free point around this axis dependent on the wind force (expressed as an angle) or until the cloth is touching the underlying surface. Figure 4 shows examples of the shorts and skirt with frontal or backward wind. The wind force and direction are later part of the minimization function during pose tracking. Since the motion dynamics of the cloth are determined dynamically, we need no information about the cloth type or weight since they are implicitly determined from the minimized cloth dynamics in the image data; we only need the measurements of the cloth.

### 4 Combined Cloth Draping and MoCap

The assumptions are as follows: We assume the representation of a subject’s lower torso (i.e. for the hip and legs) in terms of free-form surface patches. We also assume known joint positions along the legs. Furthermore we assume the wearing of a skirt

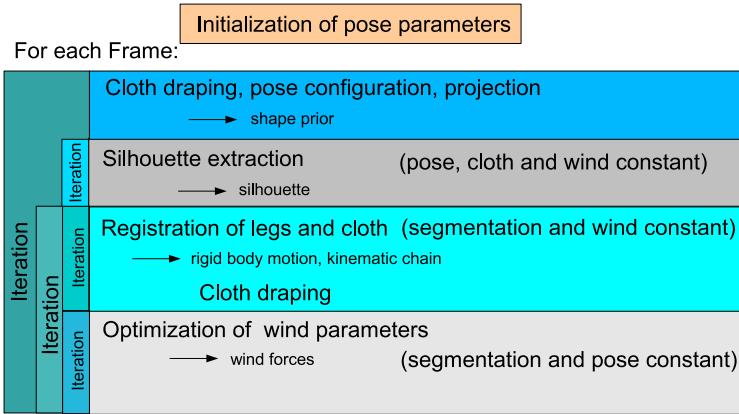


Fig. 5. The basic algorithm for combined cloth draping and motion capturing

or shorts with known measures. The person is walking or stepping in a four-camera setup. These cameras are triggered and calibrated with respect to one world coordinate system. The task is to determine the pose of the model and the joint configuration. For this we minimize the image error between the projected surface meshes to the extracted image silhouettes. The unknowns are the pose, kinematic chain and the cloth parameters (wind forces, cloth thickness, etc.). The task can be represented as an error functional as follows:

$$E(\Phi, p_1, p_2, \theta\xi, \theta_1, \dots, \theta_n, c, w) = - \underbrace{\int_{\Omega} (H(\Phi) \log p_1 + (1 - H(\Phi)) \log p_2 + v|\nabla H(\Phi)|) dx}_{\text{segmentation}} + \lambda \underbrace{\int_{\Omega} (\Phi - \Phi_0(\underbrace{\theta\xi, \theta_1, \dots, \theta_n}_{\text{pose and kinematic chain}}, \underbrace{c, w}_{\text{wind parameters}})) dx}_{\text{shape error}}$$

Due to the large number of parameters and unknowns we decided for an iterative minimization scheme, see Figure 5: Firstly, the pose, kinematic chain and wind parameters are kept constant, while the error functional for the segmentation (based on  $\Phi, p_1, p_2$ ) is minimized (section 2.1). Then the segmentation and wind parameters are kept constant while the pose and kinematic chain are determined to fit the surface mesh and the cloth to the silhouettes (section 2.2). Finally, different wind directions and wind forces are sampled to refine the pose result (section 3). Since all parameters influence each other, the process is iterated until a steady state is reached. In our experiments, we always converged to a local minimum.

## 5 Experiments

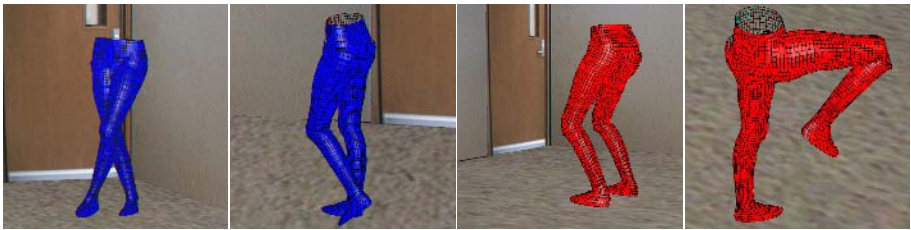
For the experiments we used a four-camera set up and grabbed image sequences of the lower torso with different motion patterns: The subject was asked to wear the skirt and the shorts while performing walking, leg crossing and turning, knee bending and walking with knees pulled up. We decided on these different patterns, since they are not only of importance for medical studies (e.g. walking), but they are also challenging



**Fig. 6.** Example sequences for tracking clothed people. **Top row:** walking, leg crossing, knee bending and knee pulling with a skirt. **Bottom row:** walking, leg crossing, knee bending and knee pulling with shorts. The pose is determined from 4 views (just one of the views is shown, images are cropped).



**Fig. 7.** Error during grabbing the images



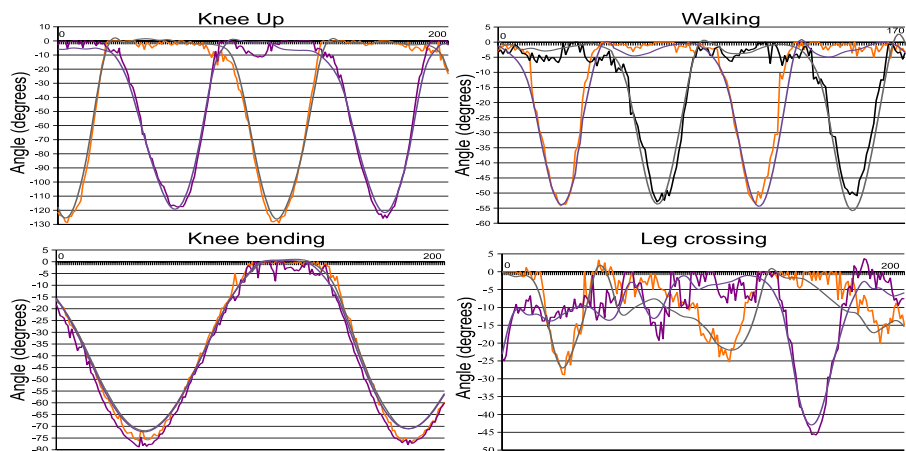
**Fig. 8.** Example leg configurations of the sequences. The examples are taken from the subject wearing the shorts (blue) and the skirt (red) (leg crossing, walking, knee bending, knee pulling).

for the cloth simulator, since the cloth is partially stretched (knee pulling sequence) or hanging down loosely (knee bending). The turning and leg crossing sequence is interesting due to the higher occlusions. Figure 6 shows some pose examples for the subject wearing the skirt (top) and shorts (bottom). The pose is visualized by overlaying the projected surface mesh onto the images. Just one of the four cameras is shown. Each sequence consists of 150-240 frames. Figure 7 visualizes the stability of our approach: While grabbing the images, a couple of frames were stored completely wrong. These sporadic outliers can be compensated from our algorithm, and a few frames later (see the image on the right) the pose is correct. Figure 8 shows leg configurations in a virtual environment. The position of the body and the joints reveal a natural configuration.

Finally, the question about the stability arises. To answer this question, we attached markers to the subject and tracked the sequences simultaneously with the commercially available Motion Analysis system [11]. The markers are attached to the visible parts of the leg and are not disturbed by the cloth. We then compare joint angles for different sequences with the results of the marker based system, similar to [16]. The overall errors for both types of cloth varies between 1.5 and 4.5 degrees, which indicates a stable result.

The diagrams in Figure 9 shows the overlay of the knee angles for two skirt and two shorts sequences. Due to space limits, we just show four sequences, the remaining four





**Fig. 9.** **Left:** Knee angles from sequences wearing the shorts. **Right:** Knee angles from sequences wearing the skirt. **Top left:** Angles of the knee up sequence. **Bottom left:** Angles of the knee bending sequence. **Top right:** Angles of the walking sequence. **Bottom right:** Angles of the leg crossing sequence.

are available upon request. The two systems can be identified by the smooth curves from the Motion Analysis system and unsmoothed curves (our system).

## 6 Summary

The contribution presents an approach for motion capture of clothed people. To achieve this we extend a silhouette-based motion capture system, which relies on image silhouettes and free-form surface patches of the body with a cloth draping procedure. Due to the limited time constraints for cloth draping we decided on a geometric approach based on kinematic chains. We call this cloth draping procedure kinematic cloth draping. This model is very well suited to be embedded in a motion capture system since it allows us to minimize the cloth draping parameters (and wind forces) within the same error functional such as the segmentation and pose estimation algorithm. Due to the number of unknowns for the segmentation, pose estimation, joints and cloth parameters, we decided on an iterative solution. The experiments with a skirt and shorts show that the formulated problem can be solved. We are able to determine joint configurations and pose parameters of the kinematic chains, though they are considerably covered with clothes. Indeed, we use the cloth draping appearance in images to recover the joint configuration and simultaneously determine wind dynamics of the cloth. We further performed a quantitative error analysis by comparing our method with a commercially available marker based tracking system. The experiments show that we are in the same error range as marker based tracking systems [15].

For future works we plan to extend the cloth draping model with more advanced ones [9] and we will compare different draping approaches and parameter optimization schemes in the motion capturing setup.

## References

1. C. Bregler, J. Malik, and K. Pullen. Twist based acquisition and tracking of animal and human kinetics. *International Journal of Computer Vision*, 56(3):179–194, 2004.
2. T. Brox, M. Rousson, R. Deriche, and J. Weickert. Unsupervised segmentation incorporating colour, texture, and motion. In N. Petkov and M. A. Westenberg, editors, *Proc. Computer Analysis of Images and Patterns*, volume 2756 of *Lecture Notes in Computer Science*, pages 353–360. Springer, Berlin, 2003.
3. A. Dervieux and F. Thomasset. A finite element method for the simulation of Rayleigh–Taylor instability. In R. Rautman, editor, *Approximation Methods for Navier–Stokes Problems*, volume 771 of *Lecture Notes in Mathematics*, pages 145–158. Springer, Berlin, 1979.
4. P. Fua, R. Plänkers, and D. Thalmann. Tracking and modeling people in video sequences. *Computer Vision and Image Understanding*, 81(3):285–302, March 2001.
5. D.M. Gavrilla. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding*, 73(1):82–92, 1999.
6. J. Haddon, D. Forsyth, and D. Parks. The appearance of clothing. <http://http.cs.berkeley.edu/haddon/clothingshade.ps>, June 2005.
7. L. Herda, R. Urtasun, and P. Fua. Implicit surface joint limits to constrain video-based motion capture. In T. Pajdla and J. Matas, editors, *Proc. 8th European Conference on Computer Vision*, volume 3022 of *Lecture Notes in Computer Science*, pages 405–418, Prague, May 2004. Springer.
8. D.H. House, R.W. DeVaul, and D.E. Breen. Towards simulating cloth dynamics using interacting particles. *Clothing Science and Technology*, 8(3):75–94, 1996.
9. N. Magnenat-Thalmann and P. Volino. From early draping to haute couture models: 20 years of research. *Visual Computing*, 21:506–519, 2005.
10. I. Mikic, M. Trivedi, E. Hunter, and P. Cosman. Human body model acquisition and tracking using voxel data. *International Journal of Computer Vision*, 53(3):199–223, 2003.
11. MoCap-System. Motion analysis: A marker based tracking system. [www.motionanalysis.com](http://www.motionanalysis.com), June 2005.
12. T.B. Moeslund and E. Granum. A survey of computer vision based human motion capture. *Computer Vision and Image Understanding*, 81(3):231–268, 2001.
13. R.M. Murray, Z. Li, and S.S. Sastry. *Mathematical Introduction to Robotic Manipulation*. CRC Press, Baton Rouge, 1994.
14. S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton–Jacobi formulations. *Computational Physics*, 79:12–49, 1988.
15. J. Richards. The measurement of human motion: A comparison of commercially available systems. *Human Movement Science*, 18:589–602, 1999.
16. B. Rosenhahn, U. Kersting, A. Smith, J. Gurney, T. Brox, and R. Klette. A system for markerless human motion estimation. In W. Kropatsch, R. Sablatnig, and A. Hanbury, editors, *Pattern Recognition, 27th DAGM-symposium*, volume 3663 of *Lecture Notes in Computer Science*, pages 230–237, Vienna, Austria, September 2005. Springer.
17. J. Weil. The synthesis of cloth objects. *Computer Graphics (Proc. SigGraph)*, 20(4):49–54, 1986.

# Unconstrained Multiple-People Tracking

Daniel Rowe<sup>1</sup>, Ian Reid<sup>2</sup>, Jordi González<sup>3</sup>, and Juan Jose Villanueva<sup>1</sup>

<sup>1</sup>Computer Vision Centre, Universitat Autònoma de Barcelona, Spain

<sup>2</sup>Active Vision Lab, Oxford University, United Kingdom

<sup>3</sup>Institut de Robòtica i Informàtica Industrial, UPC, Barcelona, Spain

**Abstract.** This work presents two main contributions to achieve robust multiple-target tracking in uncontrolled scenarios. A novel system which consists on a hierarchical architecture is proposed. Each level is devoted to one of the main tracking functionalities: target detection, low-level tracking, and high-level tasks such as target-appearance representation, or event management. Secondly, tracking performances are enhanced by on-line building and updating multiple appearance models. Successful experimental results are accomplished on sequences with significant illumination changes, grouping, splitting and occlusion events.

## 1 Introduction

Multiple human-beings tracking has become an active research field among the computer-vision community. This interest is motivated by an increasing number of applications related to Human Sequence Evaluation (HSE) [6]. Despite this interest, this still constitutes an open problem far from been solved. People tracking involves dealing with non-rigid targets whose dynamics are subject to sudden changes. In open-world applications, the number of agents within the scene may vary over time, and neither their appearance, nor their shape can be specified in advance. In unconstrained environments, the illumination and background-clutter distracters are uncontrolled, affecting the perceived appearance, which depends on issues such as the agents' position or orientation. Finally, agents interact among themselves, grouping and splitting, and causing occlusions.

Our goal is to implement and experimentally verify a novel approach which deals with the aforementioned difficulties. As a result, agents' trajectories will be obtained, as well as quantitative and qualitative information about their state at any time —such as their speed or whether they are being occluded. This paper is organized as follows: section 2 covers the most common current approaches; section 3 outlines the proposal; section 4 describes the low-level modules, whereas section 5 details the high-level appearance tracker; finally, section 6 shows some experimental results, and section 7 concludes this paper.

## 2 Related Work

Tracking can be carried out relying either on a bottom-up or a top-down approach. The former consists on foreground segmentation, and a subsequent target association, while the latter is based on complex shape and motion modelling.

Motion Segmentation can be performed by means of optical flow, background subtraction, or frame differencing. Correspondences can be accomplished using nearest neighbour techniques, or by means of Data Association filters [2]. A prediction stage is usually incorporated, thereby providing better chances of tracking success. Filters such as the Kalman filter, or extensions such as the EKF or UKF are commonly used. More general dynamics and measurement functions can be dealt with by means of Particle Filters (PF) [1].

On the other hand, high-level approaches rely on accurate target modelling [5]. Thus, complex templates and high-level motion patterns are a-priori learned, and used to reduce the state-space search region. Contour tracking have been widely explored [9], although this may be inappropriate in crowded scenarios with multiple target-occlusions. BraMBLe [8] is an interesting approach to multiple-blob tracking which models both background and foreground using MoG. However, no model update is performed, there is a common foreground model for all targets, and it may require an extremely large number of samples, since one sample contains information about the state of all targets. Nummiaro et al. [10] use a PF based on colour-histogram cues. However, no multiple-target tracking is considered, and it lacks from an independent observation process, since samples are evaluated according to the predicted image region histograms.

Comaniciu et al. [4] introduce an attractive technique — called mean shift — which tackles target localisation by performing a gradient-descent search on a image region of interest. However, their method tracks just one target, initialised by hand, and the appearance model is never updated. Collins et al. [3] present an effective enhancement with on-line selection of discriminative features. It aims to maximise the distinction between the target appearance and its surroundings. Still, it tracks just one target, initialised by hand and which may suffer from model drift. In both cases, just rigid target regions are tracked, and since multiple-target tracking is not considered, interaction events are not studied.

### 3 Approach Outline

Non-supervised multiple-human tracking is a complex task which demands a structured framework. This work presents a hierarchical system whose levels are devoted to the different functionalities to be performed, see Fig. 1.

Reliable target segmentation is critical in order to achieve an accurate feature extraction without considering prior knowledge about the potential targets, specially in dynamic scenes. However, complex agents who move through cluttered environments require high-level reasoning. Thus, this proposal consists on a bottom-up approach, whose results are eventually refined by a top-down process.

The lower level performs target detection. The first module accomplishes the segmentation task, while the second one filters the obtained image masks, extracts object blobs, and obtains object representations which can be handled by low-level trackers. The latter establish coherent target relations between frames. Firstly, *gates* —regions where the observations are expected to appear— are computed. Subsequently, *data association* is performed by setting correspondences

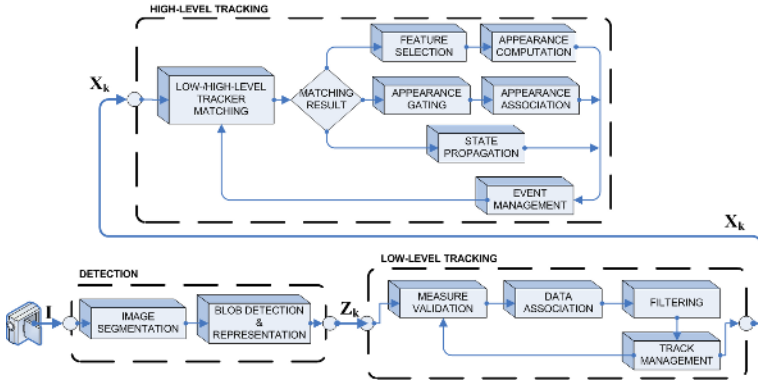


Fig. 1. System architecture

between observations and trackers. Afterwards, *filtering* is carried out by estimating new target states according to the associated observations using a bank of Kalman filters. Finally, the *track-management* module (i) initiates tentative tracks for those observations which are not associated; (ii) confirms tracks with enough supporting observations; and (iii) removes low-quality ones. Results are forwarded to high-level trackers, and fed back to the measure-validation module.

Confirmed low-level tracks are associated to high-level trackers. Hence, tracking events can be managed, and target tracking can be achieved even when image segmentation is not feasible, and low-level trackers are removed (during long-duration occlusions or grouping). Therefore, whenever the track is stable, the target appearance is computed and updated; those high-level trackers which remain orphans are processed to obtain an appearance-based data association, thereby establishing correspondences between lost high-level trackers and new ones; finally, those targets which have no correspondence are propagated according to the learned motion model. The *event* module determines what is happening within the scene, such as target grouping or entering the scene. These results are fed back allowing low-/high-level tracker matching.

## 4 Blob Detection and Low-Level Tracking

The first level aims to detect targets within the scene. Two modules are implemented to segment the image, and obtain a suitable object representation. Image segmentation is performed following the method proposed by Horprasert et al. [7] which is based on a colour background-subtraction approach. The background is statistically modelled on a pixel-wise basis, using a window of  $N$  frames. During this training period, the mean  $\mathbf{E}_i$  and standard deviation  $\sigma_i$  of each  $i$ th-pixel RGB-colour channel is computed. Two distortion measures are established:  $\alpha$ , the brightness distortion, and  $CD$ , the chromacity distortion. The variation over time of both distortions for each pixel is subsequently computed, and used as normalising factors for  $\alpha$  and  $CD$ , so that a single threshold —automatically

computed according to the learned pixels distribution— can be set for the whole image. This 4-tuple constitutes the pixel background model.

Pixels are classified into five categories, depending on their chromacity and brightness distortion: foreground, dark foreground (where no chromacity cues can be used), shadows, highlights, and normal background. Foreground blobs are subsequently detected: both foreground maps are fused; morphological operations are applied and a minimum-area filter is used; and remaining pixels are grouped into labelled blobs, their contours are extracted, and an ellipse representation is computed. Thus, the  $j$ -observed blob at time  $t$  is given by  $\mathbf{z}_j^t = (x_j^t, y_j^t, h_j^t, w_j^t, \theta_j^t)$ , where  $x_j^t, y_j^t$  represent the ellipse centroid,  $h_j^t, w_j^t$  are the major and minor axes, and  $\theta_j^t$  the ellipse orientation.

The target state is then estimated by filtering the sequence of noisy measures. In this work, it is assumed that human beings move slowly enough compared to the frame rate. Since their long-run dynamics are hardly predictable, a first-order dynamic model is adopted. This assumption holds in most HSE applications. The observation vector at time  $t$  is given by the blob detection module. The target state is then defined by  $\mathbf{x}_j^t = (x_j^t, \dot{x}_j^t, y_j^t, \dot{y}_j^t, h_j^t, \dot{h}_j^t, w_j^t, \dot{w}_j^t, \theta_j^t)$ . Thus, a constant-speed approach is used, where the acceleration is modelled as WAGN.

In a multiple-target tracking scenario, numerous observations may be obtained at every sampling period. Measure validation consists in establishing the regions where the target observations are expected. Thus, gates are set according to the innovation covariance matrix  $\mathbf{S}_k$ , and a specific Mahalanobis Square Distance (MSD), thereby defining an ellipsoid which encloses a probability mass given by the confidence interval associated with the MSD. This means that measures can be validated for a given confidence interval by calculating the MSD between the predicted observation and the actual one, and comparing this value with the Mahalanobis radius for this confidence interval.

Measures are associated to the nearest tracker in whose gate they lie, since observations are usually just within one target gate. This is intrinsic to motion segmentation: close targets are likely to be segmented just as one blob corresponding to the group. A bank of Kalman filters estimates the state of all targets detected within the scene. When no observation is associated to a particular target, its state is propagated according to the dynamic model.

Target tracks are instantiated, confirmed and removed according to the values of two indicators: the square root of  $\mathbf{S}_k$  determinant, and the observation MSD. The former is related to the track uncertainty given by the variance of the eigenvector dimensions. While an observation is associated,  $|\mathbf{S}_k|^{\frac{1}{2}}$  will decrease to its asymptotic value, and the time taken depends only on the system model. It is a reliable indicator of how many observations have been consecutively associated, without setting thresholds or specifying cases. The quality of the observation is taken into account by evaluating the MSD of each target associated observation. Therefore, a track is instantiated every time an observation remains orphan. When  $|\mathbf{S}_k|^{\frac{1}{2}}$  and the MSD value indicate that the track is stable, the tracker is confirmed. If  $|\mathbf{S}_k|^{\frac{1}{2}}$  grows far beyond reasonable values, the tracker is removed.

## 5 High-Level Appearance Tracker

The aforementioned bank of Kalman filters estimates the state of multiple targets. However, it cannot cope with those situations where segmentation fails, such as grouping events, or non-smooth changes in position or shape. These issues are addressed by implementing high-level trackers which include information relative to the target appearance and tracking events. Unfortunately, the target appearance cannot be specified in advance, and it should be continuously updated, since it strongly depends on the target position and orientation, and on the light sources. Further, in order to be able to track them when target segmentation is not feasible, it is modelled taking into account the local clutter.

In this work, the appearance-modelling approach presented by Collins et al. [3] is followed. This uses multiple colour features, which are evaluated and ranked. However, contrary to their method, a pool of features is now maintained, and smoothed characteristics are computed. Thus, the initialisation is solved, and tracker association is feasible once the event that cause the target loss is over. Possibilities of inconsistent localisations due to feature switch are minimised by introducing the distinction between long-run features and the current best ones.

### 5.1 Tracker Matching

This module performs the matching between low- and high-level trackers. Whenever a low-level tracker is confirmed, a high-level tracker is instantiated and associated. In case that the new-born tracker does not collide with two or more existing trackers, the target appearance will be computed (see Fig 1). In other case, it is marked as a group tracker. In subsequent tracker matchings, high-level tracker parameters relative to the target position and shape are updated. Further, while the track is still confirmed, appearances will also be updated.

Low-level trackers are removed during long-duration occlusions or groupings, since no observation is received, and the track loses confidence. In this case, the high-level tracker is not matched to any low-level tracker. Then, the system tries to associate it to new-born ones, presumably created once the event is over. If there are no tracker candidates, or they are not similar enough in the appearance sense, their state is propagated according to the learned motion model.

### 5.2 Feature Selection

The target appearance is represented using colour histograms, since they are less sensitive to rotations in depth or target deformations. Features are selected from a set of independent linear combinations of RGB channels, including raw R, G, and B, intensity, or chrominance approximations. Features are then normalised to the range  $[0 - 255]$ , and subsequently discretised into 64 bins. This is a sensitive decision: a low number of bins prevents from target-clutter disambiguation; on the other hand, a high value favours erroneous representations that appear when distributions are estimated from an insufficient number of samples. The

$i$ -feature target histogram is given by  $\mathbf{p}^i = \{p_k^i; k = 1 : K\}$ , where  $K$  is the number of bins. The probability of each feature is calculated as:

$$p_k^i = C^i \sum_{a=1}^M \delta(b(x_a) - k), \quad (1)$$

where  $C^i$  is a normalisation constant which ensures  $\sum_{k=1}^K p_k^i = 1$ ,  $\delta$  the Kronecker delta,  $\{x_a; a = 1 : M\}$  the pixel locations,  $M$  the number of target pixels, and  $b(x_a)$  a function that associates pixels to corresponding bins. In a similar way,  $\mathbf{q}^i$  represents the  $i$ -feature background histogram, computed from the background model. Then, log-likelihood ratios of each feature are computed as:

$$L^i(k) = \log \frac{\max(p_k^i, \epsilon)}{\max(q_k^i, \epsilon)}, \quad (2)$$

where  $\epsilon$  is set to the minimum histogram value to prevent dividing by zero or taking the logarithm of zero, but avoiding also magnifying the corresponding log-likelihood value. Thus, shared colour bins have a log-likelihood close to zero, whereas foreground bins have a positive one, and background bins a negative one. Features are then evaluated according to the variance-ratio of the log-likelihood:

$$VR^i(L; p, q) = \frac{\text{var}(L^i; (p^i + q^i)/2)}{\text{var}(L^i; p^i) + \text{var}(L^i; q^i)}, \quad (3)$$

which maximises the inter-class variance —background and target bin clusters—, while minimises the intra-class variance. Thus, features can be ranked according to their variance ratio: the higher, the better.

### 5.3 Appearance Computation

Contrary to the work of Collins, long-run features are kept and smoothed. These will be crucial for target loss recovery. Further, by smoothing the histograms the representation is less sensitive to possible localisation errors, and sudden and temporal appearance changes due to illumination fluctuations. A pool of  $M + N$  features is kept. These are the best  $M$  features at time  $t$ , and the best  $N$  long-run features: those which have been at top of the feature rank more times. These features are only dropped when new features enter the pool, and overcome them. For each  $M$  feature, the mean appearance histogram is recursively computed:

$$\mathbf{m}_t^i = \mathbf{m}_{t-1}^i + \frac{1}{n_i} (\mathbf{p}_t^i - \mathbf{m}_t^i), \quad (4)$$

where  $n_i$  is the number of times that the histogram has been updated. Similarity between two histograms is computed using the *Bhattacharyya distance*



$d_B = \sqrt{1 - \sum_{k=1}^K \sqrt{p_k q_k}}$ . A similarity criterion must establish when two histograms are close enough. Thus, the mean and variance of  $d_B$  between the smoothed histogram and the new one are also computed and updated:

$$\mu_t^i = \mu_{t-1}^i + \frac{1}{n-1} (d_{B,t}^i - \mu_t^i), \quad (5)$$

$$\sigma_t^2 = \frac{n-3}{n-2} \sigma_{t-1}^2 + (n-1) (\mu_t^i - \mu_{t-1}^i). \quad (6)$$

In this way, the Bhattacharyya distance distribution can be parameterised.

#### 5.4 Appearance Association

Low-level trackers lost their track during long-duration segmentation failures, such an occlusion event. Once the event is over, the target is again detected and a new tracker is instantiated. When this track become stable, it is confirmed and a high-level tracker is created. The former high-level tracker —and the target appearance models— were propagated. A tracker association process is performed, and the system concludes that both trackers are in fact representing the same target. This is done as follows: new-born trackers are handled as observations, and they are gated according to the lost trackers in the feature space. Thus, coincident features between both trackers are selected. Since feature selection depends on the local environment, and the targets move while they are grouped, the feature pool is subject to changes. However, the assumption that some long-run features are still good enough to be selected holds in most scenarios.

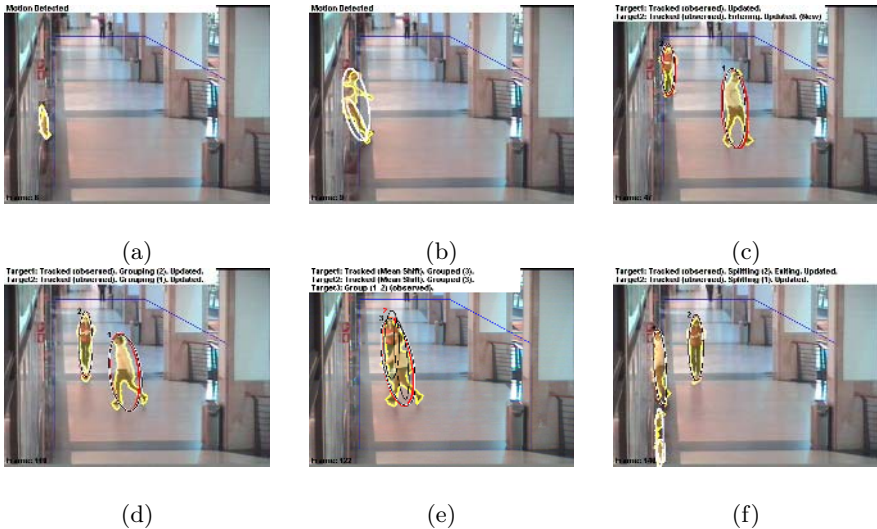
The Bhattacharyya distance between the histograms of each coincident feature is evaluated. Those which correspond to the the lost tracker are in fact smoothed models computed while the segmentation was reliable. Features are gated using the previously calculated mean and variance of the Bhattacharyya distance. Finally, the tracker is associated to the nearest one, according to the Bhattacharyya distance, within the gate. If none of the features is within the gate of the lost tracker, a new association process is tried at the next time step.

#### 5.5 Event Management

Six significant states are defined: single target, target grouping, grouped and splitting, and target entering and exiting the scene. Once the target position and size is estimated, a collision map is computed. Thus, when two single targets are colliding, their state change into *grouping*. If they also collide with a confirmed group tracker, their state is set to *grouped*. Once they no longer collide with a confirmed group tracker, their state change to *splitting*. If they stop colliding at any state, they become *single* again. The collision map is used also to determine whether a new-born tracker represents a group.



**Fig. 2.** (a) Segmentation: foreground pixels are painted on white, while those ones classified as dark foreground are on yellow, shadows on green, and highlights on red. (b) Detection: red ellipses represent each target, and yellow lines denote their contour.

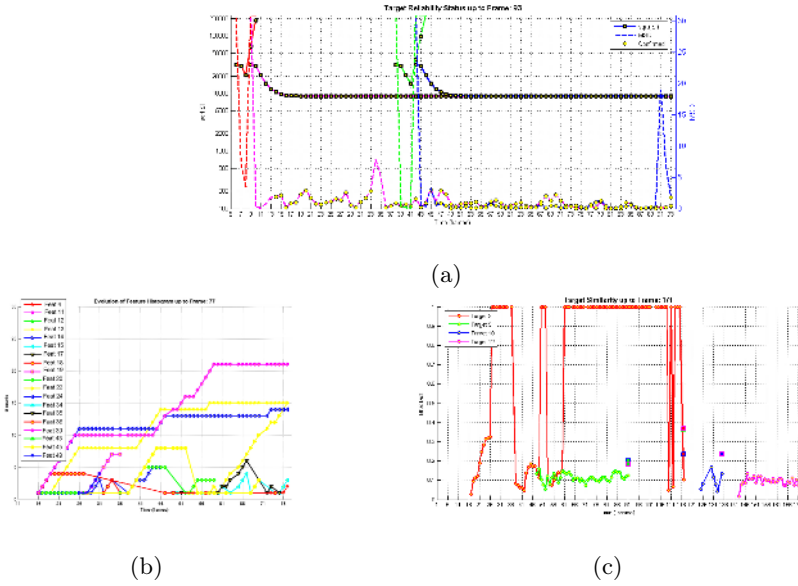


**Fig. 3.** Tracking results: red ellipses denote detections, whereas white and black ones are low- and high-level tracker estimates, respectively. The blue box denotes the ROI.

## 6 Experimental Results

The approach performance has been tested using the CAVIAR database. Two targets are tracked simultaneously, despite their being articulated and deformable objects whose dynamics are highly non-linear. One of them performs a rotation in depth and heads towards the second one, eventually occluding it. The background colour constitutes a strong source of clutter. Furthermore, the illuminant depends on both position and orientation. Significant speed, size, shape and appearance changes can be observed, jointly with events such as grouping or splitting, and occlusions.

Detection results are shown in Fig. 2, tracking ones in Fig. 3, and the low-tracking evolution in Fig 4.(a). At frame 6, an agent enters the scene, motion is



**Fig. 4.** (a) Tracks confirmation and removing. (b) Feature pool: the best  $M = 3$  features at time  $t$ , and the best  $N = 3$  long-run features are kept for appearance modelling. (c) Colour lines represent the intra-target similarity. Squares give the inter-tracker  $d_B$ .

detected, and a Kalman filter is created. A major size change occurs at frame 9: the target is completely inside the scene. Thus, a new Kalman filter is implemented, but both trackers are kept while their tracks have enough confidence, see Fig. 3.(b), and MSD value in Fig 4.(a). At frame 11, the first low-level tracker is removed. At frame 14, the track become stable, and a high-level tracker is instantiated. It is marked as a new born, entering the scene, and its appearance models as being updated. At frame 108, the segmentation of target 2 partially fails due to local illuminant changes, which leads to stop model updating. Grouping is detected at frame 110. At frame 122, a high-level tracker following the group is created. When the group splits, trackers are correctly re-associated.

The evolution of the feature pool for target 1 is shown in Fig 4.(b). Several facts can be noticed: some features are periodically among the best ones (features 13, 24 and 39); this repetitive behaviour is presumably due to the agent orientation and gait. Some features join the pool and quickly become one of the best ones as the agent moves and the background changes. Finally, other features are dropped and re-selected several times. These behaviours suggest that keeping a stable set of features may be useful for tracker association after tracking failure.

The Bhattacharyya distance between each new target detection and the smoothed model of feature 20 is represented in Fig 4.(c). When this feature is not selected or target cannot be detected, the distance is set to one. The inter-target  $d_B$  is also represented by two-colour squares, denoting both targets involved. At frame 91, the distance between the model of tracker 5 the one of

tracker 10 (the group) and 12 (the same target after the grouping) is computed. At frame 115 the same is done for tracker 3. The distance between tracker 1 and trackers 5 and 12 is almost double than the distance between the tracker 5 and 12, and in the same range of the intra-target distance computed during the successive detections. Thus, the association can be successfully carried out.

## 7 Conclusions

In this work a principle and structured system is presented in an attempt to take a step towards solving the numerous difficulties which appear in unconstrained tracking applications. It takes advantages of both bottom-up and top-down approaches. A robust and accurate tracking is achieved in a non-friendly environment with several non-white light sources, high appearance and shape target variability, and grouping, occlusion and splitting. Both targets are successfully tracked despite no a-priori knowledge is used. The system adapts itself depending on the number of targets, the best local features, or which events are taking place. Future research will be focused on developing a method to perform target localisation within a group region, once the best features for disambiguating targets from background are already computed and smoothed.

**Acknowledgements.** This work has been supported by the Research Department of the Catalan Government, the EC grant IST-027110 for the HERMES project, and by the Spanish MEC under projects TIC2003-08865 and DPI-2004-5414. J. González also acknowledges the support of a Juan de la Cierva Post-doctoral fellowship from the Spanish MEC.

## References

1. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A Tutorial on PFs for On-line Non-linear/Non-Gaussian Bayesian Tracking. *SP*, 50(2):174-188, 2002.
2. Y. Bar-Shalom and T. Fortran. *Tracking and Data Association*. A. Press, 1988.
3. R. Collins, Y. Liu, and M. Leordeanu. Online Selection of Discriminative Tracking Features. *PAMI*, 27(10):1631-1643, 2005.
4. D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based Object Tracking. *PAMI*, 25(5):564-577, 2003.
5. J. Deutscher and I. Reid. Articulated Body Motion Capture by Stochastic Search. *IJCV*, 61(2):185-205, 2005.
6. J. González. *Human Sequence Evaluation: The Key-frame Approach*. PhD thesis, UAB, Spain, 2004.
7. T. Horprasert, D. Harwood, and L. Davis. A Robust Background Subtraction and Shadow Detection. In *4th ACCV, Taipei, Taiwan*, volume 1, pages 983-988, 2000.
8. M. Isard and J. MacCormick. BraMBLe: A Bayesian Multiple-Blob Tracker. In *8th ICCV, Vancouver, Canada*, volume 2, pages 34-41. IEEE, 2001.
9. J. MacCormick and A. Blake. A Probabilistic Exclusion Principle for Tracking Multiple Objects. *IJCV*, 39(1):57-71, 2000.
10. K. Nummiaro, E. Koller-Meier, and L. Van Gool. An Adaptive Color-Based Particle Filter. *IVC*, 21(1):99-110, 2003.

# Robust Non-rigid Object Tracking Using Point Distribution Manifolds\*

Tom Mathes and Justus H. Piater

Department of Electrical Engineering and Computer Science  
Montefiore Institute, University of Liège  
Building B28, B-4000 Liège, Belgium  
mathes@montefiore.ulg.ac.be, justus.piater@ulg.ac.be

**Abstract.** We present an approach to non-rigid object tracking designed to handle textured objects in crowded scenes captured by non-static cameras. For this purpose, groups of low-level features are combined into a model describing both the shape and the appearance of the object. This results in remarkable robustness to severe partial occlusions, since overlapping objects are unlikely to be indistinguishable in appearance, configuration and velocity all at the same time. The model is learnt incrementally and adapts to varying illumination conditions and target shape and appearance, and is thus applicable to any kind of object. Results on real-world sequences demonstrate the performance of the proposed tracker. The algorithm is implemented with the aim of achieving near real-time performance.

## 1 Introduction

Typical object tracking applications include video surveillance for security or behaviour analysis, traffic-monitoring, sports analysis and human body tracking. In this work we develop a model-based technique able to cope with non-rigid objects in crowded scenes, involving many interacting targets with frequent mutual occlusions. We use single-view video streams taken by non-static cameras, which poses serious difficulties to tracking systems based on background models.

Many tracking approaches are based on more or less elaborate variants of background subtraction [1]. They can easily handle only static cameras, and object labels cannot be preserved throughout occlusions, except by using high-level scene interpretation algorithms. Most model-based object tracking methods use a fixed object representation, a so-called *template*, that describes the appearance or the shape of the tracked object. Most of these are based on colour histograms [2,3]. Such approaches tend to have problems with richly textured objects or multiple interacting objects having similar global appearance. Few convincing attempts have been made that track objects using feature points, although it is generally accepted that point-based methods should have some interesting properties. Some basic point-based solutions were developed by Arnaud and Mémmin [4] by combining a Rao-Blackwellized particle filter with a model consisting of a noisy, planar cloud of points, and by Bevilacqua et al. [5] who perform

---

\* This work has been sponsored by the Région Wallonne under DGTRE/WIST contract 031/5439.

smart point grouping based on self-organising maps. More sophisticated approaches include the work by Leordeanu and Collins [6] where feature pairs are coupled based on their pairwise statistics, by Tang and Tao [7] where objects are modelled adaptively with an attributed relational graph, and by Mathes and Piater [8], where point distribution models are learnt non-incrementally for tracking planar objects.

In our approach, each tracked object is described by a point distribution model [9] using feature vectors for local appearance instead of raw texture information. Such a model combines local appearance information with global shape information. The model is learnt incrementally and continuously, enabling it to accommodate to appearance and illumination changes. Point features tend to flicker in noisy image sequences or disappear due to occlusions, but as long as a reasonable subset of all the model points is visible in each frame, tracking can be performed reliably. The model can dynamically add good new features or remove bad old features. During occlusion by other tracked objects, model updating is disabled, rendering our tracker even more robust. Point landmarking is performed automatically, so that user interaction is only required to initialise the tracker in the first frame.

The following section explains how we extract interest points and how we describe their local appearance. Section 3 introduces the concept of point distribution manifolds, and Section 4 explains how they can be used for tracking purposes. Experimental results are given in Section 5.

## 2 Interest Points and Local Appearance

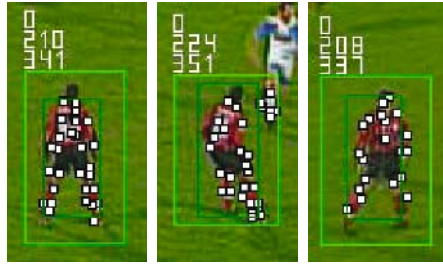
We concentrate on sparse sets of local features because they are well-suited for non-rigid objects and tend to yield methods particularly robust to partial occlusions. Local features are extracted by a colour version [10] of a scale-space, grey-scale Harris corner detector [11]. This is illustrated in Fig. 1. On each detected interest point we describe the local appearance by the 11-dimensional feature vector

$$\mathbf{v} = (x, y, r, g, b, r_x, r_y, g_x, g_y, b_x, b_y)^T, \quad (1)$$

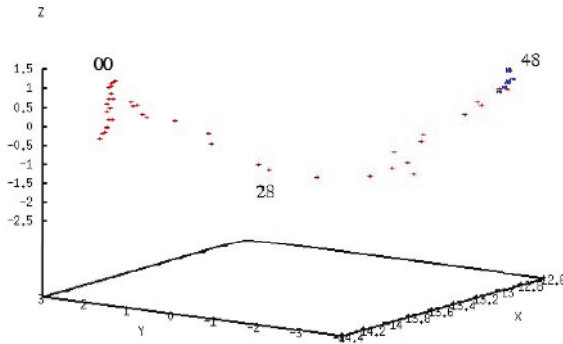
where  $\mathbf{v} \in \mathbf{V}$  with  $\mathbf{V} \subset \mathbb{R}^{11}$ .  $\mathbf{V}$  is called the *feature space*.  $\mathbf{v}$  corresponds to the first-order local jet enhanced by the interest point position. Using colour images and a rotationally variant descriptor yields enough discriminative power to obtain reliable point matchings between frames.

## 3 Point Distribution Manifolds

When using interest points to track objects, a natural approach is to use point distribution models, which are statistical models of shape and/or appearance. The shape of an object can be interpreted as all geometric information that remains when location, scale and rotational effects have been removed. Instead of using raw texture information, we describe the shape and appearance by constructing our model from a set of feature vectors that correspond to interest points that may lie anywhere on the object. Thus, each shape is represented by a vector



**Fig. 1.** Soccer player performing out-of-plane rotation (frames 0, 28 and 48)



**Fig. 2.** Projection onto the three principal components of a non-linear low-dimensional manifold corresponding to the normalised shapes of the rotating soccer player of Fig. 1

$$\mathbf{x} = \left( \mathbf{v}_x^{1T}, \mathbf{v}_x^{2T}, \dots, \mathbf{v}_x^{N^T} \right)^T \quad (2)$$

that is simply a concatenation of all the feature vectors extracted from the object in a given frame. These shape vectors lie in an  $11N$ -dimensional space, and more precisely on a low-dimensional, non-linear manifold  $\mathcal{M}$  embedded in this high-dimensional space, because the interest points on a real non-rigid object are strongly correlated. The shape and dimensionality of  $\mathcal{M}$  depend on the nature of the object deformations. Figure 2 shows the projection onto the three principal components of a typical manifold obtained for a rotating soccer player. It illustrates the potentially non-linear nature of the manifold.

### 3.1 Matching Interest Points

If  $\mathcal{M}$  is sampled densely enough and if we assume that it is locally linear, new shapes can be generated by linear interpolation of neighbouring shapes. Shapes within the image are denoted by the letter  $\mathbf{X}$ , whereas shapes that are part of the model are denoted by the letter  $\mathbf{Y}$ . Let us suppose we have used the model to generate a shape  $\mathbf{Y}$  superimposed onto the current video frame. In order to test if the current set of points  $\mathbf{X}$  taken from the image is a valid shape, the points from the image, indexed by  $i$ , and the points

from the model, indexed by  $j$ , have to be brought into correspondence. To do so, we compute a maximum-gain matching by using the *Hungarian method* [12]. We use the gain function

$$g(\mathbf{v}_{\tilde{\mathbf{X}}}^i, \mathbf{v}_{\mathbf{Y}}^j) = 1 - \frac{d(\mathbf{v}_{\tilde{\mathbf{X}}}^i, \mathbf{v}_{\mathbf{Y}}^j)}{\theta}, \quad (3)$$

where  $d(\mathbf{v}_{\tilde{\mathbf{X}}}^i, \mathbf{v}_{\mathbf{Y}}^j)$  is the distance between feature vectors  $\mathbf{v}_{\tilde{\mathbf{X}}}^i$  and  $\mathbf{v}_{\mathbf{Y}}^j$ . All edges with negative weights are ignored, meaning that matchings with distances greater than  $\theta$  are impossible. In this way,  $\theta$  acts as a gating threshold for  $d$ . The squared distance  $d^2$  between two  $N$ -sized vectors is computed as:

$$d^2(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y})^T \Sigma^{-1} (\mathbf{x} - \mathbf{y}), \quad (4)$$

where  $\Sigma$  is the covariance matrix estimated from all interest points belonging to  $\mathcal{M}$ . For computational reasons, the cross-correlations in  $\Sigma$  are assumed to be equal to 0, in order to avoid costly matrix inversions. Once the matchings have been performed, the interest points of  $\tilde{\mathbf{X}}$  can be rearranged so that they are in the same order as their correspondences in  $\mathbf{Y}$ . This new vector will be denoted by  $\mathbf{X}$ . In general, not all the points from  $\tilde{\mathbf{X}}$  (resp.  $\mathbf{Y}$ ) have a correspondence in  $\mathbf{Y}$  (resp.  $\tilde{\mathbf{X}}$ ). For this reason, we will have two vectors of equal size with missing elements, denoted by  $\mathbf{X}^\bullet$  and  $\mathbf{Y}^\bullet$ .

### 3.2 Image-to-Model and Model-to-Image Similarity

Before adding a shape to the manifold  $\mathcal{M}$ , it is centred to the origin and scaled such that the mean distance of its points to the origin is equal to 1. This operation defines a similarity transform  $\mathbf{T} = \mathbf{T}_{t_x, t_y, s, \alpha}$  that maps a shape from the manifold to the image reference frame. The inverse transform  $\mathbf{T}^{-1}$  can be used to map it back to the manifold. The translation and scaling are only applied to the  $x$  and  $y$  elements of the feature vectors, whereas the rotation must also be applied to the derivatives of the colour channels. Shapes in the image reference frame are denoted by upper-case letters (e.g.  $\mathbf{X}$ ), while the shapes in the model reference frame are denoted by lower-case letters (e.g.  $\mathbf{x}$ ). Thus we have  $\mathbf{x}^\bullet = \mathbf{T}^{-1}(\mathbf{X}^\bullet)$ .

### 3.3 Computing the Weights

The model is used to reconstruct a shape as similar as possible to the current, rearranged and reprojected image shape  $\mathbf{x}^\bullet$ . Our approach is similar to one popularly used for locally linear embedding [13]. We begin by identifying the  $K$  nearest neighbours  $\mathbf{y}_i$  of  $\mathbf{x}^\bullet$  on  $\mathcal{M}$  by applying the distance  $d$  defined earlier on the non-missing elements of  $\mathbf{x}^\bullet$ . Reconstruction errors are measured by the cost function

$$\epsilon(w) = \left| \mathbf{x}^\bullet - \sum_{i=1}^K w_i \mathbf{y}_i^\bullet \right|^2, \quad (5)$$

which is the squared distance between the image shape and its reconstruction and where again only the non-missing dimensions are considered. The weights  $w_i$  summarise the contribution of the  $i$ th model shape to the reconstruction. They are computed by min-





**Fig. 3.** A car and a soccer player with their current interest points (in white) and model points (in red) superimposed

imising the cost function subject to two constraints: First, the image shape is reconstructed only from its neighbours, enforcing  $w_i = 0$  if  $\mathbf{y}_i$  does not belong to this set; second, the weights must sum to one:  $\sum_i w_i = 1$ , thus enforcing the invariance to translation in the manifold space. The optimal weights subject to these constraints are found by solving the linear system of equations

$$\sum_j c_{ij} w_j = 1, \forall i, \quad (6)$$

where  $c_{ij}$  are the elements of the local covariance matrix defined by

$$c_{ij} = (\mathbf{x}^\bullet - \mathbf{y}_i^\bullet)^T (\mathbf{x}^\bullet - \mathbf{y}_j^\bullet). \quad (7)$$

For details on these calculations, see Appendix A of Saul and Roweis [13]. The weights are constrained to be larger than a small negative threshold to generate only shapes close to the convex hull of the  $K$  neighbouring shapes. We take  $K = 13$  in our experiments.

### 3.4 Shape Generation

If we assume that the manifold is locally linear, then we can predict the position of missing points. We compute the  $K$  nearest neighbours and the corresponding weights of the current image shape using only the dimensions corresponding to the model points that could be matched to image points, as described in the previous section. These same weights can then be used to predict the missing feature vectors, generating a complete vector  $\mathbf{x}$  equal to  $\mathbf{x}^\bullet$ , but with the missing values filled in by the corresponding values of  $\sum w_i \mathbf{y}_i$ . This is a very important step in our method because it solves the problem of flickering feature points. The generated shape  $\mathbf{y} = \mathbf{x}$  is then added to the manifold.

Due to opaque objects rotating in depth or object deformations, some feature points become hidden because they move behind the object. When generating shapes from the model, such hidden points should not be projected into the image. Therefore, for every feature point of every shape on the manifold there is a flag that indicates whether that point was visible at the moment the shape was added to the manifold. When we generate a new shape, a point is taken to be visible if among the  $K$  neighbouring shapes at least one flag is set. Figure 3 shows a car and a soccer player with the image interest points (in white) and the model points (in red) superimposed.

## 4 Tracking

### 4.1 Cleaning the Manifold

To make the model adaptive to changing object appearance or shape, we need a mechanism to add and remove points from it. Points appear and disappear usually when an object performs out-of-plane rotations or undergoes strong non-rigid deformations.

We call *new* points all those feature points in the current frame that cannot be matched to a model point. Each of these points is added to the model and matched to image points from frame to frame, but is not yet used to update the model parameters. This happens only when the point has proven to be stable, meaning that it has been matched a minimum number of times. We call these the *active* points. A similar methodology is applied to *inactive* points, which are model points that have not been matched for some time.

As we are interested in tracking objects in crowded scenes, our model was designed to be very robust to occlusions. Nevertheless, due to its incremental nature, bad points (points not belonging to the object) could be added to the model, especially if the background is very cluttered or the occluding object has similar velocity and/or appearance. The addition and deletion of points are therefore disabled as soon as the regions of interest of the two objects intersect each other.

In some situations (cluttered background or occluding object not being tracked), it still happens that a bad point is likely to be added to the model. We therefore discard points that lie too far from the 4-dimensional Gaussian cluster  $\mathcal{N}$  formed by the 4-D points  $\mathbf{q} = (x, y, v_x, v_y)$ , where  $(v_x, v_y)$  is the velocity vector of  $(x, y)$ . This means that we consider as outliers those points for which  $d^2(\mathbf{q}, \mathcal{N}) > \gamma^2$  and

$$d^2(\mathbf{q}, \mathcal{N}) = (\mathbf{q} - \mu)^T \Sigma_{\mathcal{N}}^{-1} (\mathbf{q} - \mu), \quad (8)$$

where  $\mu$  is the mean vector and  $\Sigma_{\mathcal{N}}$  is the covariance matrix of  $\mathcal{N}$ . We use  $\gamma = 3.0$ . This is analogous to the gating commonly used with Kalman filtering.

For computational reasons it is not possible to add new shapes indefinitely. We therefore generally limit the size of the manifold to a maximum of 30 shapes. When this limit is reached, the oldest shape is simply discarded. Keeping more than 30 shapes on the manifold doesn't improve the tracking results considerably.

### 4.2 Kalman Filtering

A Kalman filter is applied to the model-to-image similarity parameters. In our state vector  $\mathbf{p} = (t_x, t_y, s, \alpha, v_x, v_y) \in \mathbb{R}^6$ , the position is governed by a first-order process (constant-velocity model), whereas  $s$  and  $\alpha$  are governed by a zeroth-order process, giving  $\mathbf{p}_t = \mathbf{A}\mathbf{p}_{t-1} + \mathbf{u}_{t-1}$ , where  $\mathbf{A}$  is the state transition matrix. The corresponding measurement vector  $\mathbf{z} = (t_x, t_y, s, \alpha) \in \mathbb{R}^4$  is provided by  $\mathbf{z}_t = \mathbf{H}\mathbf{p}_t + \mathbf{v}_t$ , where  $\mathbf{H}$  is the measurement matrix. The random vectors  $\mathbf{u}_t$  and  $\mathbf{v}_t$  represent the process and measurement noise at time  $t$  respectively. They are assumed to be independent of each other, white and with normal probability distributions. In soccer or video surveillance the filter is tuned in order to allow only slow variations of scale and angle. In sequences with more chaotic movements, the scale and the angle can be made more flexible.

### 4.3 The Algorithm

Tracking is performed by applying the following algorithm to each frame:

1. In the current frame (time  $t$ ), extract interest points that lie inside the ROI from the previous frame, giving the current shape  $\tilde{\mathbf{X}}_t$ . The ROI is equal to the smallest rectangle enclosing the model shape from the previous frame plus a small border.
2. Project the model shape from the previous frame into the current frame by using the predicted similarity resulting from the previous time update of the Kalman filter:  $\mathbf{Y}_{t-1} = \hat{\mathbf{T}}_{t-1}(\mathbf{y}_{t-1})$ .
3. Match the points of  $\mathbf{Y}_{t-1}$  with the points of  $\tilde{\mathbf{X}}_t$ . This defines a vector  $\tilde{\mathbf{X}}_t^\bullet$  and a vector  $\mathbf{Y}_{t-1}^\bullet$  containing only the matched (and active) points of  $\tilde{\mathbf{X}}_t$  and  $\mathbf{Y}_{t-1}$ . Let  $\mathbf{X}_t^\bullet$  be the rearranged version of  $\tilde{\mathbf{X}}_t^\bullet$ .
4. Compute the new similarity  $\mathbf{T}_t$  that minimises  $|\mathbf{X}_t^\bullet - \mathbf{T}_t(\mathbf{y}_{t-1}^\bullet)|^2$ .
5. Compute the  $K$  nearest neighbours of  $\mathbf{x}_t^\bullet = \mathbf{T}_t^{-1}(\mathbf{X}_t^\bullet)$  on  $\mathcal{M}$  and the corresponding weight vector  $\mathbf{w}_t$ .
6. Use these weights  $\mathbf{w}_t$  to complete  $\mathbf{x}_t^\bullet$  in order to generate what is the most probable current image shape  $\mathbf{x}_t$ .
7. Add this completed shape  $\mathbf{y}_t = \mathbf{x}_t$  to the manifold.
8. Use the computed similarity  $\mathbf{T}_t$  as measurement for the Kalman filter and do a time update which gives  $\hat{\mathbf{T}}_t$ .
9. Clean the manifold as described in Section 4.1.

Our method can be directly applied to sequences that do not contain many background feature points. If the background is highly cluttered, meaning that it gives rise to large numbers of feature points, a pre-filtering stage may be required, e.g. to remove all static points in the case of a static camera or to remove all the image-to-model homography inliers in the case of a moving camera. This approach is different from traditional background subtraction, because it is performed only locally and does not require a background model.

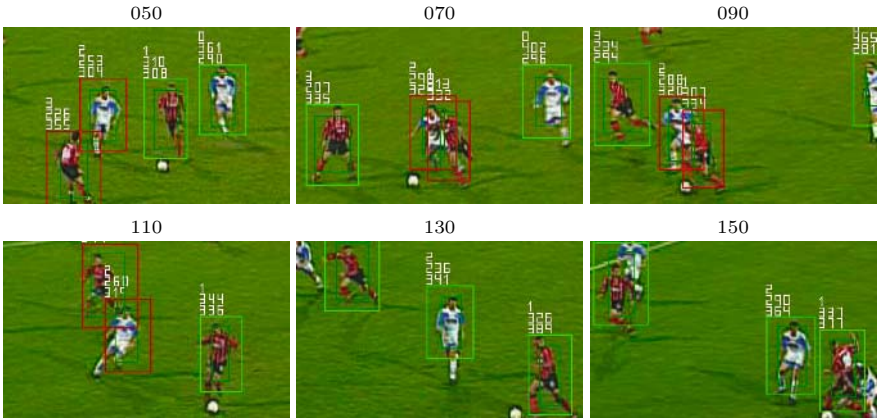
## 5 Results

We present tracking results on several challenging video sequences taken from a soccer game and from the PETS 2001 video surveillance data. Objects can be correctly tracked through scale, appearance and shape changes, as long as they exhibit sufficient texture. The tracker is not specific to people, but can also be used to track cars for example. Object labels are not lost during severe partial occlusions, even if the interacting targets look very similar. In all the examples, user interaction is only required in the first frame in order to initialise the targets to be tracked.

Example 1 is a difficult 150-frame sequence taken from a soccer game. The camera performs rotations and zooms whereas the players undergo drastic non-rigid deformations and very rapid movements, causing motion blur in some subsequences. In this sequence, four players are tracked. If their regions of interest intersect, their respective model learning is disabled, indicated by a red region of interest. The size of the regions of interest automatically adapts to the target size. The trackers are not disturbed by the



**Fig. 4.** Example 1: Global views of the sequence, illustrating the camera movement. The camera and the players perform fast movements, which causes some motion blur.

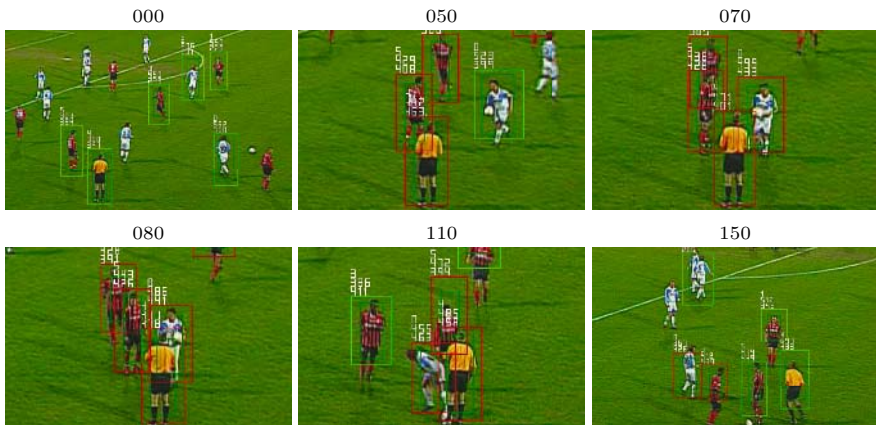


**Fig. 5.** Example 1: Local enlargements of interesting keyframes of the sequence

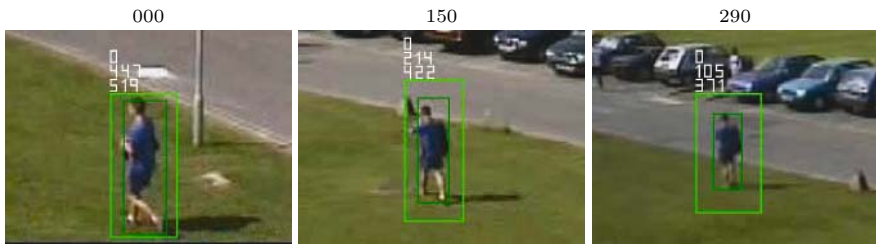
partial occlusions. Figure 4 shows four keyframes of the sequence to illustrate the camera movements. Figure 5 contains six enlargements from another subset of keyframes of the same sequence.

Example 2, illustrated in Fig. 6, is similar to the previous one, but with much more severe occlusions. The occlusions in this example are typical occlusions our tracker is able to handle without confusing target labels. In this sequence, five soccer players and the referee are tracked. Between frames 060 and 110 there is a very complicated occlusion situation between the referee and 3 other players, which is correctly handled by the tracker.

Example 3 illustrates the ability of our tracker to handle appearance and scale changes. A person walking away from the camera is correctly tracked for more than 300 frames. Three keyframes of this sequence are shown in Fig. 7. After frame 290, the tracker fails, mainly for two reasons: First, the target becomes very small and no longer generates enough interest points; secondly, in its current form, our algorithm still has some problems with very slowly-moving targets in front of highly cluttered backgrounds. A possible solution to this problem might be to eliminate static interest points (required only locally, inside the region of interest) in the case of a static camera or the inter-frame homography inliers in the case of a rotating and zooming camera.



**Fig. 6.** Example 2: 160-frame soccer sequence with 6 tracked targets undergoing complex mutual occlusions. Each player is correctly tracked throughout the sequence without labels being lost. Frame 000 is the frame in which the targets are initialised.



**Fig. 7.** Example 3: three representative frames from a 320-frame sequence taken from video surveillance. The tracked person undergoes strong appearance and scale changes.

All experiments were performed on a 1.7 GHz Celeron processor. Our current, non-optimised implementation runs at around 1.5 to 3.0 frames per second, depending on the image sizes ( $704 \times 576$  for the soccer sequences and  $768 \times 576$  for the PETS sequences), and linearly in the number of tracked targets. The tracking itself is very fast; the current bottleneck of our implementation is the Harris detector, which can be sped up dramatically using efficient implementations. The speed of each tracker depends on the maximum number of shapes on the manifold and on the number of interest points per shape.

## 6 Conclusion

We presented a novel, robust approach to tracking non-rigid, textured objects in crowded scenes. An incremental model is learnt that combines groups of feature points. This allows us to handle highly non-rigid targets such as running people. Our method behaves very well during partial occlusions in that target labels are generally preserved, and the objects' centres of gravity are correctly predicted. This latter property is essen-

tial for metric applications where the target position has to be mapped to the ground plane.

Our method is robust, because it takes into account the local appearance and the spatial configurations of feature points. It is highly unlikely that two targets look exactly the same, move at the same speed and are very close together. As the model is learnt automatically and incrementally, we can track any kind of object.

In contrast to histogram-based methods, our method works with any kind of object texture and can even handle objects that look similar to the background or to other tracked objects. Another advantage over background-subtraction methods is that we can easily work with non-static cameras. Our method does not necessarily work well with untextured objects, as it is based on feature points, although in many situations there are enough border points. Due to the incremental nature of our tracker, slowly-moving targets in front of cluttered backgrounds can also be lost. We will address this problem by efficient methods for removing background points.

## References

1. Stauffer, C., Grimson, W.E.L.: Adaptive background mixture models for real-time tracking. In: Proceedings IEEE Conf. on Computer Vision and Pattern Recognition. (1999)
2. Nummiaro, K., Koller-Meier, E., Gool, L.J.V.: An adaptive color-based particle filter. *Image Vision Comput.* **21**(1) (2003) 99–110
3. Okuma, K., Taleghani, A., de Freitas, N., Little, J.J., Lowe, D.G.: A boosted particle filter: Multitarget detection and tracking. In: ECCV (1). (2004) 28–39
4. Arnaud, E., Mémmin, E.: An efficient rao-blackwellized particle filter for object tracking. (2005)
5. Bevilacqua, A., Stefano, L.D., Vaccari, S.: Using local and global object's information to track vehicles in urban scenes. In: IEEE International Conference on Advanced Video and Signal based Surveillance (AVSS). (2005)
6. Leordeanu, M., Collins, R.: Unsupervised learning of object features from video sequences. In: Proc. of CVPR. (2005)
7. Tang, F., Tao, H.: Object tracking with dynamic feature graph. In: Proc. of ICCV. (2005)
8. Mathes, T., Piater, J.: Robust non-rigid object tracking using point distribution models. In: Proc. of British Machine Vision Conference (BMVC'05). (2005)
9. Cootes, T.F., Taylor, C.J., Cooper, D.H., Graham, J.: Active shape models – their training and application. *Computer Vision and Image Understanding* **61**(1) (1995) 38–59
10. Gouet, V., Boujema, N.: About optimal use of color points of interest for content-based image retrieval. Technical report, INRIA Rocquencourt (2002)
11. Dufournaud, Y., Schmid, C., Horaud, R.: Matching images with different resolutions. In: Proc. of the Conf. on Computer Vision and Pattern Recognition. (2000) 612–618
12. Kuhn, H.W.: The Hungarian method for solving the assignment problem. *Naval Research Logistics Quarterly* **2** (1955) 83–97
13. Saul, L.K., Roweis, S.T.: An introduction to locally linear embedding. (2001)

# A Variational Approach to Joint Denoising, Edge Detection and Motion Estimation

Alexandru Telea<sup>1</sup>, Tobias Preusser<sup>2</sup>, Christoph Garbe<sup>3</sup>,  
Marc Droske<sup>4</sup>, and Martin Rumpf<sup>5</sup>

<sup>1</sup> Eindhoven University of Technology  
alexto@win.tue.nl

<sup>2</sup> CeVis, University of Bremen  
tp@mevis.de

<sup>3</sup> IWR, University of Heidelberg  
Christoph.Garbe@iwr.uni-heidelberg.de

<sup>4</sup> UCLA, Los Angeles

Marc.Droske@math.ucla.edu

<sup>5</sup> INS, University of Bonn

martin.rumpf@ins.uni-bonn.de

**Abstract.** The estimation of optical flow fields from image sequences is incorporated in a Mumford–Shah approach for image denoising and edge detection. Possibly noisy image sequences are considered as input and a piecewise smooth image intensity, a piecewise smooth motion field, and a joint discontinuity set are obtained as minimizers of the functional. The method simultaneously detects image edges and motion field discontinuities in a rigorous and robust way. It comes along with a natural multi-scale approximation that is closely related to the phase field approximation for edge detection by Ambrosio and Tortorelli. We present an implementation for 2D image sequences with finite elements in space and time. It leads to three linear systems of equations, which have to be iteratively in the minimization procedure. Numerical results underline the robustness of the presented approach and different applications are shown.

## 1 Introduction

The task of motion estimation is a fundamental problem in computer vision. In low-level image processing, the accurate computation of object motion in scenes is a long standing problem, which has been addressed extensively. In particular, global variational approaches initiated by the work of Horn and Schunck [1] are increasingly popular. Initial problems such as the smoothing over discontinuities or the high computational cost have been resolved successfully [2,3,4]. Motion also poses an important cue for object detection and recognition. While a number of techniques first estimate the motion field and segment objects later in a second phase [5], an approach of both computing motion as well as segmenting objects at the same time is much more appealing. First advances in this direction were investigated in [6,7,8,9,10,11].

The idea of combining different image processing tasks into a single model in order to cope with interdependencies has drawn attention in several different fields. In image registration, for instance, a joint discontinuity approach for simultaneous registration, segmentation and image restoration has been proposed by Droske & Ring [12] and extended in [13] incorporating phase field approximations. Yezzi, Zöllei and Kapur [14] and Unal *et al.* [15] have combined segmentation and registration applying geodesic active contours described by level sets in both images. Vemuri *et al.* have also used a level set technique to exploit a reference segmentation in an atlas [16]. We refer to [17] for further references.

Cremers and Soatto [18,19] presented an approach for joint motion estimation and motion segmentation with one functional. Incorporating results from Bayesian inference, they derived an energy functional, which can be seen as an extension to the well-known Mumford–Shah [20] approach. Their functional involves the length of boundaries separating regions of different motion as well as a “fidelity-term” for the optical-flow assumption. Our approach is in particular motivated by their investigations, resolving the drawback of detecting edges in a parametric model, by a non-parametric approach.

Recently, highly accurate motion estimation [21] has been extended to contour-based segmentation [22] following a well known segmentation scheme [23]. The authors demonstrate that extending the motion estimator to edge detection in a variational framework leads to an increase in accuracy. However, as opposed to our framework, the authors do not include image denoising in their framework. Including a denoising functional together with motion estimation in a variational framework has been achieved by [24]. They report significant increases the accuracy of motion estimation, particularly with respect to noisy image sequences. However, edges are not detected, but errors of smoothing over discontinuities are lessened by formulating the smoothness constraint in a  $L_1$  metric.

We present the first approach of combining motion estimation, image denoising and edge detection in the same variational framework. This step will allow us to produce more accurate estimations of motion while detecting edges at the same time and preventing any smoothing across them.

The combination of denoising and edge detection with the estimation of motion results in an energy functional incorporating fidelity- and smoothness-terms for both the image and the flow field. Moreover, we incorporate an anisotropic enhancement of the flow along the edges of the image in the sense of Nagel and Enkelmann [2]. The model is implemented using the phase-field approximation in the spirit of Ambrosio’s and Tortorelli’s [25] approach for the original Mumford–Shah functional. The identification of edges is phrased in terms of a phase field function, no *a-priori* knowledge of objects is required, as opposed to formulations of explicit contours. In contrast to a level set approach, the built-in multi-scale enables a robust and efficient computation and no initial guess for the edge set is required. We present here a truly  $d + 1$  dimensional algorithm, considering time as an additional dimension to the  $d$ -dimensional image data.



This fully demonstrates the conceptual advantages of the joint approach. The characteristics of our approach are:

- The distinction of smooth motion fields and optical flow discontinuities is directly linked to edge detection, improving the reliability of estimates.
- Image denoising and segmentation profits from explicit coupling of the sequence via the brightness constancy assumption.
- The phase field approximation converges to a limit problem for a vanishing scale parameter, with a representation of edges and motion discontinuities without any additional filtering.
- The algorithm is an iterative approach. In each step, a set of three simple linear systems are solved, requiring only a small number of iterations.

## 2 Generalized Optical Flow Equation

In image sequences, we observe different types of motion fields: locally smooth motion visible via variations of object shading and texture in time, or jumps in the motion velocity apparent at edges of objects moving in front of a background. We aim for an identification of corresponding piecewise smooth optical flow fields in piecewise smooth image sequences

$$u : [0, T] \times \Omega \mapsto \mathbb{R}; \quad (t, x) \rightarrow u(t, x)$$

for a finite time interval  $[0, T]$  and a spatial domain  $\Omega \subset \mathbb{R}^d$  with  $d = 1, 2, 3$ . The flow fields are allowed to jump on edges in the image sequence. Hence, the derivative  $Du$  splits into a singular and a regular part. The regular part is a classical gradient  $\nabla_{(t,x)}u$  in space and time, whereas the singular part lives on the singularity set  $S$  - the set of edge surfaces in space-time. Time slices of  $S$  are the actual image edges  $S$  with respect to space-time. The singular part represents the jump of the image intensity on  $S$ , i. e., one observes that  $D^s u = (u^+ - u^-)n_s$ . Here,  $u^+$  and  $u^-$  are the upper and lower intensity values on both sides of  $S$ , respectively. We now suppose that the image sequence  $u$  reflects an underlying motion with a piecewise smooth motion velocity  $v$ , which is allowed to jump only on  $S$ . Thus,  $S$  represents object boundaries moving in front of a possibly moving background. In this general setting, without any smoothness assumption on  $u$  and  $v$ , we ask for a generalized optical flow equation. Apart from moving object edges, we derive from the brightness constancy assumption  $u(t + s, x + s v) = \text{const}$  on motion trajectories  $\{(t + s, x + s v) : s \in [0, T]\}$ , that

$$\nabla_{(t,x)}u \cdot w = 0 \quad \text{and on the edge} \quad n_s \cdot (w^+ + w^-) = 0. \tag{1}$$

where  $w = (1, v)$  is the space-time motion velocity. This in particular includes the case of a sliding motion without any modification of the object overlap, where  $n_s \cdot w^+ = n_s \cdot w^- = 0$ .

### 3 Mumford–Shah Approach to Optical Flow

In their pioneering paper, Mumford and Shah [20] proposed the minimization of the following energy functional:

$$E_{MS}[u, S] = \lambda \int_{\Omega} (u - u_0)^2 \, d\mathcal{L} + \frac{\mu}{2} \int_{\Omega \setminus S} \|\nabla u\|^2 \, d\mathcal{L} + \nu \mathcal{H}^{d-1}(S), \quad (2)$$

where  $u_0$  is the initial image defined on an image domain  $\Omega \subset \mathbb{R}^d$  and  $\lambda, \mu, \nu$  are positive weights. Here, one asks for a piecewise smooth representation  $u$  of  $u_0$  and an edge set  $S$ , such that  $u$  approximates  $u_0$  in the least-squares sense.  $u$  should be smooth apart from the free discontinuity  $S$ . In addition,  $S$  should be smooth and thus small with respect to the  $(d-1)$ -dimensional Hausdorff-measure  $\mathcal{H}^{d-1}$ . Mathematically, this problem has been treated in the space of functions of bounded variation  $BV$ , more precisely in the specific subset  $SBV$  [26]. In this paper, we will pick up a phase field approximation for the Mumford–Shah functional (2) proposed by Ambrosio and Tortorelli [25]. They describe the edge set  $S$  by a phase field  $\phi$  which is supposed to be small on  $S$  and close to 1 apart from edges, i. e., one asks for minimizers of the energy functional

$$E_{\epsilon}[u, \phi] = \int_{\Omega} \lambda(u - u_0)^2 + \frac{\mu}{2}(\phi^2 + k_{\epsilon})\|\nabla u\|^2 + \nu\epsilon\|\nabla\phi\|^2 + \frac{\nu}{4\epsilon}(1 - \phi)^2 \, d\mathcal{L}, \quad (3)$$

where  $\epsilon$  is a scale parameter and  $k_{\epsilon} = o(\epsilon)$  a small positive regularizing parameter, which mathematically ensures strict coercivity with respect to  $u$ . Hence, the second term measures smoothness of  $u$  but only apart from edges. On edges, the weight  $\phi^2$  is expected to vanish. The last two terms in the integral encode the approximation of the  $d - 1$  dimensional edge set area and strongly favours a phase field value 1 away from edges, respectively. For larger  $\epsilon$ , one obtains coarse, blurred representations of the edge set and corresponding smoother images  $u$ . With decreasing  $\epsilon$  we successively refine the representation of the edges and include more image details.

Now, we ask for a simultaneous denoising, segmentation and flow extraction on image sequences. Hence, we will incorporate the motion field generating an image sequence into a variational method. We first formulate a corresponding minimization problem in the spirit of the Mumford–Shah model:

**Mumford–Shah type optical flow approach.** *Given a noisy initial image sequence  $u_0 : D \mapsto \mathbb{R}$  on the space time domain  $D = [0, T] \times \Omega$ , we ask for a piecewise smooth image sequence  $u$ , which jumps on a set  $S$ , and a piecewise smooth motion field  $w = (1, v)$ , which is allowed to jump on the same set  $S$ , with the constraint  $n_s \cdot (w^+ + w^-) = 0$ , such that  $(u, w, S)$  minimize the energy*

$$\begin{aligned}
 E[u, w, S] = & \int_D \frac{\lambda_u}{2} (u - u_0)^2 + \frac{\lambda_w}{2} (w \cdot \nabla_{(t,x)} u)^2 \, d\mathcal{L} + \int_D \frac{\mu_u}{2} (\phi^2 + k_\epsilon) \|\nabla_{(t,x)} u\|^2 \, d\mathcal{L} \\
 & + \int_D \frac{\mu_w}{2} \|P[\phi] \nabla_{(t,x)} w\|^q \, d\mathcal{L} + \int_D \left( \nu \epsilon \|\nabla \phi\|^2 + \frac{\nu}{4\epsilon} (1 - \phi)^2 \right) \, d\mathcal{L}. \quad (4)
 \end{aligned}$$

The first and second term of the energy are fidelity terms with respect to the image intensity and the regular part of the optical-flow-constraint, respectively. The third and fourth term encode the smoothness requirement of  $u$  and  $w$ . Finally, the last terms represents the area of the edge surfaces  $S$ , parameterized by the phase file  $\pi$ . The projection operator  $P[\phi]$  couples the smoothness of the motion field  $w$  to the image geometry:

$$P[\phi] = \alpha(\phi^2) \left( \mathbb{I} - \beta(\phi^2) \frac{\nabla_{(t,x)} \phi}{\|\nabla_{(t,x)} \phi\|} \otimes \frac{\nabla_{(t,x)} \phi}{\|\nabla_{(t,x)} \phi\|} \right).$$

Here,  $k_\epsilon = o(\epsilon)$  is a "safety" coefficient, which will ensure existence of solutions of our approximate problem.  $\alpha : \mathbb{R} \rightarrow \mathbb{R}_0^+$  and  $\beta : \mathbb{R} \rightarrow \mathbb{R}_0^+$  are continuous blending functions. For vanishing  $\epsilon$  and a corresponding steepening of the slope of  $u$ , this operator basically leads to a 'one sided diffusion' in the energy relaxation. The fidelity weights  $\lambda_u, \lambda_w$ , the regularity weights  $\mu_u, \mu_w$  and the weight  $\nu$  controlling the phase field are supposed to be positive and  $q \geq 2$ . We emphasize that, without any guidance from the local time-modulation of shading or texture on both sides of an edge, there is still a undecidable ambiguity with respect to foreground and background.

### 4 Variations of the Energy and an Algorithm

In what follows, we will consider the Euler-Lagrange equations of the above energies. Thus, we need to compute the variations of the energy contributions with respect to the involved unknowns  $u, w, \phi$ . Using straightforward differentiation for sufficiently smooth  $u, w, \phi$  and initial data  $u_0$  and summing up the resulting terms, we can integrate by parts and end up with the following system of PDEs

$$-\operatorname{div}_{(t,x)} \left( \frac{\mu_u}{\lambda_u} (\phi^2 + k_\epsilon) \nabla_{(t,x)} u + \frac{\lambda_w}{\lambda_u} w (\nabla_{(t,x)} u \cdot w) \right) + u = u_0 \quad (5)$$

$$-\epsilon \Delta_{(t,x)} \phi + \left( \frac{1}{4\epsilon} + \frac{\mu_u}{2\nu} \|\nabla_{(t,x)} u\|^2 \right) \phi = \frac{1}{4\epsilon} \quad (6)$$

$$-\frac{\mu_w}{\lambda_w} \operatorname{div}_{(t,x)} (P[\phi] \nabla_{(t,x)} w) + (\nabla_{(t,x)} u \cdot w) \nabla_{(t,x)} u = 0 \quad (7)$$

as the Euler-Lagrange equations characterizing the necessary conditions for a solution  $(u, w, \phi)$  of the above stated phase field approach. Let us emphasize that the full Euler-Lagrange equations, characterizing a global minimizer of the energy, would in addition involve variations of  $E_{\text{reg},w}$  with respect to  $\phi$ .

Following again Ambrosio and Tortorelli, our resulting algorithm involves an iteration solving three linear partial differential equations:

- Step 0.** Initialize  $u = u_0$ ,  $\phi \equiv 1$ , and  $w \equiv (1, 0)$ .
- Step 1.** Solve (5) for fixed  $w$ ,  $\phi$ .
- Step 2.** Solve (6) for fixed  $u$ ,  $w$ .
- Step 3.** Solve (7) for fixed  $u$ ,  $\phi$ , return to **Step 1** if not converged.

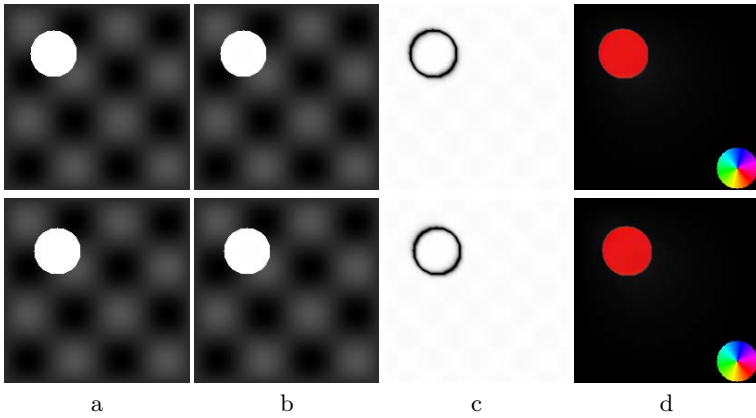
## 5 Finite Element Discretization

We proceed similarly to the Finite Element method proposed by Bourdin and Chambolle [27,28] for the phase field approximation of the Mumford–Shah functional. To solve the above system of PDEs, we discretize  $[0, T] \times \Omega$  by a regular hexahedral grid. In the following, the spatial and temporal grid cell sizes are denoted by  $h$  and  $\tau$  respectively, i.e. image frames are at a distance of  $\tau$  and pixels of each frame are sampled on a regular mesh with cell size  $h$ . To avoid tri-linear interpolation problems, we subdivide each hexahedral cell into 6 tetrahedra. On this tetrahedral grid, we consider the space of piecewise affine, continuous functions  $\mathcal{V}$  and ask for discrete functions  $U, \Phi \in \mathcal{V}$  and  $V \in \mathcal{V}^2$ , such that the discrete and weak counterparts of the Euler Lagrange equations (5), (6) and (7) are fulfilled. This leads to solving systems of linear equations for the vectors of the nodal values of the unknowns  $U, \Phi, V$ . Using an efficient custom-designed compressed row sparse matrix storage, we can treat datasets of up to  $K = 10$  frames of  $N = 500, M = 320$  pixels in less than 1GB memory. The linear systems of equations are solved applying a classical conjugate gradient method. For the pedestrian sequence (Fig. 5), one such iteration takes 47 seconds on a Pentium IV PC at 1.8 GHz running Linux. The complete method converges after 2 or 3 such iterations. Large video sequences are computed by shifting a window of  $K = 6$  frames successively in time. Thus temporal boundary effects are avoided.

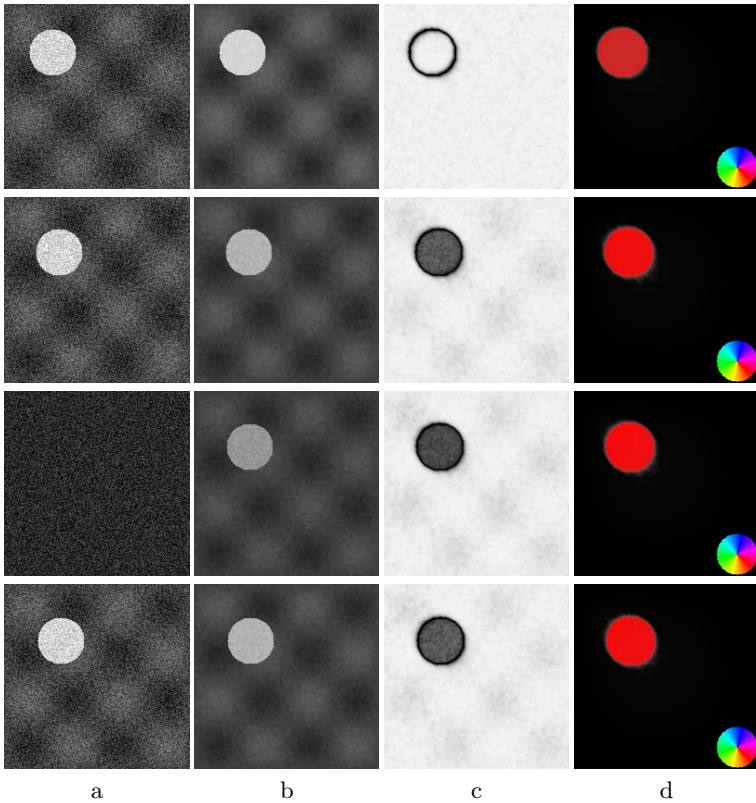
## 6 Results and Discussion

We present here several results of the proposed method for two dimensional image sequences. In the considered examples, the parameter setting  $\epsilon = h/4$ ,  $\mu_u = h^{-2}$ ,  $\mu_w = \lambda_u = 1$ ,  $\lambda_w = 10^5 h^{-2}$  and  $C(\epsilon) = \epsilon$ ,  $\delta = \epsilon$  has proven to give good results.

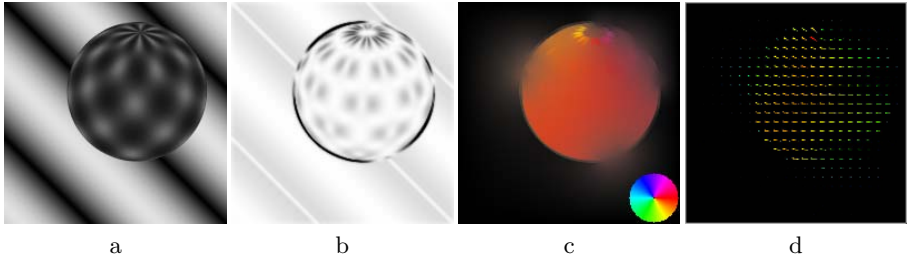
We first consider a simple example of a white disk moving with constant speed  $v = (1, 1)$  on a black background (Fig. 1). A small amount of smoothing results from the regularization energy  $E_{\text{reg},u}^\epsilon$  (Fig. 1(b)), which is desirable to ensure robustness in the resulting optical flow term  $\nabla_{(t,x)} u \cdot w$  and removes noisy artifacts in real-world videos, e.g. Fig. 4 and Fig. 5. The phase field clearly captures the moving object’s contour. The optical flow is depicted in Fig. 1(c) by color coding the vector directions as shown by the lower-right color wheel. Clearly, the method is able to extract the uniform motion of the disc. The optical flow information, available only on the motion edges (black in Fig. 1(c)), is propagated into the information-less area inside the moving disk, yielding the final result.



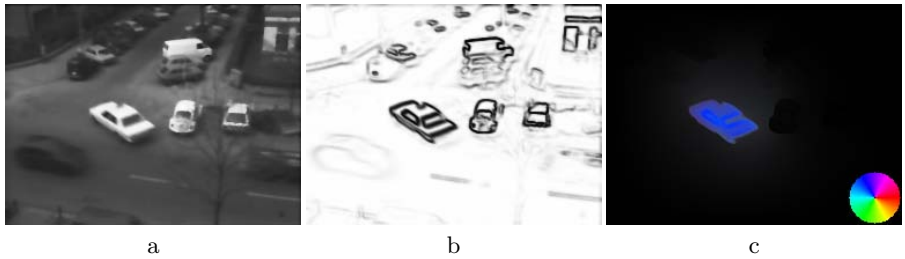
**Fig. 1.** Top to bottom two frames of the test sequence (a) and corresponding smoothed image (b), phase field (c) and optical flow (color coded) (d)



**Fig. 2.** Noisy circle sequence: From top to bottom, frames 3 and 9 – 11 are shown. (a) original image sequence, (b) smoothed images, (c) phase field, (d) estimated motion (color coded).



**Fig. 3.** Rotating sphere: smoothed image (a), phase field (b), optical flow (color coded) (c), optical flow (vector plot, color coded magnitude) (d)

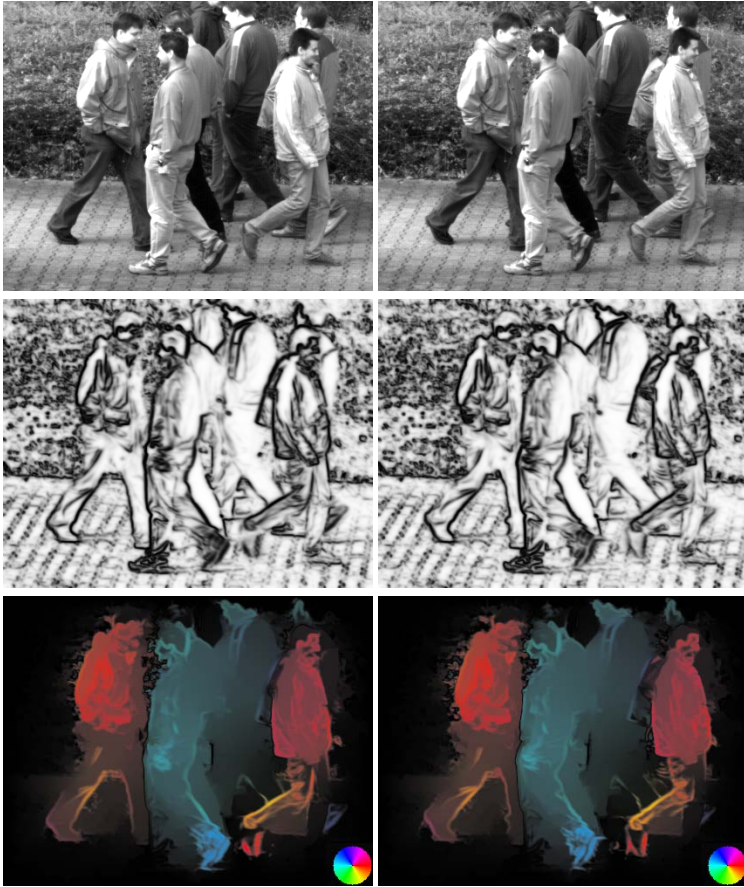


**Fig. 4.** Taxi sequence: smoothed image (a), phase field (b), and flow field (c)

In the next example, we revisit the simple moving circle sequence, but add noise to it. We also completely destroy the information of frame 10 in the sequence (Fig. 2). Figure 2 shows the results for frames 3 and 9 – 11. We see that the phase field detects the missing circle in the destroyed frame as a temporal edge surface in the sequence, i.e.  $\phi$  drops to zero in the temporal vicinity of the destroyed frame. This is still visible in the previous and next frames, shown in the second and third row. However, this does not hamper the restoration of the correct optical flow field, shown in the fourth column. This result is due to the anisotropic smoothing of information from the frames close to the destroyed frame. For this example, we used  $\epsilon = 0.4h$ .

A second synthetic example is shown in Fig. 3, using data from the publicly available collection at [29]. Here, a textured sphere spins on a textured background (Fig. 3(a)). Again, our method is able to clearly segment the moving object from the background, even though the object doesn't change position. We used a phase field parameter  $\epsilon = 0.15h$ . The extracted optical flow clearly shows the spinning motion (Fig. 3(d)) and the discontinuous motion field.

We next consider a known real video sequence, the so-called Hamburg taxi sequence. Figure 4 shows the smoothed image ( $u$ ), phase field  $\phi$  and color-coded optical flow field ( $w$ ). Our method detects well the image edges (Fig. 4 b). Also, the upper-left rotating motion of the central car is extracted accurately (Fig. 4 c). As it should be, the edges of the stationary objects, clearly visible in the phase field, do not contribute to the optical flow. Moreover, the moving



**Fig. 5.** Pedestrian video: frames from original sequence (top); phase field (middle); optical flow, color coded (bottom)

car is segmented as one single object in the optical flow field, i.e. the motion information is extended from the moving edges, i.e. car and car windscreen contours, to the whole moving shape.

Finally, we consider a complex video sequence, taken under outdoor conditions by a monochrome video camera. The sequence shows a group of walking pedestrians (Fig. 5 (top)). The human silhouettes are well extracted and captured by the phase field (Fig. 5(middle)). We do not display a vector plot of the optical flow, as it is hard to interpret it visually at the video sequence resolution of 640 by 480 pixels. However, the color-coded optical flow plot (Fig. 5(bottom)) shows how the method is able to extract the moving limbs of the pedestrians. The overall red and blue color corresponds to the walking directions of the pedestrians. The estimated motion is smooth inside the areas of the individual pedestrians and not smeared across the motion boundaries. In addition, the algorithm nicely

segments the different moving persons. The cluttered background poses no big problem to the segmentation, nor are the edges of occluding and overlapping pedestrians, who are moving at almost the same speed.

## References

1. Horn, B.K.P., Schunk, B.: Determining optical flow. *Artificial Intelligence* **17** (1981) 185–204
2. Nagel, H.H., Enkelmann, W.: An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. *IEEE Trans. on PAMI* **8**(5) (1986) 565–593
3. Weickert, J., Schnörr, C.: A theoretical framework for convex regularizers in pde-based computation of image motion. *Int. J. of Comp. Vision* **45**(3) (2001) 245–264
4. Bruhn, A., Weickert, J., Feddern, C., Kohlberger, T., Schnörr, C.: Real-time optical flow computation with variational methods. In Petkov, N., Westenberg, M.A., eds.: *CAIP 2003*. Volume 2756 of LNCS., Springer (2003) 222–229
5. Wang, J.Y.A., Adelson, E.H.: Representating moving images with layers. *IEEE Trans. on Im. Proc.* **3**(5) (1994) 625–638
6. Schnörr, C.: Segmentation of visual motion by minimizing convex non-quadratic functionals. In: *12th ICPR*. (1994)
7. Odobez, J.M., Bouthemy, P.: Robust multiresolution estimation of parametric motion models. *J. of Vis. Comm. and Image Rep.* **6**(4) (1995) 348–365
8. Odobez, J.M., Bouthemy, P.: Direct incremental model-based image motion segmentation for video analysis. *Sig. Proc.* **66** (1998) 143–155
9. Caselles, V., Coll, B.: Snakes in movement. *SIAM J. Num. An.* **33** (1996) 2445–2456
10. Memin, E., Perez, P.: A multigrid approach for hierarchical motion estimation. In: *ICCV*. (1998) 933–938
11. Paragios, N., Deriche, R.: Geodesic active contours and level sets for the detection and tracking of moving objects. *IEEE Trans. on PAMI* **22**(3) (2000) 266–280
12. Droske, M., Ring, W.: A Mumford-Shah level-set approach for geometric image registration. *SIAM Appl. Math.* (2005) to appear.
13. Authors: Mumford-shah based registration. *Computing and Visualization in Science* (2005) submitted.
14. Kapur, T., Yezzi, L., Zöllei, L.: A variational framework for joint segmentation and registration. *IEEE CVPR* (2001) 44–51
15. Unal, G., Slabaugh, G., Yezzi, A., Tyan, J.: Joint segmentation and non-rigid registration without shape priors. (2004)
16. Vemuri, B., Ye, J., Chen, Y., Leonard, C.: Image registration via level-set motion: Applications to atlas-based segmentation. *Med. Im. Analysis* **7** (2003) 1–20
17. Davatzikos, C.A., Bryan, R.N., Prince, J.L.: Image registration based on boundary mapping. *IEEE Trans. Med. Imaging* **15**(1) (1996) 112–115
18. Cremers, D., Soatto, S.: Motion competition: A variational framework for piecewise parametric motion segmentation. *Int. J. of Comp. Vision* **62**(3) (2005) 249–265
19. Cremers, D., Kohlberger, T., Schnörr, C.: Nonlinear shape statistics in mumford-shah based segmentation. In: *7th ECCV*. Volume 2351 of LNCS. (2002) 93–108
20. Mumford, D., Shah, J.: Optimal approximation by piecewise smooth functions and associated variational problems. *Comm. Pure Appl. Math.* **42** (1989) 577–685



21. Brox, T., Bruhn, A., Papenber, N., Weickert, J.: High accuracy optical flow estimation based on a theory for warping. In Pajdla, T., Matas, J., eds.: Proc. of the 8th ECCV. Volume 3024 of LNCS. (2004) 25–36
22. Amiaz, T., Kiryati, N.: Dense discontinuous optical flow via contour-based segmentation. In: Proc. ICIP 2005. Volume III. (2005) 1264–1267
23. Vese, L., Chan, T.: A multiphase level set framework for image segmentation using the mumford and shah model. *Int. J. Computer Vision* **50** (2002) 271–293
24. Nir, T., Kimmel, R., Bruckstein, A.: Variational approach for joint optic-flow computation and video restoration. Technical report, Dep. of C. S. - Israel Inst. of Tech., Haifa, Israel (2005)
25. Ambrosio, L., Tortorelli, V.M.: On the approximation of free discontinuity problems. *Boll. Un. Mat. Ital. B* **6**(7) (1992) 105–123
26. Ambrosio, L., Fusco, N., Pallara, D.: Functions of bounded variation and free discontinuity problems. Oxford University Press (2000)
27. Bourdin, B.: Image segmentation with a Finite Element method. *ESIAM: Math. Modelling and Num. Analysis* **33**(2) (1999) 229–244
28. Bourdin, B., Chambolle, A.: Implementation of an adaptive Finite-Element approximation of the Mumford-Shah functional. *Numer. Math.* **85**(4) (2000) 609–646
29. Group, C.V.R.: Optical flow datasets. Univ. of Otago, New Zealand, [www.cs.otago.ac.nz/research/vision](http://www.cs.otago.ac.nz/research/vision) (2005)

# Multi-step Multi-camera View Planning for Real-Time Visual Object Tracking

Benjamin Deutsch\*, Stefan Wenhardt, and Heinrich Niemann

Chair for Pattern Recognition, University of Erlangen-Nuremberg  
{deutsch, wenhardt, niemann}@informatik.uni-erlangen.de  
<http://www5.informatik.uni-erlangen.de>

**Abstract.** We present a new method for planning the optimal next view for a probabilistic visual object tracking task. Our method uses a variable number of cameras, can plan an action sequence several time steps into the future, and allows for real-time usage due to a computation time which is linear both in the number of cameras and the number of time steps. The algorithm can also handle object loss in one, more or all cameras, interdependencies in the camera's information contribution, and variable action costs.

We evaluate our method by comparing it to previous approaches with a pre-recorded sequence of real world images.

## 1 Introduction

This paper describes an enhanced method for selecting a sequence of *optimal sensor actions* for a probabilistic state estimation system. The optimal actions are those that minimize the expected uncertainty of the state probability distribution function, measured by the expected state entropy. We apply this method for view planning in an object tracking task. In this task, the sensor actions affecting the view are the camera zoom settings. However, this method is not restricted to zoom planning. It can also handle other camera actions, such as panning, tilting or translation, and is equally applicable to other active state estimation tasks.

A large amount of research in the area of view planning exists for object recognition tasks [1,2,3], in which the active selection of views directly reduces the uncertainty in classification. For active object tracking many works involve the changing of zoom settings [4,5,6]. However, these methods keep the size of the object in the images constant, as opposed to minimizing the uncertainty of the estimate of the object position. Previous work in uncertainty reduction includes [7], in which a subset from a set of sensors is chosen to meet certain threshold criteria. A more general approach is followed in [8], where actions are chosen which maximally reduce the *expected entropy* of the object position in space as a measure of positional uncertainty.

Previous work [9] has extended this approach to optimize a sequence of actions for view planning. This extension allows variable action costs, such as occur due to limited camera zoom motor speeds, to be incorporated into the optimization. Potential object

---

\* This work was partially funded by the German Science Foundation (DFG) under grant SFB 603/TP B2. Only the authors are responsible for the content.

loss is dealt with by evaluating each possible sequence of object visibility in a *visibility tree* (see section 2 and Fig. 1). A subsequent improving work [10] aimed to address several shortcomings of this method, such as the inability to efficiently handle an object visible in only a subset of the cameras, with the use of the *sequential Kalman filter*. By applying a visibility tree to each camera separately, the computational cost is linear in the number of cameras, and partial visibility can be handled.

The main problem with the still remaining visibility tree is that its size, and therefore the computation costs for view planning, are still exponential in the number of time steps. In this work, we propose a new method, which flattens this visibility tree, thus achieving linear runtime.

We test our approach with a prerecorded image sequence from up to three cameras. This sequence is scaled with a variable scale factor to simulate a changing focal length, while allowing several algorithms to be compared independently on the same data.

This paper is organized as follows: The next section reviews the current state of view planning for active object tracking and describes the notation used in this work. Section 3 details the method of visibility tree linearization to reduce computation time. Section 4 covers the experiments, comparing the previous methods to our new one. The last section summarizes and concludes this paper, and lists potential future work.

## 2 Kalman Filter and Action Selection

Tracking an object in 3D is defined as a state estimation problem, which we solve with the well-known Kalman filter [11], extended to handle sensor actions. To accommodate the non-linear nature of the observation functions involved, we use the *extended Kalman filter* [12,13], although this distinction is not relevant for this work.

The (extended) Kalman filter estimates the state of a discrete-time dynamic system. At time  $t$ , the state is described in the state vector  $\mathbf{q}_t \in \mathbb{R}^n$ . The cameras generate an observation  $\mathbf{o}_t \in \mathbb{R}^m$  from the state. The state change and observation equations are

$$\mathbf{q}_t = \mathbf{f}(\mathbf{q}_{t-1}) + \mathbf{w} \quad , \quad \mathbf{o}_t = \mathbf{h}(\mathbf{q}_t, \mathbf{a}_t) + \mathbf{r} \quad (1)$$

where  $\mathbf{f}(\cdot) \in \mathbb{R}^n$  is the state transition function and  $\mathbf{h}(\cdot, \cdot) \in \mathbb{R}^m$  the observation function, based on the cameras' projection function.  $\mathbf{w}$  and  $\mathbf{r}$  are normal error processes with zero mean and covariance matrices  $\mathbf{W}$  and  $\mathbf{R}$ .

For active object tracking, the observation function also contains an *action* parameter  $\mathbf{a}_t \in \mathbb{R}^l$ , which combines all influences on the observation process, such as zooming, panning, tilting, or translating the camera. For this work, we focus on zoom planning as the camera action. The action is performed *before* an observation is made.

Given the noise terms, the state must be estimated each time step. Specifically, we must calculate the state probability distribution  $p(\mathbf{q}_t | \langle \mathbf{o} \rangle_t, \langle \mathbf{a} \rangle_t)$ , given the sequences of all observations  $\langle \mathbf{o} \rangle_t$  and all actions  $\langle \mathbf{a} \rangle_t$  taken up to, and including, time  $t$ . Within the Kalman filter framework, this distribution is assumed to be a normal, or Gaussian, distribution.

We use the following Kalman filter notation:  $\hat{\mathbf{q}}_t^-$  and  $\hat{\mathbf{q}}_t^+$  are the *a priori* and *a posteriori* state estimate means at time  $t$ .  $\mathbf{P}_t^-$  and  $\mathbf{P}_t^+$  are the covariance matrices for the

errors of the *a priori* and *a posteriori* state estimates. The extended Kalman filter performs the following steps for each time-step  $t$ :

1. Prediction of the state mean  $\hat{\mathbf{q}}_t^-$  and covariance  $\mathbf{P}_t^-$ :

$$\hat{\mathbf{q}}_t^- = \mathbf{f}(\hat{\mathbf{q}}_{t-1}^+) \quad , \quad \mathbf{P}_t^- = \mathbf{F}_t \mathbf{P}_{t-1} \mathbf{F}_t^T + \mathbf{W} \quad (2)$$

2. Computation of the filter gain  $\mathbf{K}_t$ :

$$\mathbf{K}_t = \mathbf{P}_t^- \mathbf{H}_t^T(\mathbf{a}_t) (\mathbf{H}_t(\mathbf{a}_t) \mathbf{P}_t^- \mathbf{H}_t^T(\mathbf{a}_t) + \mathbf{R})^{-1} \quad (3)$$

3. State update with the observation  $\mathbf{o}_t$ :

$$\hat{\mathbf{q}}_t^+ = \hat{\mathbf{q}}_t^- + \mathbf{K}_t \left( \mathbf{o}_t - \mathbf{h}(\hat{\mathbf{q}}_t^-, \mathbf{a}_t) \right) \quad , \quad \mathbf{P}_t^+(\mathbf{a}_t) = (\mathbf{I} - \mathbf{K}_t \mathbf{H}_t(\mathbf{a}_t)) \mathbf{P}_t^- \quad (4)$$

$\mathbf{F}_t$  and  $\mathbf{H}_t(\mathbf{a}_t)$  denote the Jacobians of  $\mathbf{f}(\cdot)$  and  $\mathbf{h}(\cdot, \cdot)$  at  $\hat{\mathbf{q}}_{t-1}^+$  and  $\hat{\mathbf{q}}_t^-$  respectively, to account for non-linear functions. Since  $\mathbf{H}_t(\mathbf{a}_t)$  depends on the selected action  $\mathbf{a}_t$ , the *a posteriori* state covariance  $\mathbf{P}_t^+$  does, too. If no valid observation  $\mathbf{o}_t$  is made at time  $t$ , the update step cannot be performed and  $\hat{\mathbf{q}}_t^+, \mathbf{P}_t^+$  are equal to  $\hat{\mathbf{q}}_t^-, \mathbf{P}_t^-$ .

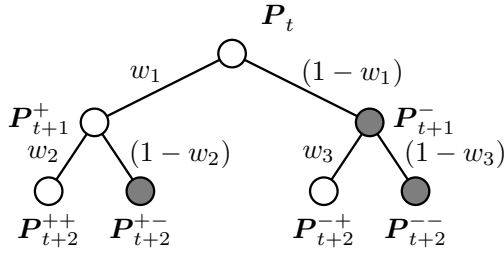
Since we are interested in obtaining the most information about the state, we need to determine the optimal action  $\mathbf{a}_t^*$  where the uncertainty is lowest. In [8], this is achieved by finding the action where the *entropy* of the *a posteriori* probability distribution  $p(\mathbf{q}_t | \langle \mathbf{o} \rangle_t, \langle \mathbf{a} \rangle_t)$  is minimal. As this is a normal distribution  $\mathcal{N}(\hat{\mathbf{q}}_t^+, \mathbf{P}_t^+)$ , the entropy is equal to  $\log(|\mathbf{P}_t^+|)$  up to constant terms and factors. These constants can be ignored during optimization. The entropy depends on the covariance  $\mathbf{P}_t^+$ , and therefore on  $\mathbf{a}_t$ , but *not* on  $\mathbf{o}_t$ . This allows to determine  $\mathbf{a}_t^*$  *before* making an observation.

The problem of visibility in object tracking is also addressed in [8]. An observation  $\mathbf{o}_t$ , containing the position of the object being tracked in each camera, is only valid if the object is in *every* camera's field of view. We refer to an  $\mathbf{o}_t \in \mathbb{R}^m$  lying outside of the field of view as a *non-visible observation*. The probability  $w$  that the object lies in the field of view of all cameras can be calculated from the predicted observation for any action  $\mathbf{a}_t$  by integrating the probability density of the observation over the camera sensor. The expected entropy for an action is then the weighted combination of the entropies for each case of visibility, or for optimization purposes

$$\mathbf{a}_t^* = \arg \min_{\mathbf{a}_t} (w \cdot \log(|\mathbf{P}_t^+(\mathbf{a}_t)|) + (1 - w) \cdot \log(|\mathbf{P}_t^-|)) \quad (5)$$

This action selection has been extended to a sequence of future actions in [9]. For a sequence,  $w$  is extended to a *visibility tree*, which is a binary tree in which each branching represents a visible or non-visible outcome. The entropy for each possible sequence of visible or non-visible observations is calculated and then summed up by walking up the tree again.

An example of such a tree for two time steps is shown in Fig. 1. In this example, each node represents one of the two possible *a posteriori* covariance matrices. Light nodes are the predicted result of a visible observation, dark nodes of a non-visible one. In each time step, the probabilities of visibility or non-visibility are given by the  $w$  and  $(1 - w)$  terms (note that  $w_2 \neq w_3$ ). The total expected entropy is the weighted sum of the



**Fig. 1.** The visibility tree for two time steps. The calculation starts at the top at time  $t$ . The nodes represent possible *a posteriori* covariance matrices for subsequent time steps. Light nodes are the result of a visible observation, dark nodes of a non-visible one. See the text for a deeper discussion of the visibility tree.

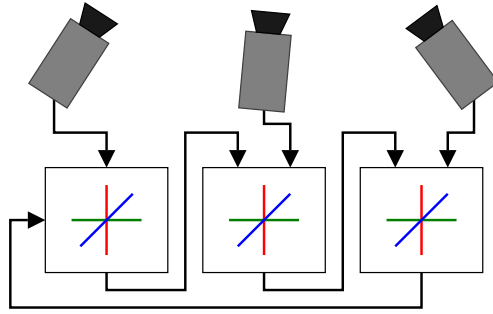
four entropies based on the four final covariances. The covariances are tagged with the visibility sequence they resulted from. For example,  $P_{t+2}^{+-}$  is obtained by first assuming a visible observation (+) followed by a non-visible one (-). The final expected entropy for this example is (ignoring constant terms):

$$w_1 w_2 \log(|P_{t+2}^{++}|) + w_1 (1 - w_2) \log(|P_{t+2}^{+-}|) \\ + (1 - w_1) w_3 \log(|P_{t+2}^{-+}|) + (1 - w_1) (1 - w_3) \log(|P_{t+2}^{--}|)$$

This action selection can also be performed with the *sequential Kalman filter* [12]. The sequential Kalman filter is a sequential evaluation method of the standard Kalman filter, in which the sensors are processed in sequence. This method is equivalent to providing each sensor with its private Kalman filter and can be used when the observation noise for each sensor is uncorrelated. The *a posteriori* distribution of one sensor's filter is used as the *a priori* distribution for the next. Fig. 2 gives an overview of the sequential state estimation process. The advantage of the sequential Kalman method is that the visibility is no longer determined by the object being in the field of view of all cameras; partially visible observations can also be handled by skipping a camera's filter if the object is not visible.

The disadvantages are that the sensor noise must be uncorrelated between sensors for the sequential Kalman filter, and that the result may depend on the order in which the sensors are processed. While the traditional Kalman filter with linear prediction and observation models does not depend on the order of the sensors, the extended Kalman filter obtains the Jacobians  $F_t$  and  $H_t(a_t)$  by deriving at the current state estimate. Since this state estimate is affected by the observations from previous sensors, the Jacobians will differ if different sensors are processed beforehand. However, this difference is comparable to the differences encountered in the Jacobians in the non-sequential extended Kalman filter, where the Taylor expansion is also performed on the current best estimate instead of the true state, and is usually ignored.

The sequential Kalman filter is used for multi-step action selection in [10]. Each camera action is optimized independently by the method of [9], on the assumption that changing the zoom level in one camera will not influence the information gained in



**Fig. 2.** The sequential Kalman filter. Each camera adds its observation to the state estimate in sequence. The *a posteriori* state estimate of the last camera is transformed to the *a priori* estimate of the first camera on the next time step.

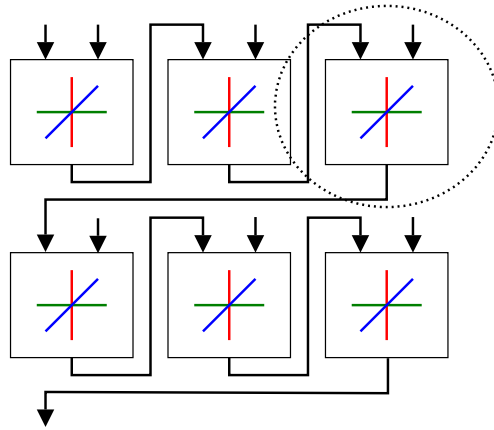
another camera. This sequential multi-step action selection still uses the visibility tree from the original multi-step method. Partial visibility (in some, but not all cameras) is handled explicitly during tracking and implicitly in the optimization process.

However, this gives rise to another problem. During the action planning step, the predicted uncertainty is calculated by the contribution of a single camera, disregarding the others. In the Kalman filter's update step (equations (3) and (4)),  $P_t^+$  is derived from  $P_t^-$  using only the observation function of this single camera. This leads to an overestimation of the *a posteriori* covariance in the planning phase, which results in an overly cautious action planning. This omission can be rectified by including the effects of the other cameras on  $P_t^+$ . However, to avoid another visibility dependency and keep the visibility tree small, these other cameras must follow actions which are assumed to guarantee an observation, which still overestimates the covariance during planning.

### 3 Linearization of the Visibility Tree

For the multi-step multi-camera sequential Kalman filter, as seen in Fig. 3, the output of each individual filter during tracking (such as the one marked in the figure) becomes the input of the next one. This output is the probability density of the state estimate at this time, with the observation of this camera embedded if it was visible, and skipped if it was not. For view planning, this means that each individual filter has *two* possible outputs which need to be considered, with covariance matrices  $P_t^+$  and  $P_t^-$ , since the expected state mean is the same in both cases.

The previous methods have handled these with a visibility tree, as detailed in the last section. Spanning a visibility tree for the full sequential filter is prohibitive, since the size of the tree is exponential in the number of cameras and time steps. The solution which uses the sequential Kalman filter reduces this complexity somewhat by optimizing actions for each camera separately. However, the visibility tree size is still exponential in the number of time steps, and the expected covariance is overestimated, as mentioned previously.



**Fig. 3.** The sequential Kalman filter during multi-step evaluation. The output of one individual filter becomes the input of the next filter in the same time step or the first filter in the next time step (after the state transformation, not shown here). The dashed circle marks one individual filter.

The visibility tree can be flattened by closely looking at the two probability distributions that can result in one time step. The two resulting distributions are Gaussian distributions and differ only in their covariances,  $\mathbf{P}_t^+$  and  $\mathbf{P}_t^-$ , but not in their means, as the expected mean does not depend on the visibility in the view planning step. Since we know the probability  $w \in [0, 1]$  that one of these two distributions will be the actual output, we can consider them to be two components of a mixture distribution  $\mathcal{M}$ ,

$$\mathcal{M} = w \cdot \mathcal{N}(\hat{\mathbf{q}}_t^+, \mathbf{P}_t^+) + (1 - w) \cdot \mathcal{N}(\hat{\mathbf{q}}_t^+, \mathbf{P}_t^-), \quad (6)$$

which describes the expected distribution of the state after performing action  $\mathbf{a}_t$ . Since this is an unimodal distribution, we can approximate it by a new Gaussian distribution with the same covariance. As known in statistics, the covariance matrix of  $\mathcal{M}$  is:

$$\mathbf{P}_t^\circ = w \cdot \mathbf{P}_t^+ + (1 - w) \cdot \mathbf{P}_t^- \quad (7)$$

Therefore, our approximating Gaussian is of the form  $\mathcal{N}(\hat{\mathbf{q}}_t^+, \mathbf{P}_t^\circ)$ .

This distribution can now be used as an estimate of the resulting state probability distribution after visibility is considered. Note that the Gaussian distribution is an approximation of the mixture distribution, with same mean and covariance, but with different density functions.

The benefits of this approach are obvious. Since each individual filter in a multi-step multi-camera now only results in a single output distribution *during view planning as well*, the effects of an action can now be calculated in linear time in the number of cameras times the number of time steps. This can be seen in Fig. 3, which is now equally valid for the view planning process. Since the actions are optimized for all cameras at the same time, this approach also fully handles dependencies in the actions of different cameras, unlike the previous sequential method which used a separate optimization.



**Fig. 4.** The views of the test setup from all three cameras at the same time. The colored bottle is tracked as it turns on the turntable. The calibration pattern is used to initially calibrate all cameras, it is not used during tracking.

Although the entropy of the final expected distribution, based on  $\log(|P_{t+k}^\circ|)$ , differs from the actual expected entropy, the *behavior* of the system is close enough such that the optimal action can be searched for. This can be seen in the next section, where several approaches are compared on the same data. The behaviour is visible when comparing the original single-step approach to the one based on  $P_{t+k}^\circ$ : both approaches are very similar when no visibility problems are encountered.

## 4 Experiments

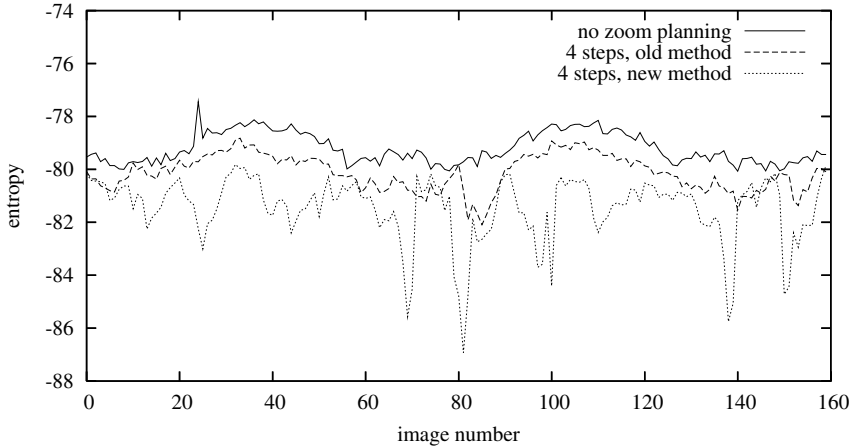
We test our new method on a recorded video sequence, shown in Fig. 4. Three cameras take a high-resolution image of the scene, consisting of an object moved by a turntable. The cameras are calibrated to a global coordinate system with the calibration pattern, which is not used for the tracking process. The object is tracked with a color histogram tracker [14].

The prerecorded images allow several view planning methods to be compared on the same data. However, this precludes the effect of the camera zoom on these images, unless we simulate this zoom on the original images. The original images are 640 by 480 pixels, but the tracking process uses images of size 320 by 240 pixels. To obtain this size reduction, and simulate the camera zoom, we scale and crop the original image by an amount which depends on the associated zoom level. When fully zoomed in, the transformation only crops a 320 by 240 pixel image from the center of the original. As the zoom level decreases, the cropped region becomes larger and is subsequently scaled to the correct size. When fully zoomed out, the original images are only scaled, no cropping occurs. Using such a reduced image size ensures that, even when fully zoomed in, no upsampling artifacts occur.

The advantages of multi-step view planning have been detailed in [9] and [10]; no detailed comparison to single-step planning will be made here. We will focus primarily on a direct comparison between the previous planning method (cf. section 2), which used separate optimization, and the newly proposed one (cf. section 3).

We test both systems on the same data, as detailed above. Each planning system recommends the next view for the tracking system in the form of a set of actions. The optimal action set is planned with the global optimization technique of Adaptive Ran-





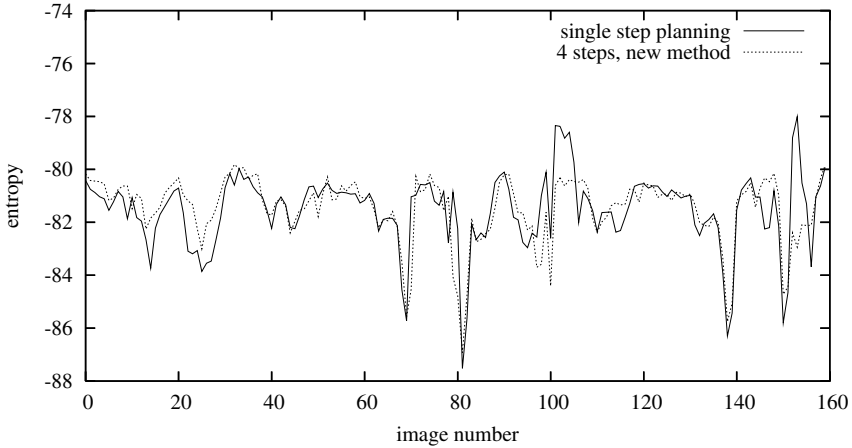
**Fig. 5.** The entropy as  $\log(|\mathbf{P}_t^+|)$  with three cameras plotted for image number  $t$ . Shown are the plots for no planning, planning 4 steps ahead using the old, independently optimizing method, and the new proposed method.

dom Search [15], evaluating a total of 400 separate action sequences per time frame. Each action sequence contains the next actions for each camera for the next time steps. In our experiments, we planned one to four steps ahead with up to three cameras, so each action sequence contains up to 12 separate zoom settings.

Fig. 5 shows the entropy of the object position during the tracking process by measuring  $\log(|\mathbf{P}_t^+|)$ . Note that  $\mathbf{P}_t^+$  is the result of the actual tracking phase, and not a result from the view planning. However, the view planning influences the tracking results both positively (by providing zoomed in views, reducing the entropy) and negatively (by zooming in too far, causing object loss and raising the entropy). This experiment uses three cameras. The plots show the results for unplanned tracking, planning 4 steps with the independent optimization, and 4 steps with the new method.

Both planning methods result in an uncertainty, measured by the entropy, which is lower than when no zoom planning is used. But it can clearly be seen that the original approach with separate optimization still results in a higher uncertainty than the new approach. Since the expected *a posteriori* covariance is overestimated, the actions planned by this system are not as aggressive as those calculated with the new system. The new system plans views which are zoomed further in, which lowers the entropy, in many cases by quite a large amount. Only in a few cases (near image numbers 72 and 142) do these zoom levels prove overconfident, resulting in short object losses and a higher entropy than the original approach.

Fig. 6 compares the new multi-step approach, looking 4 time steps into the future, to the single step approach. The experiments are the same as in Fig. 5. Both plots are very similar, showing that the behaviour of the view planning using the combined covariance  $\mathbf{P}_{t+k}^\circ$  is very close to the original behaviour, if at times slightly worse due to the more cautious approach of multi-step planning. The most notable differences occur around image numbers 101 and 153 in the right half of the plot. The object starts moving out of



**Fig. 6.** The entropy as  $\log(|P_t^+|)$  with three cameras plotted for image number  $t$ . The single-step planning method and the new proposed multi-step method for 4 time steps are compared.

**Table 1.** Computation time. Shown are the average computation times for evaluating a single action sequence for one to four time steps, two and three cameras, with the old and the new method. All times in ms.

Time steps planned	old method		new method	
	two cameras	three cameras	two cameras	three cameras
1 step	0.115	0.173	0.095	0.124
2 steps	0.390	0.597	0.163	0.224
3 steps	0.945	1.457	0.247	0.333
4 steps	2.055	3.022	0.293	0.431

the field of view of one or even several cameras. The single step planning is caught off guard by this, resulting in object loss and large spikes in the uncertainty. The multi-step approach is able to predict the object loss better and avoids these spikes.

Another important aspect is the comparison of running times. The running times per frame for several different cases are given in table 1. Note that while the original algorithm required exponential time per frame (yet was linear in the number of cameras due to the independent treatment), the new approach is about linear in the number of time steps as well. All times are in milliseconds on a Pentium IV processor at 2.66 GHz.

## 5 Conclusion

We have presented a new approach for multi-step multi-camera view planning for object tracking, based on the method of entropy minimization. This approach runs in linear time in the number of cameras and time steps. It can incorporate action costs through the evaluation of several time steps into the future. It is capable of handling a variable

number of cameras, partial visibility, and interdependence in the camera actions. The general nature of this approach allows it to be applied to a wide variety of active state estimation problems outside of visual object tracking.

Additional work will focus on expanding the action space to also allow camera pan and tilt motions. Another topic is the combination of view planning for tracking with view planning for other tasks, such as object reconstruction or object recognition.

## References

1. Paletta, L., Pinz, A.: Active object recognition by view integration and reinforcement learning. *Robotics and Autonomous Systems* **31**, Issues 1-2 (2000) 71–86
2. Denzler, J., Brown, C.: Information Theoretic Sensor Data Selection for Active Object Recognition and State Estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24** (2002) 145–157
3. Deinzer, F., Denzler, J., Niemann, H.: Viewpoint Selection – Planning Optimal Sequences of Views for Object Recognition. In: *Computer Analysis of Images and Patterns – CAIP 2003*. LNCS 2756, Heidelberg (2003) 65–73
4. Fayman, J., Sudarsky, O., Rivlin, E., Rudzsky, M.: Zoom tracking and its applications. *Machine Vision and Applications* **13** (2001) 25–37
5. Tordoff, B., Murray, D.: Reactive zoom control while tracking using an affine camera. In: *Proc 12th British Machine Vision Conference*, September 2001. Volume 1. (2001) 53–62
6. Micheloni, C., Foresti, G.L.: Zoom on Target While Tracking. In: *Proceedings of the International Conference on Image Processing*. Volume 3., Genua, Italy (2005) 117–120
7. Kalandros, M.K., Pao, L.Y., Ho, Y.: Randomization and super-heuristics in choosing sensor sets in target tracking applications. In: *Proc. IEEE Conf. Decision and Control*, Phoenix, AZ (1999) 1803–1808
8. Denzler, J., Zobel, M., Niemann, H.: Information Theoretic Focal Length Selection for Real-Time Active 3-D Object Tracking. In: *International Conference on Computer Vision*, Nice, France (2003) 400–407
9. Deutsch, B., Zobel, M., Denzler, J., Niemann, H.: Multi-Step Entropy Based Sensor Control for Visual Object Tracking. In: *Pattern Recognition, 26th DAGM Symposium*, Tübingen, Germany (2004) 359–366
10. Deutsch, B., Deinzer, F., Zobel, M., Denzler, J.: Multi-Step Active Object Tracking with Entropy Based Optimal Actions Using the Sequential Kalman Filter. In Araújo, H., Vieira, A., Braz, J., Encarnação, B., Carvalho, M., eds.: *Proceedings of the International Conference on Image Processing*. Volume 2., Setúbal, Portugal, INSTICC Press, Setúbal, Portugal (2005) 169–176
11. Kalman, R.: A new approach to linear filtering and prediction problems. *Journal of Basic Engineering* (1960) 35–44
12. Chui, C.K., Chen, G.: *Kalman Filtering*. Springer, Heidelberg (1991)
13. Bar-Shalom, Y., Fortmann, T.: *Tracking and Data Association*. Academic Press, Boston, San Diego, New York (1988)
14. Pérez, P., Hue, C., Vermaak, J., Gangnet, M.: Color-based probabilistic tracking. In: *Proceedings of the European Conference on Computer Vision 2002*, Copenhagen, Springer, Heidelberg (2002) 661–675
15. Törn, A., Žilinskas, A.: *Global Optimization*. Volume 350 of *Lecture Notes in Computer Science*. Springer, Heidelberg (1987)

# Nonparametric Density Estimation for Human Pose Tracking<sup>\*</sup>

Thomas Brox<sup>1</sup>, Bodo Rosenhahn<sup>2</sup>, Uwe G. Kersting<sup>3</sup>, and Daniel Cremers<sup>1</sup>

<sup>1</sup> CVPR Group, University of Bonn  
Römerstr. 164, 53117 Bonn, Germany  
{brox, dcremers}@cs.uni-bonn.de

<sup>2</sup> MPI for Computer Science,  
Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany  
rosenhahn@mpi-inf.mpg.de

<sup>3</sup> Department of Sport and Exercise Science  
The University of Auckland, New Zealand

**Abstract.** The present paper considers the supplement of prior knowledge about joint angle configurations in the scope of 3-D human pose tracking. Training samples obtained from an industrial marker based tracking system are used for a nonparametric Parzen density estimation in the 12-dimensional joint configuration space. These learned probability densities constrain the image-driven joint angle estimates by drawing solutions towards familiar configurations. This prevents the method from producing unrealistic pose estimates due to unreliable image cues. Experiments on sequences with a human leg model reveal a considerably increased robustness, particularly in the presence of disturbed images and occlusions.

## 1 Introduction

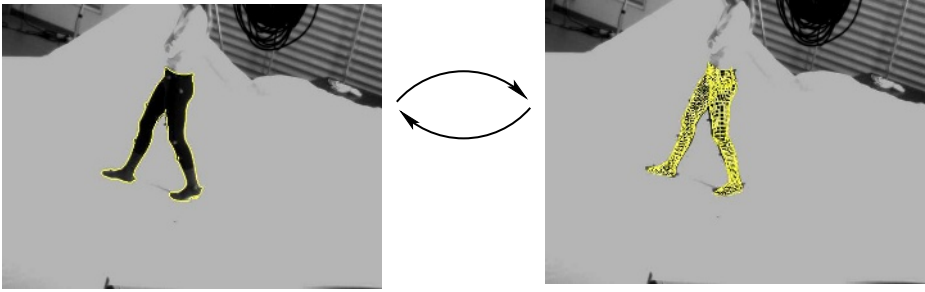
This paper is concerned with the task of human pose tracking, also known as motion capturing (MoCap). It is a subtopic of pose tracking where the object/body consists of multiple parts, i.e. limbs, constrained by a kinematic chain [2]. The goal of pose estimation then is to determine the 3-D rigid body motion as well as the joint angles in the kinematic chain.

There are basically two ways to approach the problem. In the *discriminative* approach, one extracts some basic features from the image(s), the raw pixels in the simplest case, and directly learns a mapping from these observed features to the set of pose parameters from a large set of training data. Hence, the method does not care about the meaning of intermediate states, but solely acts as kind of a black box that yields a certain output given a certain input. A recent representative of the discriminative approach is the work in [1].

The *generative* approach, on the other hand, is model based, i.e., there is a more or less detailed object model that, for a given pose, can approximately generate the images

---

<sup>\*</sup> We gratefully acknowledge funding by the DFG project CR250/1 and the Max-Planck Center for visual computing and communication.



**Fig. 1.** The MoCap system in [10]: **Left:** The object contours are extracted in the input images (just one frame is shown). **Right:** These are used for pose estimation. The pose result is applied as shape prior for the segmentation process and the process is iterated. The goal of the present paper is to extend this system to situations with heavily disturbed image data by supplementing a prior on the pose configuration.

that are seen by the camera. The pose parameters are optimized in such a way that the model optimally explains the images.

This paper builds upon a generative approach presented in [3] for rigid bodies with a free-form surface model given. The technique has been extended in [10] to kinematic chains. It determines the pose parameters by matching the projected surface to the object contours in the images. These contours are extracted by assuming a local Gaussian distribution of the object and background region and taking the projected surface model as shape prior into account.

Generative approaches like the one in [3,10] can be described by Bayesian inference:

$$p(\chi, C|I) = \frac{p(I|C, \chi)p(C|\chi)p(\chi)}{p(I)} \quad (1)$$

where  $\chi$  denotes the sought pose parameters,  $I$  the input image, and  $C$  the object contour that is obtained together with the pose parameters. The technique in [10] uses a prior on the contour by means of  $p(C|\chi)$ , yet  $p(\chi)$  has been ignored by assuming a uniform prior.

The goal of the present paper is to integrate such prior knowledge about the probability of pose configurations into generative approaches like the one in [10]. This is achieved by learning a probability density from training samples. To cope with the non-Gaussian nature of the configuration space, we suggest the approximation of the density by a nonparametric kernel density estimate. Such density estimates have been used in computer vision in the context of image segmentation [6,12,5] and shape priors [4].

While learning from training samples is a prerequisite in many discriminative approaches [13,1] and well-known in the context of shape priors [17,18,10,4], there is very few work with regard to prior knowledge in the context of 3-D generative models apart from the introduction of hard constraints such as explicit joint angle limits or prevention of self-intersections [16]. In [15] it has been suggested to learn a Gaussian mixture in a previously reduced space. Like the nonparametric density estimates suggested here, this work aims at capturing the complex, non-Gaussian configuration space of human pose.

Our experiments with a leg model having 6+12 degrees of freedom show a considerably increasing robustness when the prior is involved. Particularly in cases where the images yield few and unreliable information due to occlusion or noise, the prior helps to keep the result close to familiar pose configurations.

**Paper organization.** In the next section, we briefly review the technique described in [3,10] which yields the image-driven part of the pose estimates. After that, Section 3 introduces the modeling of the pose prior, motivates the choice of a kernel density estimate, and demonstrates the integration of the prior into the numerical optimization scheme. In Section 4, we show the effect of the prior and compare the quality of the results to pose estimates obtained with an industrial marker based tracking system. The paper is concluded by a brief summary.

## 2 Image-Driven Pose Tracking

### 2.1 Pose and Joints

To represent rigid body motions, we use the exponential form,

$$\mathbf{M} = \exp(\theta \hat{\xi}) = \exp \begin{pmatrix} \hat{\omega} & \mathbf{v} \\ 0_{3 \times 1} & 0 \end{pmatrix}. \quad (2)$$

The matrix  $\theta \hat{\xi}$  in the exponent is called a *twist*, which consists of two components, a  $3 \times 3$  matrix  $\hat{\omega}$  and a 3-D vector  $\mathbf{v}$ . The matrix  $\hat{\omega}$  is restricted to be skew-symmetric, which means  $\hat{\omega} \in so(3)$ , with  $so(3) = \{\mathbf{A} \in \mathbb{R}^{3 \times 3} | \mathbf{A} = -\mathbf{A}^T\}$ . The exponent of such a twist results in a rigid body motion [7], which is given as a screw motion with respect to a velocity  $\theta$ . It is common to represent the components of a twist as a 6-D vector  $\xi = (\omega_1, \omega_2, \omega_3, \mathbf{v})^T$ . Twists have two advantages: firstly, they can easily be linearized and used in a fixed point iteration scheme for pose estimation [11]. Secondly, restricted screws (with no pitch component) can be employed to model joints. A kinematic chain is modeled as the consecutive evaluation of such exponential functions, i.e., a point at an endeffector, transformed by a rigid body motion is given as

$$X'_i = \exp(\theta \hat{\xi})(\exp(\theta_1 \hat{\xi}_1) \dots \exp(\theta_n \hat{\xi}_n)) X_i \quad (3)$$

For abbreviation, we will in the remainder of this paper note a pose configuration by the  $(6+n)$ -D vector  $\chi = (\xi, \theta_1, \dots, \theta_n) = (\xi, \Theta)$  consisting of the 6 degrees of freedom for the rigid body motion  $\xi$  and the joint angles  $\Theta$ . During optimization there is need to generate a transformation matrix from a twist and, vice-versa, to extract a twist from a given matrix. Both can be done efficiently by applying the Rodriguez formula, see [7] for details.

### 2.2 Model

Coupled extraction of the object contours and registration of the model to these contours can be described by minimization of an energy functional that contains both the pose parameters  $\chi$  and the object contour as unknowns [3]:

$$\begin{aligned}
 E(\chi, \Phi) = & - \int_{\Omega} H(\Phi) \log p_1 + (1 - H(\Phi)) \log p_2 \, dx \\
 & + \nu \int_{\Omega} |\nabla H(\Phi)| \, dx + \lambda \int_{\Omega} (\Phi - \Phi_0(\chi))^2 \, dx.
 \end{aligned}
 \tag{4}$$

The contour is represented as the zero-level line of a level set function  $\Phi : \Omega \rightarrow \mathbb{R}$ , such that one can access the interior and exterior of the object region via the step function  $H(s)$ . Object and background are described by the probability densities  $p_1$  and  $p_2$ , respectively. They are modeled by local Gaussian densities, as described in [3]. Hence, minimizing the first two terms yields a contour that maximizes the total a-posteriori probability of all pixel assignments.

The two remaining terms constitute a prior for the contour. The first term seeks to minimize the length of the contour. The second one depends on the pose parameters and seeks to draw the contour close to the projected surface model  $\Phi_0(\chi)$ . Vice-versa, this term relates the pose parameters to the image data by matching the surface model to the extracted contour, and thereby to the raw pixels. It is a generative model, since given the pose parameters one can use the projected surface  $\Phi_0$  and the region densities  $p_1$  and  $p_2$  to generate a simplified version of the image. The tuning parameters  $\nu = 1.5$  and  $\lambda = 0.05$  have been kept fixed in our experiments.

For  $M > 1$  camera views, which are calibrated with respect to the same world coordinate system, the energy functional can easily be extended to  $M$  views by minimizing the joint energy

$$E(\chi, \Phi_1, \dots, \Phi_M) = \sum_{i=1}^M E(\chi, \Phi_i).
 \tag{5}$$

Whereas the densities  $(p_1)_i$  and  $(p_2)_i$  are independent for each image, the contour extraction is coupled via the pose parameters  $\chi$  that influence the contours due to the shape prior.

### 2.3 Optimization Scheme

We minimize energy (5) by alternating an optimization of the contours for fixed pose parameters and an update of the pose parameters for fixed contours. Keeping the pose parameters fixed yields the gradient descent

$$\partial_i \Phi_i = H'(\Phi_i) \left( \log \frac{(p_1)_i}{(p_2)_i} + \nu \operatorname{div} \left( \frac{\nabla \Phi_i}{|\nabla \Phi_i|} \right) \right) + 2\lambda (\Phi_0(\chi) - \Phi_i).
 \tag{6}$$

Obversely, by keeping the contours fixed, one can derive point correspondences between contour and surface points via shape matching. From the point correspondences, a nonlinear system of equations can be formulated using the twist representation and Clifford algebra. Each point correspondence contributes three equations of rank 2. For details we refer to [10].

The nonlinear system can be solved with a fixed point iteration scheme. Linearizing the equations yields an over-determined linear system of equations, which can be solved with the Householder method in the sense of least squares. Updating the nonlinear system with the new estimates and linearizing again leads to a new linear system. The process is iterated until convergence.

### 3 Constraining the Pose by Kernel Density Estimates

The energy functional from the previous section can be motivated from a probabilistic point of view by considering the a-posteriori probability

$$p(\chi, \Phi | I) \propto p(I | \Phi) p(\Phi | \chi) p(\chi). \quad (7)$$

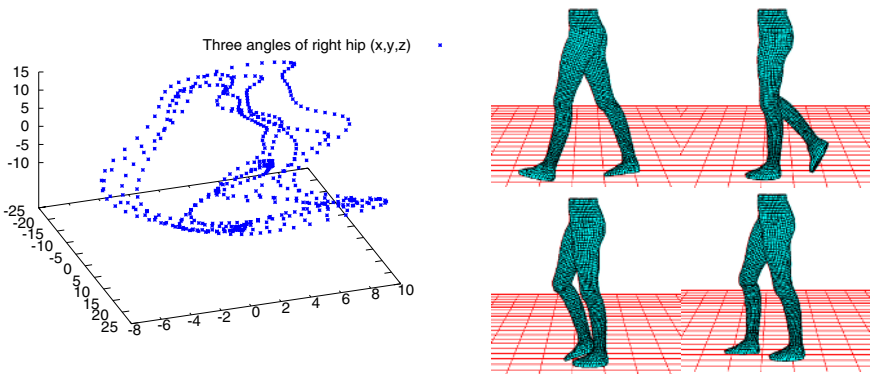
Maximizing this probability is equivalent to minimizing its negative logarithm, which leads to the energy in (4) plus an additional term that constrains the pose to familiar configurations:

$$E_{\text{Prior}} = -\log(p(\chi)). \quad (8)$$

As we want the prior to be independent from the translation and rotation of the body in the training sequences, we apply a uniform prior to the parameters  $\xi$  of the rigid body motion. The remaining probability density for the joint angle configuration  $p(\Theta)$  is supposed to be learned from a set of training samples.

Fig. 2 visualizes the training data consisting of MoCap data from two walking sequences obtained by a marker based tracking system with a total of 480 samples. Only a projection to three dimensions (the three joint angles of the right hip) of the actually 12-dimensional space is shown.

There are many possibilities to model probability densities from such training samples. The most common way is a parametric representation by means of a Gaussian density, which is fully described by the mean and covariance matrix of the training samples. Such representations, however, tend to oversimplify the sample data. Having, for instance, two training samples with the left leg in front and the right leg in back, and vice-versa, a Gaussian density would yield the highest probability for the configuration with both legs in the middle. Configurations close to the samples, on the other hand, would have a comparatively small probability. Although showing only a projection of the full configuration space, Fig. 2 clearly demonstrates that a walking motion cannot



**Fig. 2. Left:** Visualization of the training data obtained from two walking sequences. Only a 3-D projection (the three joint angles of the right hip) of the 12-D space is shown. **Right:** Some training samples applied to the body model.



be described accurately by a Gaussian density. In the 12-D space, this becomes even more obvious.

For this reason, we suggest a nonparametric density estimate by means of the Parzen-Rosenblatt estimator [9,8]. It approximates the probability density by a sum of kernel functions centered at the training samples. A common kernel is the Gaussian function, which leads to:

$$p(\Theta) = \frac{1}{\sqrt{2\pi}\sigma N} \sum_{i=1}^N \exp\left(-\frac{(\Theta_i - \Theta)^2}{2\sigma^2}\right) \quad (9)$$

where  $N$  is the number of training samples  $\Theta_i \in \mathbb{R}^{12}$ . This probability density estimator involves the kernel width  $\sigma$  as a tuning parameter. Whereas small kernel sizes lead to an accurate representation of the training data, the estimated density may not generalize well, i.e., unseen test samples may be assigned a too small probability. Large kernel sizes are more conservative, leading to a smoother approximation of the density, which in the extreme case comes down to a uniform distribution. Numerous works on how to optimally choose the kernel size are available in the statistics literature. A detailed discussion can be found in [14]. In our work, we fix  $\sigma$  as the maximum nearest neighbor distance between all training samples, i.e., the next sample is always within one standard deviation. This ensures a smooth approximation between samples while it still keeps the density model flexible.

Note that (9) does not involve a projection but acts on the full 12-dimensional configuration space of the 12-D joint model. This means, also the interdependency between joint angles is taken into account.

The gradient descent of (8) in  $\Theta$  reads

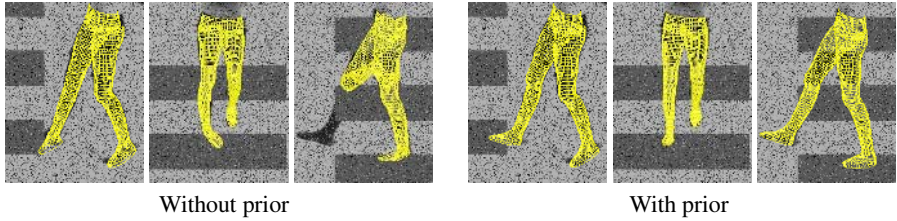
$$\partial_t \Theta = -\frac{\partial E_{\text{Prior}}}{\partial \Theta} = \frac{\sum_{i=1}^N w_i (\Theta_i - \Theta)}{\sigma^2 \sum_{i=1}^N w_i} \quad (10)$$

$$w_i := \exp\left(-\frac{|\Theta_i - \Theta|^2}{2\sigma^2}\right). \quad (11)$$

This can be interpreted as the pose configuration being drawn to the next local maximum of the probability density, i.e., the local mode. We integrate this equation into the linear system of the fixed point iteration scheme from Section 2 by appending for each joint  $j$  an additional equation  $\theta_j^{k+1} = \theta_j^k + \tau \partial_t \theta_j^k$  to the linear system. These equations are weighted by the number of point correspondences in order to achieve an equal weighting between the image- and the prior-driven part. In our experiments, the step size parameter  $\tau = 0.125\sigma^2$  yielded stable results.

In contrast to a simple alternation between the image-driven and the prior-driven part, the integration of (10) into the linear system efficiently allows to compensate discrepancies not only locally in the respective joint angle, but globally in all pose parameters including the overall rigid body motion. Therefore, a large discrepancy in the angle of the leg, for instance, can also be compensated by a rotation of the hip.

A second advantage is the implicit regularization of the equation system. Assume a foot is not visible in any camera view. Without prior knowledge, this would automatically lead to a singular system of equations, since there are no correspondences that generate any constraint equation with respect to the joint angles at the foot. Due to the



**Fig. 3.** Relevance of the learned configurations for the tracking stability. Distracting edges from occlusions locally disturb the image-driven pose estimation. This can finally cause a global tracking failure. The prior couples the body parts and seeks the most familiar configuration given *all* the image data.

interdependency of the joint angles, the prior equation draws the joint angles of the invisible foot to the most probable solution given the angles of the visible body parts.

## 4 Experiments

For the experiments we used a four-camera set-up and grabbed image sequences of a female lower torso. The cameras were calibrated using a calibration cube, synchronized via a genlock interface, and we grabbed with 60 frames per second. The person wore a black leg suit (see Fig. 1).

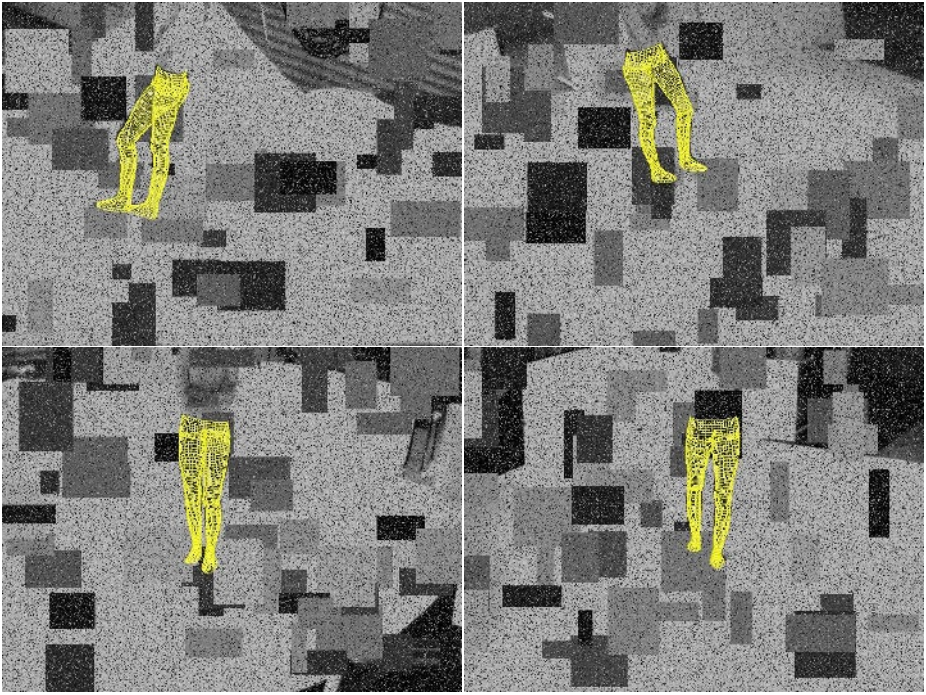
To allow for a quantitative error analysis, we installed parallel to this set-up a second camera ring for a marker based system. Markers were attached to the leg suit and tracked by a commercially available MoCap system<sup>1</sup>. We grabbed a series of sequences and are able to compare our marker-free approach with the marker based system.

Fig. 4 and 5 visualize results of a walking sequence in which we replaced 25% of all pixels by a uniform random value. Additionally, we added heavy occlusions of two different types to all camera views. In the first case, box-shaped occlusions of random size and gray value were randomly distributed across the images. In the second case, we added enduring horizontal stripes to the images. For the last quarter of the sequence, the person is not visible in the first camera anymore. All these difficulties frustrate the acquisition of contour data needed for pose estimation.

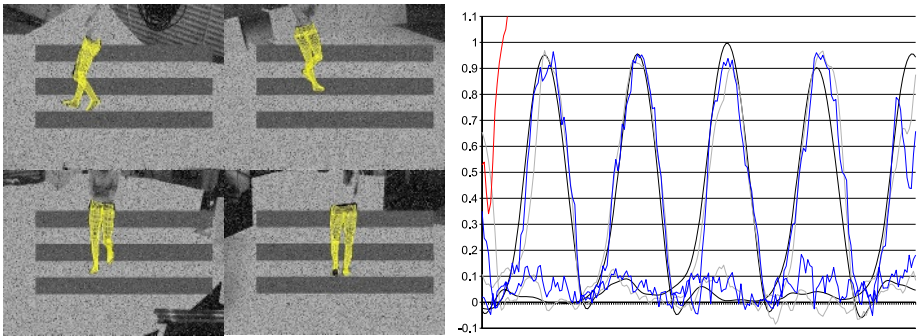
Thanks to the joint angle prior, however, the sequence is tracked reliably in both cases despite these disturbances. The training set did *not* contain the test sequence. The diagram in Fig. 5 compares the obtained tracking curves to the marker based result, which can be regarded as ground truth  $\pm 3$  degree (0.05 radians), and the result obtained when the joint angle prior is ignored. Despite the occlusions, the errors are almost within the accuracy of the marker based system. Without the prior, however, tracking fails nearly right from the beginning.

In order to test the generalization capabilities of the Parzen estimator, we further applied the method to a sequence where the person was asked to perform a series of jumping jacks. Again we added 25% uniform noise to the images. As pose configurations of this type of motion pattern were not contained in the training data, one expects problems

<sup>1</sup> We used the Motion Analysis system with 8 Falcon cameras.

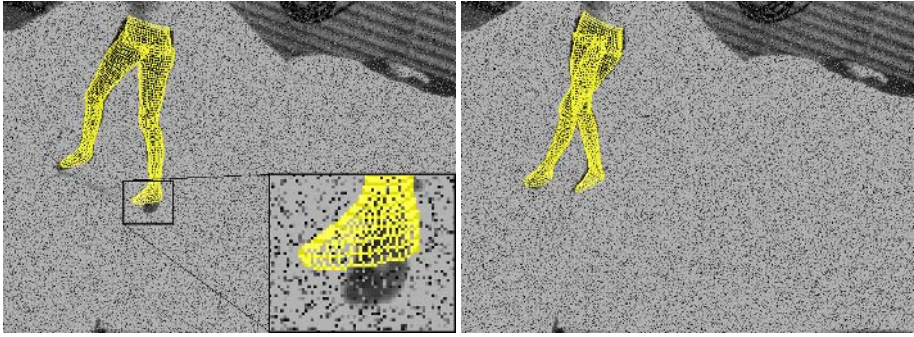


**Fig. 4.** Pose estimates in a sample frame disturbed by 50 varying rectangles with random position, size, and gray value and 25% uncorrelated pixel noise



**Fig. 5. Left:** Pose estimates in a sample disturbed by three enduring gray bars and 25% uncorrelated pixel noise. **Bottom:** Joint angles in radians of the left and right knee, respectively. **Black:** marker based system. **Gray:** occlusion by permanent bars. **Blue:** occlusion by random rectangles (see figure 4). **Red:** tracking without prior fails after a couple of frames.

concerning the accuracy of tracking. Indeed Fig. 6 reveals errors for some frames where the true configuration was too far away from the training samples (enlarged in Fig. 6). Nevertheless, the tracking remains stable and yields sufficiently accurate non-walking



**Fig. 6.** Generalization capabilities of the prior: two frames from a jumping sequence tracked with solely training data from walking sequences available. **Left:** The enlarged part reveals inaccuracies, as the prior prevents the foot angle from further bending. **Right:** However, the prior is able to accurately handle many other configurations not consistent with those of a walking person.

configurations. For all experiments we used the same internal parameters. Computation takes, like in the method without a pose prior, around 1 minute per frame in the four-camera setup.

## 5 Summary

We have suggested to learn joint angle configurations from training samples via a Parzen density estimator and to integrate this prior via Bayesian inference into a numerical scheme for contour based human pose tracking from multiple views. The learned density draws the solution towards familiar configurations given the available data from the images. In case the image does not provide enough information for a unique solution, the most probable solution according to the prior is preferred. The experimental evaluation demonstrates that this allows to handle situations with seriously disturbed images where tracking without knowledge about reasonable angle configurations is likely to fail.

## References

1. A. Agarwal and B. Triggs. Recovering 3D human pose from monocular images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(1):44–58, Jan. 2006.
2. C. Bregler, J. Malik, and K. Pullen. Twist based acquisition and tracking of animal and human kinematics. *International Journal of Computer Vision*, 56(3):179–194, 2004.
3. T. Brox, B. Rosenhahn, and J. Weickert. Three-dimensional shape knowledge for joint image segmentation and pose estimation. In W. Kropatsch, R. Sablatnig, and A. Hanbury, editors, *Pattern Recognition*, volume 3663 of *LNCS*, pages 109–116. Springer, Aug. 2005.
4. D. Cremers, S. Osher, and S. Soatto. Kernel density estimation and intrinsic alignment for shape priors in level set segmentation. *International Journal of Computer Vision*. To appear.
5. T. Kadir and M. Brady. Unsupervised non-parametric region segmentation using level sets. In *Proc. Ninth IEEE International Conference on Computer Vision*, volume 2, pages 1267–1274, 2003.

6. J. Kim, J. Fisher, A. Yezzi, M. Cetin, and A. Willsky. A nonparametric statistical method for image segmentation using information theory and curve evolution. *IEEE Transactions on Image Processing*, 14(10):1486–1502, 2005.
7. R. Murray, Z. Li, and S. Sastry. *Mathematical Introduction to Robotic Manipulation*. CRC Press, Baton Rouge, 1994.
8. E. Parzen. On the estimation of a probability density function and the mode. *Annals of Mathematical Statistics*, 33:1065–1076, 1962.
9. F. Rosenblatt. Remarks on some nonparametric estimates of a density function. *Annals of Mathematical Statistics*, 27:832–837, 1956.
10. B. Rosenhahn, T. Brox, U. Kersting, A. Smith, J. Gurney, and R. Klette. A system for marker-less motion capture. *Künstliche Intelligenz*, (1):45–51, 2006.
11. B. Rosenhahn and G. Sommer. Pose estimation of free-form objects. In T. Pajdla and J. Matas, editors, *Proc. 8th European Conference on Computer Vision*, volume 3021 of *LNCS*, pages 414–427. Springer, May 2004.
12. M. Rousson, T. Brox, and R. Deriche. Active unsupervised texture segmentation on a diffusion based feature space. In *Proc. International Conference on Computer Vision and Pattern Recognition*, pages 699–704, Madison, WI, June 2003.
13. G. Shakhnarovich, P. Viola, and T. Darell. Fast pose estimation with parameter sensitive hashing. In *Proc. International Conference on Computer Vision*, pages 750–757, Nice, France, Oct. 2003.
14. B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, New York, 1986.
15. C. Sminchisescu and A. Jepson. Generative modeling for continuous non-linearly embedded visual inference. In *Proc. International Conference on Machine Learning*, 2004.
16. C. Sminchisescu and B. Triggs. Estimating articulated human motion with covariance scaled sampling. *International Journal of Robotics Research*, 22(6):371–391, 2003.
17. J. Zhang, R. Collins, and Y. Liu. Representation and matching of articulated shapes. In *Proc. International Conference on Computer Vision and Pattern Recognition*, pages 342–349, June 2004.
18. J. Zhang, R. Collins, and Y. Liu. Bayesian body localization using mixture of nonlinear shape models. In *Proc. International Conference on Computer Vision*, pages 725–732, Beijing, China, Oct. 2005.

# Learning to Mimic Motion of Human Arm and Hand Grabbing for Constraint Adaptation

Stephan Al-Zubi and Gerald Sommer

Cognitive Systems, Christian Albrechts University, Kiel, Germany  
{sa, gs}@ks.informatik.uni-kiel.de

**Abstract.** We propose a model for learning the articulated motion of human arm and hand grabbing. The goal is to generate plausible trajectories of joints that mimic the human movement using deformation information. The trajectories are then mapped to a constraint space. These constraints can be the space of start and end configuration of the human body and task-specific constraints such as avoiding an obstacle, picking up and putting down objects. Such a model can be used to develop humanoid robots that move in a human-like way in reaction to diverse changes in their environment and as a priori model for motion tracking. The model proposed to accomplish this uses a combination of principal component analysis (PCA) and a special type of a topological map called the dynamic cell structure (DCS) network. Experiments on arm and hand movements show that this model is able to successfully generalize movement using a few training samples for free movement, obstacle avoidance and grabbing objects.

## 1 Introduction

Human motion is characterized as being smooth, efficient and adaptive to the state of the environment. In recent years a lot of work has been done in the fields of robotics and computer animation to capture, analyze and synthesize this movement with different purposes [1,2,3]. In robotics there has been a large body of research concerning humanoid robots. These robots are designed to have a one to one mapping to the joints of the human body but are still less flexible. The ultimate goal is to develop a humanoid robot that is able to react and move in its environment like a human being. So far the work that has been done is concerned with learning single gestures like drumming or pole balancing which involves restricted movements primitives in a simple environment or a preprogrammed movement sequence like a dance. An example where more adaptivity is needed would be a humanoid tennis robot which, given its current position and pose and the trajectory of the incoming ball, is able to move in a human-like way to intercept it. This idea enables us to categorize human movement learning from simple to complex as follows: (A) Imitate a simple gesture, (B) learn a sequence of gestures to form a more complex movement, (C) generalize movement over the range allowed by the human body, and (D) learn different classes of movement specialized for specific tasks (e.g. grasping, pulling, etc.).

This paper introduces two small applications for learning movement of type (C) and (D). The learning components of the proposed model are not by themselves new. Our contribution is presenting a supervised learning algorithm which learns to imitate human movement that is specifically more adaptive to constraints and tasks than other models. This also has the potential to be used for motion tracking where more diverse changes in movement occur. We will call the state of the environment and the body which affects the movement as constraint space. This may be as simple as object positions which we must reach or avoid, a target body pose or more complex attributes such as the object's orientation and size when grabbing it. The first case we present is generating realistic trajectories of a simple kinematic chain representing a human arm. These trajectories are adapted to a constraint space which consists of start and end positions of the arm as shown in fig. 1. The second case demonstrates how the learning algorithm can be adapted to the specific task of avoiding an obstacle where the position of the obstacle varies. The third case demonstrates how hand grabbing can be adapted to different object sizes and orientations.

The model accomplishes this by aligning trajectories. A trajectory is the sequence of body poses which change in time from the start to the end of a movement. Aligning trajectories is done by scaling and rotation transforms in angular space which minimizes the distance between similar poses between trajectories. After alignment we can analyze their deformation modes which describe the principal variations of the shape of trajectories. The constraint space is mapped to these deformation modes using a topological map.

Next, we describe an overview of the work done related to movement learning and compare them with the proposed model.

## 2 State of the Art

There are two representations for movements: pose based and trajectory based. We will describe next pose based methods.

Generative models of motion have been used in [2,1] in which a nonlinear dimensionality reducing method called Scaled Gaussian Latent Variable Model (SGPLVM) is used on training samples in pose space to learn a nonlinear latent space which represents the probability distribution of each pose. Such a likelihood function was used as a prior for tracking in [1] and finding more natural poses for computer animation in [2] that satisfy constraints such as that the hand has to touch some points in space. Another example of using a generative model for tracking is [4] in which a Bayesian formulation is used to define a probability distribution of a pose in a given time frame as a function of the previous poses and current image measurements. This prior model acts as a constraint which enables a robust tracking algorithm for monocular images of a walking motion. Another approach using Bayesian priors and nonlinear dimension reduction is used in [5] for tracking.

After reviewing pose probabilistic methods, we describe in the following trajectory based methods. Schaal [3] has contributed to the field of learning

movement for humanoid robots. He describes complex movements as a set of movement primitives (DMP). From these a nonlinear dynamic system of equations are defined that generate complex movement trajectories. He described a reinforcement learning algorithm that can efficiently optimize the parameters (weights) of DMPs to learn to imitate a human in a high dimensional space. He demonstrated his learning algorithm for applications like drumming and a tennis swing.

To go beyond a gesture imitation, in [6] a model for segmenting and morphing complex movement sequences was proposed. The complex movement sequence is divided into subsequences at points where one of the joints reaches zero velocity. Dynamic programming is used to match different subsequences in which some of these key movement features are missing. Matched movement segments are then combined with each other to build a morphable motion trajectory by calculating spatial and temporal displacement between them. For example, morphable movements are able to naturally represent movement transitions between different people performing martial arts with different styles.

Another aspect of motion adaptation and morphing with respect to constraints comes from computer graphics on the topic of re-targeting. As an example, Gleicher [7] proposed a nonlinear optimization method to re-target a movement sequence from one character to another with an identical structure but different segment lengths. The problem is to satisfy both the physical constraints and the smoothness of movement. Physical constraints are contact with other objects like holding the box.

The closest work to the model presented in this paper is done by Banerjee [8]. He described a method for learning movement adaptive to start and end positions. His idea is to use a topological map called Dynamic Cell Structure (DCS) network [9]. The DCS network learns the space of valid arm configurations. The shortest path of valid configurations between the start and end positions represents the learned movement. He demonstrated his algorithm to learn a single gesture and also obstacle avoidance for a single fixed obstacle.

### 3 Contribution

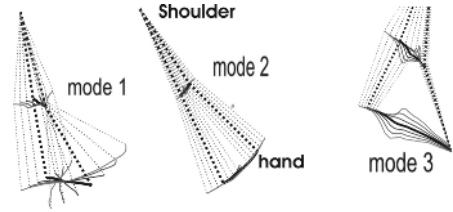
The main difference between pose based methods and our approach is that instead of learning the probability distribution in pose space, we model the variation in trajectory space (each trajectory being a sequence of poses). This representation enables us to generate trajectories that vary as a function of environmental constraints and to find a more compact representation of variations than allowed by pdfs in pose space alone. Pose pdfs would model large variations in trajectories as a widely spread distribution which makes it difficult to trace the sequence of legal poses that satisfy the constraints the human actually makes without some external reference like motion sequence data.

Our approach models movement variation as a function of the constraint space. However, style based inverse kinematics as in [2] selects the most likely poses that satisfy these constraints. This works well as long as the pose constraints do not deviate much from the training data. This may be suitable for





**Fig. 1.** Movements of the arm



**Fig. 2.** Movement modes of the arm constructed in 3D space

animation applications but our goal here is to represent realistic trajectories adapted to constraints without any explicit modeling. Banarer [8] uses also a pose based method and the model he proposed does not generalize well because as new paths are learned between new start and end positions, the DCS network grows very quickly and cannot cope with the curse of dimensionality. Our DCS network generalizes over trajectory space not poses enabling more adaptivity.

Gleicher [7] defines an explicit adaptation model which is suitable to generate a visually appealing movement but requires fine tuning by the animator because it may appear unrealistic. This is because it explicitly morphs movement using a prior model rather than learning how it varies in reality as done in [2].

In the case of Schaal [3], we see that DMPs although flexible are not designed to handle large variations in trajectory space. This is because reinforcement learning adapts to a specific target human trajectory.

Morphable movements [6] define explicitly the transition function between two or more movements without considering the constraint space. Our method can learn the nonlinear mapping between constraint space and movements by training from many samples. The variation of a movement class is learned and not explicitly pre-defined.

To sum up, we have a trajectory based learning model which learns the mapping between constraints and movements. The movement can be more adaptive and generalizable over constraint space. It learns movements from samples and avoids explicit modeling which may generate unrealistic trajectories.

## 4 Learning Model

After describing the problem, the concept for learning movement will be explained and how this model is implemented.

In order to develop a system which is able to generalize movement, we need a representation of movement space. The first step is to learn the deformations of the articulated movement itself and the second is to learn how movement changes with start and end configuration and environmental constraints. The mechanics of movement are called *intrinsic features*. The changes of intrinsic features with respect to absolute position and environment are called *extrinsic features*. The intrinsic features describe movement primitives that are characteristic for a

human being. These features are the relative coordination of joints in space and time. Extrinsic features can be characterized as the variation of intrinsic features in the space of all possible absolute start and end positions of the joints and any environmental constraints such as obstacle positions.

The difference between intrinsic and extrinsic features that characterizes movement enables the formulation of a learning model. This model consists of two parts: The first part is responsible for learning intrinsic features which uses principal component analysis (PCA). It is applied on the aligned trajectories of the joints to reduce the dimensionality. The second part models the extrinsic features using a special type of an adaptive topological map called the dynamic cell structure (DCS) network. The DCS learns the nonlinear mapping from the extrinsic features to intrinsic features that are used to construct the correct movement that satisfies these extrinsic features.

#### 4.1 Intrinsic Features Using PCA

We assume in this section for demonstration purposes a kinematic chain representing a human arm shown in Fig. 1. It consists of 2 joints: shoulder and elbow. Each joint has 2 degrees of freedom  $(\phi, \theta)$  which represent the direction of the corresponding limb in spherical coordinates.

To perform statistical analysis, we record several samples of motion sequences. In each motion sequence the 3D positions of the joints are recorded with their time. The first step is to interpolate between the 3D points from the stereo cameras of each movement sequence. We end up with a set of parametric curves  $\{\mathbf{p}_k(t)\}$  for each motion sequence  $k$  where  $\mathbf{p}_k(t)$  returns the position vector of all the joints at time  $t$ . After that, each  $\mathbf{p}_k(t)$  is sampled at  $n$  equal time intervals from the start of the sequence  $k$  to its end forming a vector of positions  $\mathbf{v}_k = [\mathbf{p}_{1,k}, \mathbf{p}_{2,k} \dots \mathbf{p}_{n,k}]$ . By Using the time  $t$  as an interpolation variable, the trajectory is sampled such that there are more pose samples at high curvature regions where the arm slows down than at low curvature regions where the arm speeds up. Then the Euclidean coordinates of each  $\mathbf{v}_k$  are converted to relative orientation angles of all joints  $\mathbf{s}_{j,k} = (\phi_{j,k}, \theta_{j,k}), j = 1 \dots n$  in spherical coordinates:  $\mathbf{S}_k = [\mathbf{s}_{1,k}, \mathbf{s}_{2,k}, \dots \mathbf{s}_{n,k}]$ . After this we align the trajectories taken by all the joints with respect to each other. Alignment means to find rotation and scaling transformations on trajectories that minimize the distances between them. This alignment makes trajectories comparable with each other in the sense that all extrinsic features are eliminated leaving only deformation information. The distance measure between two trajectories is the mean radial distance between corresponding direction vectors formed from the orientation angles of the joints. Two transformations are applied on trajectories to minimize the distance between them: 3D rotation and angular scaling between the trajectory's direction vectors, where a scale factor is centered at any point on the trajectory. We can extend this method to align many sample trajectories with respect to their mean until the mean converges. An example of aligning a group of trajectories is shown in Fig. 3. The left image shows hand and elbow direction trajectories before alignment and the right is after. We see how the hand trajectories cluster together. The

$p$  aligned trajectories are represented as  $X = [\mathbf{S}_1^T \dots \mathbf{S}_k^T \dots \mathbf{S}_p^T]^T$ . Principal component analysis is applied on  $X$  yielding latent vectors  $\Psi = [\psi_1 \psi_2 \dots \psi_n]$ . Only the first  $q$  components are used where  $q$  is chosen such that the components cover a large percentage of the data  $\Psi_q = [\psi_1 \psi_2 \dots \psi_q]$ . Any point in eigenspace can then be converted to the nearest plausible data sample using the following equation

$$\mathbf{S} = \bar{\mathbf{S}} + \Psi_q \mathbf{b} \quad (1)$$

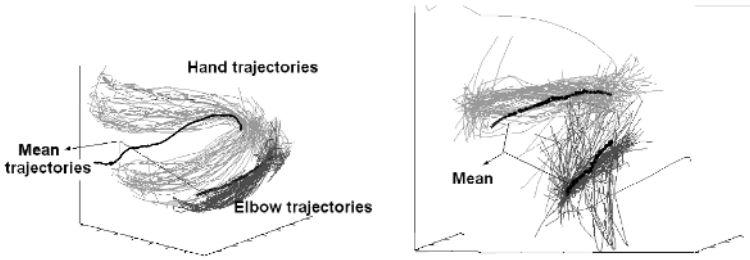
where  $\bar{\mathbf{S}} = \frac{1}{p} \sum_{k=1}^p \mathbf{S}_k$  and  $\mathbf{b}$  is an eigenpoint.

The latent coordinates  $\mathbf{b}$  represent the linear combination of deformations from the average paths taken by the joints. An example of that can be seen in Fig. 2. In this example, the thick lines represent the mean path and the others represent  $\pm 3$  standard deviations in the direction of each eigenvector which are called modes. The first mode (left) represents the twisting of the hand's path around the elbow and shoulder. The second mode (middle) shows the coordination of angles when moving the hand and elbow together. The third mode (right) represent the curvatures of the path taken by the hand and shoulder. The reason for using a linear subspace method like PCA in this paper is because the trajectories are highly covariant since they change in direct response to a low dimensional constraint space. The advantage of this representation is that the dimension reduction depends only on the dimension of the constraint space and not on the dimension of the trajectory which is much higher. As a result we do not require many training samples to extract the deformation modes but only enough samples to cover the constraint space.

## 4.2 Extrinsic Features Using DCS

PCA performs a linear transform (i.e. rotation and projection in (1)) which maps the trajectory space into the eigenspace. The mapping between constraint space and eigenspace is generally nonlinear. To learn this mapping we use a special type of self organizing maps called Dynamic Cell Structure which is a hybrid between radial basis networks and topologically preserving maps [9]. DCS networks have many advantages: They have a simple structure which makes it easy to interpret results, they adapt efficiently to training data and they can cope with changing distributions. They consist of neurons that are connected to each other locally by a graph distributed over the input space. These neurons also have radial basis functions which are Gaussian functions used to interpolate between these neighbors. The DCS network adapts to the nonlinear distribution by growing dynamically to fit the samples until some error measure is minimized. When a DCS network is trained, the output  $\mathbf{b}_{DCS}(\mathbf{x})$  which is a point in eigenspace can be computed by summing the activations of the best matching neuron (i.e. closest) to the input vector  $\mathbf{x}$  representing a point in constraint space and the local neighbors to which it is connected by an edge which is defined by the function  $A_p(\mathbf{x})$ . The output is defined as

$$\mathbf{b}_{DCS}(\mathbf{x}) = f_P^{nrbf}(\mathbf{x}) = \frac{\sum_{i \in A_p(\mathbf{x})} \mathbf{b}_i h(\|\mathbf{x} - \mathbf{c}_i\| / \sigma_i)}{\sum_{j \in A_p(\mathbf{x})} h(\|\mathbf{x} - \mathbf{c}_j\| / \sigma_j)}, \quad (2)$$



**Fig. 3.** Example of aligning a training set of trajectories represented as direction vectors tracing curves on a unit sphere

where  $\mathbf{c}_i$  is the receptive center of the neuron  $i$ ,  $\mathbf{b}_i$  represents a point in eigenspace which is the output of neuron  $i$ ,  $h$  is the Gaussian kernel and  $\sigma_i$  is the width of the kernel at neuron  $i$ .

The combination of DCS to learn nonlinear mapping and PCA to reduce dimension enables us to reconstruct trajectories from  $\mathbf{b}(\mathbf{x})$  using (1) which are then fitted to the constraint space by using scale and rotation transformations. For example, a constructed trajectory is fitted to a start and end position.

## 5 Experiments

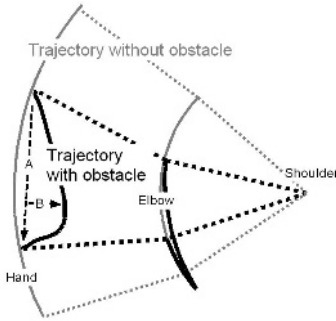
In order to record arm movements, a marker-based stereo tracker was developed in which two cameras track the 3D position of three markers placed at the shoulder, elbow and hand at a rate of 8 frames per second. This was used to record trajectory samples. Two experiments were conducted to show two learning cases: moving between two positions and avoiding an obstacle.

The first experiment demonstrates that our learning model reconstructs the nonlinear trajectories in the space of start-end positions. A set of 100 measurements were made for an arm movement consisting of three joints. The movements had the same start position but different end positions as shown in Fig. 1.

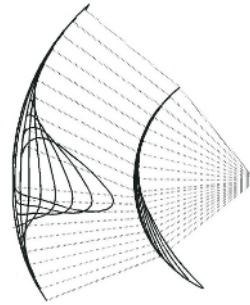
The first three eigenvalues have a smooth nonlinear unimodal distribution with respect to the start-end space. The first component explained 72% of the training samples, the second 11% and the third 3%.

The performance of the DCS network was first tested by a  $k$ -fold cross validation on randomized 100 samples. This was repeated for  $k = 10$  runs. In each run the DCS network was trained and the number of neurons varied between 6 to 11. The average distance between the DCS-trajectory and the data sample was  $3.9^\circ$  and the standard deviation was  $2.1^\circ$ . This shows that the DCS network was able to generalize well using only a small sample size (about 100).

We can compare with Banarar [8] who fixed the DCS network with an upper bound of 15 neurons to learn a single gesture and not many as in our experiment. He used simulated data of 70 samples with a random noise of up to  $5^\circ$  and the mean error was  $4.3^\circ$  compared to our result of  $3.9^\circ$  on real data. The



**Fig. 4.** Trajectory for obstacle avoidance in 3D space



**Fig. 5.** Variation of arm trajectory with respect to the obstacle

measurement error of the tracker is estimated to be  $4.6^\circ$  standard deviation which accounts for the similar mean errors. This shows that our model scales well.

Next, we demonstrate the algorithm for obstacle avoidance. In this case 100 measurements were taken for the arm movement with different obstacle positions as shown in Fig. 4. The black lines show the 3D trajectory of the arm avoiding the obstacle which has a variable position determined by the distance  $B$ . We see how the hand backs away from the obstacle and the elbow goes down and then upward to guide the hand to its target.  $A$  is the Euclidian distance between the start and end positions of the hand. The grey lines represent a free path without obstacles. In this case we need to only take the first eigenvector from PCA to capture the variation of trajectories due to obstacle position. This deformation mode is shown in Fig. 5. We define the relative position of the obstacle to the movement as simply  $p = \frac{B}{A}$ . The DCS network learns the mapping between  $p$  and the eigenvalue with only 5 neurons. The learned movement can thus be used to avoid any obstacle between the start and end positions regardless of orientation or movement scale. This demonstrates how relatively easy it is to learn new specialized movements that are adaptive to constraints.

Finally, this model was demonstrated on hand grabbing. In this case 9 markers were placed on the hand to track the index and thumb fingers using a monocular camera as in Fig. 6. The 2D positions of the markers were recorded at a rate of 8.5 frames per second from a camera looking over a table. The objects to be grabbed are placed over the table and they vary by both size and orientation. The size ranged from 4 to 12 cm and orientation ranged from 0 to 60 degrees as depicted in Fig. 7 and 8. The tracker recorded 350 grabbing samples of which 280 was used for training the DCS and 70 for testing. The DCS learned the variation of movement with 95 neurons and PCA reduced the dimension from 600 to just 23. The first two modes characterize variation of scale and orientation as shown in Fig. 6. Fig. 7 and 8 depict an example comparison between grabbing movement generated by the DCS and an actual sample. Below we used two measures that characterize well grabbing: distance between the tips of the index finger and the

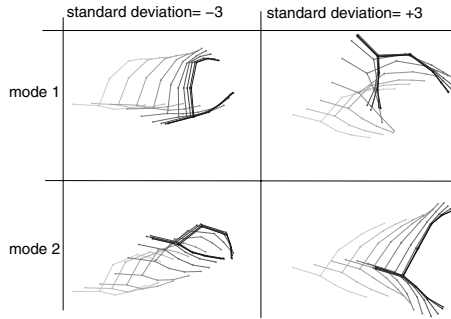


Fig. 6. The first two variation modes of grabbing

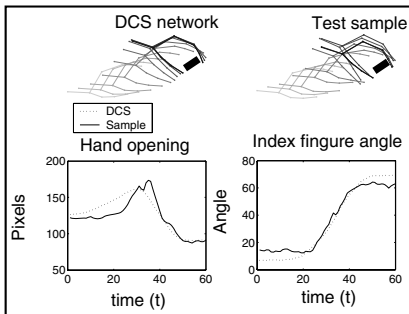


Fig. 7. Comparison between DCS and a grabbing movement for a 4 cm object at  $60^\circ$  with respect to the hand

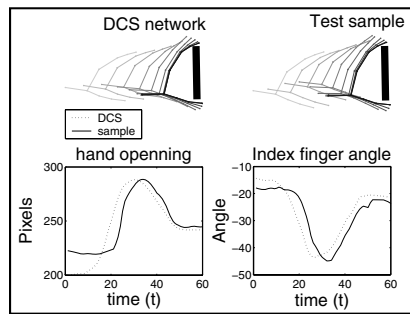


Fig. 8. Comparison between DCS and a grabbing movement for a 12 cm object at  $0^\circ$

thumb and the direction of the index finger’s tip with respect the the direction of the arm. We see that the DCS and sample profiles look very similar. In general, the model’s root mean square error for the first measure was 18 pixels for a  $800 \times 600$  images and  $8.5^\circ$  for the second measure.

## 6 Conclusion

We proposed a learning model for generation of realistic articulated motion. The model characterizes deformation modes that vary according to constraint space. A combination of DCS network to learn the nonlinear mapping and PCA to reduce dimensionality enables us to find a representation that can adapt to constraint space with a few samples. This trajectory based method is more suited for movement generation than pose based methods which are concerned with defining priors for good fitting with image data such as tracking. The proposed method models variation of movement with respect to constraints in a more clear way than the previously proposed methods. The potential uses of our method

is in developing humanoid robots that are reactive to their environment and also motion tracking algorithms that use prior knowledge of motion to make them robust. Specifically, trajectory prior knowledge about motion can help in cases where the tracked object is occluded in several successive frames. In such a case pose based pdfs will fail. Three small applications towards that goal were experimentally validated.

*Acknowledgments.* The work presented here was supported by the the European Union, grant COSPAL (IST-2003-004176). However, this paper does not necessarily represent the opinion of the European Community, and the European Community is not responsible for any use which may be made of its contents.

## References

1. Urtasun, R., Fleet, D.J., Hertzmann, A., Fua, P.: Priors for people tracking from small training sets. In: International Conference on Computer Vision (ICCV). (2005) 403–410
2. Grochow, K., Martin, S.L., Hertzmann, A., Popovic, Z.: Style-based inverse kinematics. *ACM Trans. Graph.* **23**(3) (2004) 522–531
3. Schaal, S., Peters, J., Nakanishi, J., Ijspeert, A.: Learning movement primitives. In: International Symposium on Robotics Research (ISPR2003), Springer Tracts in Advanced Robotics, Ciena, Italy (2004)
4. Sidenbladh, H., Black, M.J., Fleet, D.J.: Stochastic tracking of 3d human figures using 2d image motion. In: Proceedings of the 6th European Conference on Computer Vision (ECCV '00), London, UK, Springer-Verlag (2000) 702–718
5. Sminchisescu, C., Jepson, A.: Generative modeling for continuous non-linearly embedded visual inference. In: Proceedings of the twenty-first International Conference on Machine Learning (ICML '04), New York, NY, USA, ACM Press (2004)
6. Ilg, W., Bakir, G.H., Mezger, J., Giese, M.A.: On the representation, learning and transfer of spatio-temporal movement characteristics. *International Journal of Humanoid Robotics* (2004)
7. Gleicher, M.: Retargeting motion to new characters. In: Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '98), New York, NY, USA, ACM Press (1998) 33–42
8. Banarer, V.: STRUKTURELLER BIAS IN NEURONALEN NETZEN MITTELS CLIFFORD-ALGEBREN. Technical Report 0501, Technische Fakultät der Christian-Albrechts-Universität zu Kiel, Kiel (2005)
9. Bruske, J., Sommer, G.: Dynamic cell structure learns perfectly topology preserving map. *Neural Computation* **7**(4) (1995) 845–865

# Visual Hand Posture Recognition in Monocular Image Sequences

Thorsten Dick\*, Jörg Zieren\*, and Karl-Friedrich Kraiss

Institute of Man-Machine-Interaction, RWTH Aachen University, Germany

{dick, zieren, kraiss}@mmi.rwth-aachen.de

<http://www.mmi.rwth-aachen.de>

**Abstract.** We present a model-based method for hand posture recognition in monocular image sequences that measures joint angles, viewing angle, and position in space. Visual markers in form of a colored cotton glove are used to extract descriptive and stable 2D features. Searching a synthetically generated database of 2.6 million entries, each consisting of 3D hand posture parameters and the corresponding 2D features, yields several candidate postures per frame. This ambiguity is resolved by exploiting temporal continuity between successive frames. The method is robust to noise, can be used from any viewing angle, and places no constraints on the hand posture. Self-occlusion of any number of markers is handled. It requires no initialization and retrospectively corrects posture errors when accordant information becomes available. Besides a qualitative evaluation on real images, a quantitative performance measurement using a large amount of synthetic input data featuring various degrees of noise shows the effectiveness of the approach.

## 1 Introduction

Automatic recognition of hand gestures is an intuitive and efficient method for human-computer interaction. Applications for gesture input include virtual reality, motion capture, and sign language recognition. Vision-based recognition methods allow to measure both hand configuration and translational motion. Another important benefit is that a camera also records the user's face – a prerequisite for recognizing sign language.

For many tasks, such as fingertip detection or gesture classification, appearance-based 2D features (i.e. shape and texture) that can be extracted directly from the input image suffice [1,2]. Reconstruction of the 3D hand posture from 2D images opens up additional applications that require knowledge of individual finger flexion.

This paper describes a model-based approach using visual markers and a matching OpenGL hand model to map an observation, described by 2D features, to a 3D hand posture. A large database of such mappings is generated offline. Efficient algorithms for identifying candidates with similar features form the core of the system. Since ambiguities and uncertainties in individual frames cannot be prevented when using 2D input data, disambiguation is performed by exploiting temporal continuity of the gesturing motion over a period of several seconds. Smoothing in posture space prevents

---

\* Supported by grant VV-Z50 from the Interdisciplinary Centre for Clinical Research "BIOMAT" within the Faculty of Medicine at the RWTH Aachen University.



jerkiness that would otherwise result from the finite number of discrete postures in the database.

The system achieves near real-time speed on a standard PC. A qualitative evaluation on signed numbers is presented, as well as an exact measurement of posture error using a total of 37,500 synthetic input images.

## 2 Related Methods and Common Difficulties

Numerous approaches for hand posture recognition have been proposed in recent years. They differ in the number of cameras used, the type of features extracted from the input data, the supported degrees of freedom (DOF), and possible limitations regarding input posture and viewing angle. The mapping of features to postures can be performed by either deriving joint and viewing angles directly through inverse kinematics, or by parameterizing a hand model so that it yields matching features.

Multi-camera approaches restrict translational hand motion at least to the intersection of all cameras' viewfields. Since stereo is less effective for remote objects, the hand is usually recorded from a short distance. Existing publications therefore do not consider significant translational motion [3,4,5] and further require controlled recording conditions, e.g. placing the hand inside a box containing light source and cameras.

Feature extraction on images of the unmarked hand constitutes a challenging problem, especially in the presence of motion blur and camera noise. A robust feature which can be extracted using a simple skin color model is the hand's contour [6,7]. However, the contour is not stable because small changes in hand posture may greatly affect it. At the same time, by discarding texture it entails yet another loss of input information in addition to the 3D-to-2D projection. Many different hand postures result in the same contour (for example, a fist and a pointing index finger seen from the pointing direction), rendering this feature problematic for unrestricted posture recognition from arbitrary viewing angles.

Texture features such as edges are more descriptive but computationally demanding due to the high amounts of data and noise involved. Several systems therefore impose restrictions on the allowed input postures. In [8] a set of 26 postures is recognized in perfectly segmented single images of real hands taken from different viewing angles. An accuracy of 13.6% is reported, counting exact matches in posture and a maximum deviation of  $30^\circ$  in viewing angle. After generating a database of 107328 synthetic hand views (26 allowed postures seen from 86 viewing angles at 48 rotation angles), each including edges, lines extracted therefrom, and orientation histograms, the corresponding input features are used as a search key. Processing time per image is 15s on a 1.2GHz PC.

The method presented in [9] processes image sequences and allows hand postures commonly used in sign language, achieving an estimated person-dependent accuracy of 10% in finger flexion and  $15^\circ$  in viewing angle. The space of viewing angles and allowed postures is represented by 60 distinct Active Appearance Models (AAMs) that are extended to track translational motion. To narrow the search space per frame and thereby stabilize the system, transitions are only possible between models corresponding to anatomically similar postures. The AAM training data comprises manually labeled sequences of real images featuring a single person's hand. The hand must be suf-

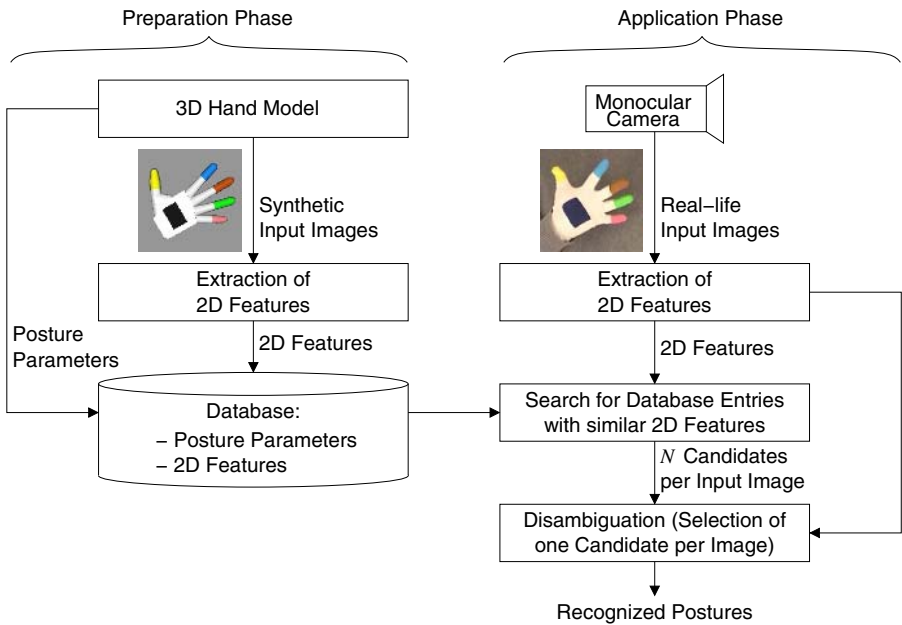
ficiently close to the camera to yield reliable texture information. Views from the finger or wrist direction, which naturally exhibit less texture, are not supported. Processing speed is 4 fps on a 1GHz PC.

Several approaches limit the degree of self-occlusion in the input images [4,5,10,11,12], recognizing only a subset of all common hand postures. Methods that iteratively refine a state estimate typically require the hand to assume a specific initialization posture in the first frame [12,13,7]. Except for [6], all above systems are susceptible to registration errors since they only pursue a single posture estimate at a time, not accounting for ambiguities in feature space.

### 3 System Overview

Fig. 1 shows an overview of the system. The user wears a cotton glove equipped with six differently colored visual markers, five covering approx. half of each finger and the thumb, and another on the back of the hand. This allows to extract descriptive and stable 2D features from a monocular view. The markers' geometry lends itself to an elliptical approximation in the image plane, resulting in a very compact representation. Hand posture recognition is performed by matching a synthetic hand model featuring identical markers to minimize deviation in feature space.

In a preparation phase the hand model is used to generate a large number of postures seen from many different view angles. Each posture, together with the corresponding 2D features extracted from the synthetic view, is stored in a database. For evaluation we



**Fig. 1.** System overview

used a database size of 2.6 million entries. This does not include rotation in the image plane, which is computed online.

Posture recognition is performed by using the 2D features extracted from the input images as a key for querying the database. For each frame a fixed number of  $N$  postures whose features have high similarity to the extracted features are retrieved. This candidate space is then searched for a sequence that maximizes continuity in both posture and feature space. Spline interpolation between successive frames, considering match quality in each, finally yields a smooth posture sequence not restricted to the discretized posture space of the database.

## 4 Hand Model

Regarding possible configurations of fingers and thumb, the human hand has 21 DOF [14]. Each finger possesses one DOF for each of its joints plus a fourth DOF for sidewise abduction. The thumb requires five DOF due to its greater flexibility. Our hand model reduces this to seven DOF by assuming dependencies between a finger's joints. Fore to little finger are modeled by a single parameter each, ranging from 0.0 (fully outstretched) to 1.0 (maximum bending). The thumb is modeled similarly, using two additional parameters to reflect its flexibility. For a posture  $P$  the seven bending parameters are denoted by  $B^P$ .

Besides dealing with finger bendings the model also handles a posture's viewing angle, i.e. the hand's orientation in space. On the surface of an imaginary sphere around the hand (called the view sphere), each point corresponds to a specific view onto the hand. A view point is thus characterized by a latitude  $v_{\text{lat}}$  and a longitude  $v_{\text{lon}}$ . Additionally, for each view point a camera (or hand) rotation  $v_{\text{rot}}$  is possible. For a posture  $P$  these three angles are indicated by  $V^P$ .

In summary, the model parameters for each posture  $P$  comprise ten values and are denoted by  $P = \langle B^P, V^P \rangle$ . For a given posture  $P$  the corresponding synthetic hand image is rendered using OpenGL, modeling finger phalanges as simple cylinders, joints as spheres, and the palm as a combination of several polygons.

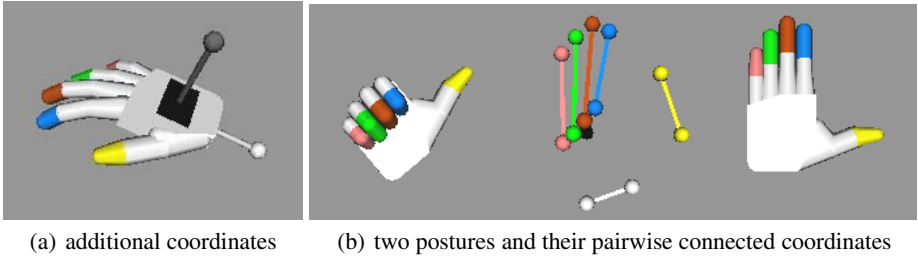
### 4.1 Posture Difference

Besides describing postures and visualizing them, the hand model offers an elegant way to express the difference between two postures. A mapping  $\Psi: \langle B^P, V^P \rangle \mapsto \langle \mathbf{c}_0^P, \dots, \mathbf{c}_6^P \rangle$  transforms the hand model parameters  $P$  to seven coordinates in space relative to the center of the palm.  $\mathbf{c}_0$  to  $\mathbf{c}_4$  represent the positions of the five finger tips.  $\mathbf{c}_5$  and  $\mathbf{c}_6$  are the coordinates of two additional "fingers" above the back of hand and below the wrist as shown in Fig. 2 (a), capturing the orientation of the hand, i.e. the view angles.

The difference  $\Delta_P$  between two postures  $Q$  and  $R$  is then defined as:

$$\Delta_P(Q, R) := \Psi(B^Q, V^Q) - \Psi(B^R, V^R) := \sum_{k=0}^6 |\mathbf{c}_k^Q - \mathbf{c}_k^R|^2 \quad (1)$$

where  $|\cdot|$  denotes the Euclidean distance. Fig. 2 (b) visualizes these distances.



**Fig. 2.** Posture difference

## 5 2D Features

In every input frame color-based segmentation is performed to detect the markers. While this is trivial for synthetic images, real-life images may yield several candidates per marker. Disambiguation is performed by pursuing multiple hypotheses over time, computing plausibility scores based on the candidates' geometry and their continuity in feature space. The winner hypothesis is chosen only at the end of the sequence, exploiting all available information. A detailed description of multiple hypotheses tracking can be found in [1].

To reduce the effects of noise and to minimize memory requirements, each detected marker  $k$  is approximated by an ellipse  $E_k$ , specified by area  $f$ , center  $\mathbf{m}$ , radii  $a$  and  $b$ , and orientation  $o$ , i.e.:  $E_k = \langle f_k, \mathbf{m}_k, a_k, b_k, o_k \rangle$ .

For invisible markers  $f$  is zero, all other features are undefined. To achieve translation independence, all centers  $\mathbf{m}_k$  are specified relative to the center of gravity (COG) of all visible marker regions. Furthermore, area  $f_k$  is normalized by  $n_2 = \sum_k f_k$ . Distances and lengths are normalized by  $n_1 = \sqrt{n_2}$ . This provides independence of the distance between hand and camera, as well as camera resolution. Thus, for each input frame  $I$  the feature set  $F^I$  describing the six markers is given by  $F^I = \langle E_0^I, E_1^I, \dots, E_5^I \rangle$ .

## 6 Static Posture Recognition

This stage extracts a set of  $N$  plausible postures from the database for each input image, where  $N$  lies in the range of approx. 100 to 1000. A further high level stage will resolve ambiguities by considering all generated candidates for successive input images.

### 6.1 Appearance Database

A database entry contains the hand model parameters  $\langle B^P, V^P \rangle$  for a posture  $P$  along with the features  $F^P$  that were extracted from the corresponding synthetic image. The set  $B = \{B^P | P \in \text{DB}\}$  of all database postures' finger bending parameters was defined by selecting eight bending values for the thumb, seven for fore and middle finger each, plus six values each for ring and little finger, totaling 14,112 postures per view.

For the set  $V = \{V^P | P \in \text{DB}\}$  of all database postures' view points, angles in steps of  $18^\circ$  have been chosen. Special care has to be taken at the view sphere's poles, where

a change of longitude resembles a rotation (an effect called gimbal lock), so  $v_{lon} = 0$  for  $v_{lat} = \pm 90$ . Rotation  $v_{rot}$  is set to zero for all database postures. Thus  $V$  contains these triples of  $\langle v_{lat}, v_{lon}, v_{rot} \rangle$ :

$$V = \{ \langle 18i, 18j, 0 \rangle \mid i, j \in \mathbb{Z} \wedge -5 < i < 5 \wedge 0 \leq j < 20 \} \cup \{ \langle \pm 90, 0, 0 \rangle \} \quad (2)$$

Because a rotation will leave the feature ellipses' areas as well as their relative distances unchanged,  $v_{rot}$  can be reconstructed before comparing database and input features. With  $|V| = 182$  the database contains  $D = 2,568,384$  entries. Considering rotations in steps of  $18^\circ$  a total of  $51,367,680$  postures can be recognized by the system.

In order to speed up database retrieval a  $B^*$ -like tree of height six with a fixed branching factor is used. Only  $f_k$  is considered, so the tree's depth equals the number of markers. The branching intervals at each node are non-overlapping, but if a query is within a certain range of an interval border, traversal continues in the neighboring subtree as well. The standard deviation of the interval's elements is used to quantify this range.

## 6.2 Feature Rotation

Let  $F^{ex}$  be the features extracted from the current input image and  $F^{db}$  those of a database candidate provided by the search tree. Like for all database entries,  $v_{rot}^{db} = 0$ . Let  $\phi(\mathbf{p}, \alpha)$  denote the rotation of point  $\mathbf{p}$  by  $\alpha$ . We define the rotation  $\hat{\alpha}(F^{ex}, F^{db})$  that estimates  $v_{rot}^{db}$  with respect to  $F^{ex}$  by

$$\hat{\alpha}(F^{ex}, F^{db}) := \underset{\alpha}{\operatorname{argmin}} \left\{ \sum_k f_k^{db} \cdot f_k^{ex} \cdot \left| \phi(\mathbf{m}_k^{db}, \alpha) - \mathbf{m}_k^{ex} \right|^2 \right\} \quad (3)$$

Due to weighting each summand by the product of the corresponding areas, bigger ellipses have more influence on the result than small ones.

Graphically, the two feature sets are aligned to their COGs and rotated until the sum of squared distances between corresponding ellipses, weighted by the product of their normalized area, is minimal. Since the rotation is actually a camera rotation, it propagates directly from features to postures. In the following,  $F^{db}$  is assumed to be the database features rotated according to (3).

## 6.3 Feature Difference

Searching the database for the  $N$  feature sets that are most similar to the extracted features  $F^{ex}$  requires to compute a scalar feature difference  $\Delta_F(F^{ex}, F^{db})$  that quantifies the similarity between  $F^{ex}$  and a set of database features  $F^{db}$  provided by the search tree. We use eight approximately equispaced points arranged counterclockwise on the ellipse's border (four of which lie at the intersection with the primary and secondary axes) and compute the sum of squared distances between these points for two corresponding ellipses  $E_k^{ex}$  and  $E_k^{db}$ . Of the eight possible mappings between both sets of points the one that minimizes this sum is used. This defines a geometric difference measure  $\Delta_E(E_k^{ex}, E_k^{db})$ .

For  $f_k^{ex} = f_k^{db} = 0$  we define  $\Delta_E = 0$ . If  $f_k^{ex} > 0 \wedge f_k^{db} = 0$  the affected marker in the database posture is made visible by not rendering any other component of the hand

model (this is done offline). The now visible marker's COG is then used in place of the eight border points to compute  $\Delta_E$ . If  $f_k^{\text{ex}} = 0 \wedge f_k^{\text{db}} > 0$  the database is searched for a posture  $Q$  that differs from  $P^{\text{db}}$  only in the bending of the affected finger, and for which  $f_k$  is minimal or zero (again this happens offline).  $\Delta_E$  for marker  $k$  is then computed between  $Q$  and  $P^{\text{db}}$ . In general, if a marker's visibility differs between  $F^{\text{ex}}$  and  $F^{\text{db}}$ , the visible marker's area will be small since the search tree returns only candidates with  $f_k^{\text{db}} \approx f_k^{\text{ex}} \forall k$ .

In order to favor shape similarity over position congruence a weighting of  $\Delta_E$  by the difference of the ellipses' area is performed. The feature difference is thus defined as

$$\Delta_F(F^{\text{ex}}, F^{\text{db}}) := \sum_k \Delta_E(E_k^{\text{ex}}, E_k^{\text{db}}) \cdot (1 + |f_k^{\text{ex}} - f_k^{\text{db}}|) \quad (4)$$

When querying the database the search tree returns  $M$  entries, where  $N \ll M \ll D$ .  $\Delta_F$  is computed for all  $M$  entries to find the  $N$  that best match  $F^{\text{ex}}$ , which form the set of hypotheses for the considered frame.

## 7 Posture Sequence Recognition

The recognition of posture sequences is based upon the hypotheses provided by the static recognition stage described in the previous section. It computes the actual recognition result and can also be used to optimize system parameters by synthetically generating input data for which ground truth is known. Fig. 3 shows a schematic overview.

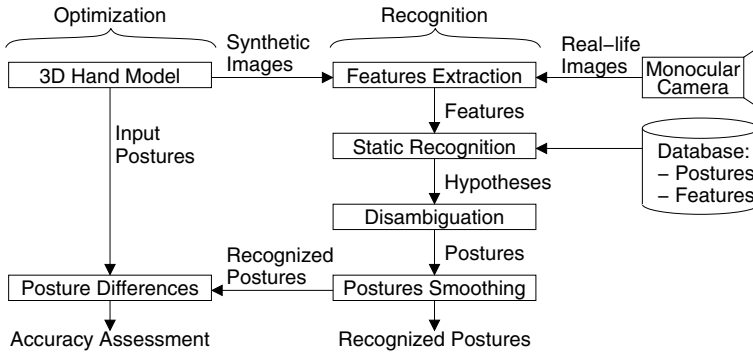


Fig. 3. Posture sequence recognition and parameter optimization

### 7.1 Disambiguation

For a sequence  $I(0), \dots, I(T-1)$  of input images, the corresponding features  $F^{\text{ex}}(0), \dots, F^{\text{ex}}(T-1)$  are extracted, for which the static recognition stage generates the sets of hypotheses  $H(0), \dots, H(T-1)$ , each containing  $N$  possible database features and postures, i.e.  $H(t) = \langle h_0(t), \dots, h_{N-1}(t) \rangle$  with  $h_n(t) = \langle F_n^{\text{db}}(t), P_n^{\text{db}}(t) \rangle$  for  $t = 0, \dots, T-1$  and  $n = 0, \dots, N-1$ .

At 25 fps,  $T = 75$  for a three second input sequence. We use  $N = 200$ , resulting in  $N^T \approx 3.8 \cdot 10^{172}$  posture sequence in hypothesis space, and apply the Viterbi algorithm, which scales with  $O(TN^2)$ . The step metric  $\sigma(h_i(t-1), h_j(t))$ , which represents the cost of choosing hypothesis  $h_j(t)$  given that  $h_i(t-1)$  is its predecessor, is defined as

$$\sigma(h_i(t-1), h_j(t)) := (1 - \gamma) \cdot \max \left\{ \Delta_P \left( P_i^{\text{db}}(t-1), P_j^{\text{db}}(t) \right) - \Delta_*, 0 \right\} + \gamma \cdot \Delta_F \left( F^{\text{ex}}(t), F_j^{\text{db}}(t) \right) \quad \text{for } t = 1, \dots, T-1 \quad (5)$$

The step metric combines the feature difference  $\Delta_F$  between a hypothesis and the corresponding extracted features and the posture difference  $\Delta_P$  between successive hypotheses, thus favoring a high feature similarity as well as temporal continuity of postures.  $\Delta_*$  is a small, positive, empirically determined value subtracted from  $\Delta_P$  in order to allow small posture deviations.  $\gamma$  weights the feature and posture differences and was optimized iteratively using sequences of synthetic images provided by the hand model (cf. left part of Fig. 3).

The path metric  $\pi(h_i(t))$  is the sum of step metrics along the path leading to  $h_i(t)$  and is initialized to

$$\pi(h_i(0)) := \gamma \cdot \Delta_F(F^{\text{ex}}(0), F_i^{\text{db}}(0)) \quad (6)$$

The path that minimizes  $\pi(h)$  for  $h \in H(T-1)$  constitutes the resulting posture sequence  $S(I(0), \dots, I(T-1)) = \langle h(0), \dots, h(T-1) \rangle$ . Since this is only computed after several seconds of input video have been observed, posture errors in individual frames are retrospectively corrected as soon as they become apparent in the light of additional observations.

## 7.2 Smoothing

The sequences provided by the disambiguation stage only contain postures from the database, i.e. from a finite, discrete subset of the infinite space of continuous postures. In order to bridge these jumps and to alleviate small recognition errors, the sequence is smoothed, which happens individually for each bending and view angle parameter.

Recall that  $h(t) = \langle F^{\text{db}}(t), P^{\text{db}}(t) \rangle$  and  $P^{\text{db}}(t) = \langle B^{P^{\text{db}}(t)}, V^{P^{\text{db}}(t)} \rangle$ . Let  $r_0, \dots, r_{R-1}$  be the indices of the most reliable hypotheses from  $\langle h(0), \dots, h(T-1) \rangle$ . Starting with  $r_0 = 0$  the next most reliable hypothesis for  $h(r_i)$  is  $h(r_{i+1}) \in \{h(r_i + \alpha), \dots, h(r_i + \beta)\}$  with minimal  $\Delta_F(F^{\text{ex}}(r_{i+1}), F^{\text{db}}(r_{i+1}))$ . Appropriate values are  $\alpha = 1$  and  $\beta = 4$ .

Let  $\rho(t)$  denote a hand model parameter in  $P^{\text{db}}(t)$ . For  $\{\langle r_i, \rho(r_i) \rangle \mid i = 0, \dots, R-1\}$  the system computes the interpolating cubic spline  $s(t)$ , i.e.  $s(r_i) = \rho(r_i)$  for  $i = 0, \dots, R-1$ . The smoothed sequence of hand model parameters is then given by  $s(0), \dots, s(T-1)$ . Performing this interpolation individually for each hand model parameter yields the smoothed sequence of postures.

## 8 Evaluation

We evaluated the system's performance on synthetic and real-life images. Using a standard PC with a 1.6 GHz CPU and 1.25 GB RAM, processing speed is approx. 5 fps.

## 8.1 Synthetic Input

Synthetic input images offer the opportunity to measure recognition precision quantitatively. We generated 500 random sequences that evenly cover the posture space, each consisting of 75 consecutive postures featuring continuous changes in both  $B^P$  and  $V^P$ . The corresponding images have been distorted by blanking  $3 \times 3$  tiles overlapping by 1 pixel with a probability  $p_{\text{noise}}$  as illustrated in Fig. 4. The results are listed in Tab. 1, where ‘‘Posture Difference’’ refers to (1), ‘‘Fingertip Distance’’ denotes the average Euclidean distance (in cm) of corresponding fingertips without consideration of view angles, and ‘‘Finger Bending Deviation’’ refers to the posture parameters  $B$  (cf. Sec. 4). These results demonstrate that the system is robust to significant noise.

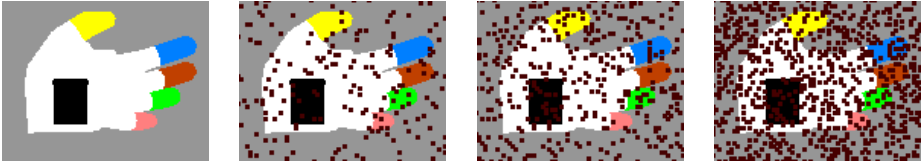


Fig. 4. Noise levels  $p_{\text{noise}} = 0\%, 5\%, 10\%, 20\%$  (left to right; image resolution  $136 \times 105$ )

Table 1. Recognition accuracy for synthetic input

$p_{\text{noise}}$	Posture Difference		Fingertip Distance		Median of Finger Bending Deviation						
	average	median	average	median	th1	th2	th3	ff	mf	rf	lf
0%	111.542	74.642	1.940	1.782	0.227	0.257	0.233	0.075	0.077	0.087	0.099
5%	108.506	75.912	1.952	1.795	0.224	0.255	0.230	0.080	0.077	0.084	0.097
10%	112.841	76.600	1.965	1.815	0.228	0.262	0.248	0.074	0.077	0.087	0.100
20%	144.636	97.338	2.272	2.102	0.241	0.242	0.257	0.105	0.096	0.100	0.134

## 8.2 Real-Life Input

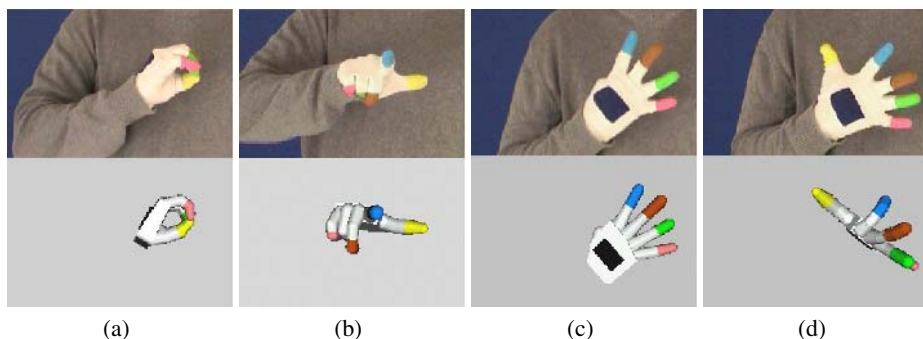
Real-life performance has been tested on sequences of signed numbers. Fig. 5 depicts some examples, each showing a magnification ( $220 \times 145$ ) of the actual input image ( $360 \times 288$ ) and the recognized posture. By visual comparison match quality is high. Fig. 5 (d) illustrates the system’s reaction to marker detection failures.

## 9 Discussion and Conclusion

We have presented a method that recognizes hand postures (finger bendings and view angle) from monocular image sequences. It imposes no posture restrictions and requires no initialization or person-dependent training. Performing appearance-based matching by searching a database, coupled with a Viterbi search in posture space, provides an efficient means of handling ambiguities.

Our experiments show promising results. Future work will primarily focus on processing speed, which can be increased by removing anatomically impossible postures and views from the database, and by improving the search tree’s efficiency.





**Fig. 5.** Real-life examples (for (d) the back-of-hand marker has been removed manually)

## References

1. Zieren, J., Kraiss, K.F.: Robust Person-Independent Visual Sign Language Recognition. In: *IbPRIA 2005. Volume Lecture Notes in Computer Science.* (2005)
2. Zieren, J.: Hand Gesture Commands. In: *Advanced Man-Machine Interaction.* Springer (2006) 7–56
3. Bebis, G., Harris, F., Erol, A., Yi, B., Martinez, J., Hernandez-Usabiaga, J., Fritzinger, S.: Development of a Nationally Competitive Program in Computer Vision Technologies for Effective Human-Computer Interaction in Virtual Environments. Technical report, BioVIS Lab. in BioVIS Technology Center of NASA Ames Research Center (2002)
4. Nölker, C.: Grefit: Ein System zur Visuellen Erkennung von Handposturen. PhD thesis, Technische Fakultät der Universität Bielefeld (2000)
5. Rehg, J.M.: Visual Analysis of High DOF Articulated Objects with Application to Hand Tracking. PhD thesis, School of Computer Science, Carnegie Mellon University (1995)
6. Imai, A., Shimada, N., Shirai, Y.: 3-D Hand Posture Recognition by Training Contour Variation. In: *International Conference on Automatic Face and Gesture Recognition.* (2004)
7. Vittrup, M., Sørensen, M.K.D., McCane, B.: Pose Estimation by Applied Numerical Techniques. In: *Image and Vision Computing, New Zealand.* (2002)
8. Athitsos, V., Sclaroff, S.: Estimating 3D Hand Pose from a Cluttered Image. In: *Proc. IEEE CVPR.* (2003)
9. Fillbrandt, H., Akyol, S., Kraiss, K.F.: Extraction of 3D Hand Shape and Posture from Image Sequences for Sign Language Recognition. In Azada, D., ed.: *IEEE International Workshop on Analysis and Modeling of Faces and Gestures (AMFG).* (2003)
10. Chua, C.S., Guan, H., Ho, Y.K.: Model-based 3D hand posture estimation from a single 2D image. *Image Vision Comput.* **20**(3) (2002)
11. Lathuilière, F., Hervé, J.Y.: Visual Tracking of Hand Posture in a Robot Control Application. In: *Proceedings of the Vision Interface Conference.* (1999)
12. Heap, T., Hogg, D.: Towards 3D Hand Tracking using a Deformable Model. In: *International Conference on Automatic Face and Gesture Recognition.* (1996)
13. Holden, E.J., Owens, R., Roy, G.G.: 3D Hand Tracker for Visual Sign Recognition. (1999)
14. Sturman, D.J.: Whole-hand Input. PhD thesis, School of Architecture and Planning, Massachusetts Institute of Technology (1992)

# Monocular Body Pose Estimation by Color Histograms and Point Tracking

Daniel Grest, Dennis Herzog, and Reinhard Koch

Christian-Albrechts-University Kiel, Germany  
Multimedia Information Processing  
{grest, dherzog, rk}@mip.informatik.uni-kiel.de

**Abstract.** Accurate markerless motion capture systems rely on images that allow segmentation of the person in the foreground. While the accuracy of such approaches is comparable to marker based systems, the segmentation step makes strong restrictions to the capture environment, e.g. homogenous clothing or background, constant lighting etc. In our approach a template model is fitted to images by an Analysis-by-Synthesis method, which doesn't need explicit segmentation or homogenous clothing and gives reliable results even with non-static cluttered background.

## 1 Introduction

Motion capture and body pose estimation are very important tasks in many applications. Motion capture products used in the film industry or for computer games are usually marker based to achieve high quality and fast processing. A lot of research is devoted to make markerless motion capture applicable. Accurate markerless systems rely on images that allow segmentation of the person in the foreground. While the accuracy of such approaches is comparable to marker based systems [13,5], the segmentation step makes strong restrictions to the capture environment, e.g. homogenous clothing or background, constant lighting, camera setups that cover a complete circular view on the person etc. Most systems create first a visual hull from the segmented images and fit a template model afterwards by minimizing an objective function.

Our approach also fits a template model by minimizing correspondences, however it doesn't need explicit segmentation or homogenous clothing and gives reliable results even with non-static cluttered background. Additionally, less views are sufficient, as the underlying motion and body model is directly incorporated in the image processing step. While motion capture from stereo depth images already allows such complex environments [8], we present here results from a single camera view, that show the efficiency of our approach even with complex movements.

Capturing human motion by pose estimation of an articulated object is done in many approaches and is motivated from inverse kinematic problems in robotics[10]. Solving the estimation problem by optimization of an objective function is also very common [13,9,11]. Silhouette information is usually part of

this function, that tries to minimize the difference between the model silhouette and the silhouette of the real person either by background segmentation [13,11] or image gradient [12,9].

Matching feature points from one image to the next in a sequence is also a useful cue to estimate the body pose as done in [3]. However this cue alone will introduce drift, because an error in the estimation accumulates over time.

The above mentioned approaches to markerless motion capture all have in common, that the underlying movement capabilities of a human (body parts are connected by joints) are formulated directly in the optimization, while the degrees of freedom and the projection model differ. In [3] a scaled orthographic projection approximates the full perspective camera model and in [13] the minimization of 2D image point distances is approximated by 3D-line 3D-point distances.

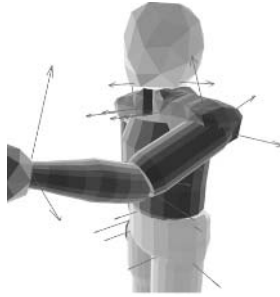
While some kind of template body model is common in most approaches, adaption of body part sizes of these template during the motion estimation is also possible like in [12]. Others assume the body model is known and fitted offline beforehand. This reduces the degrees of freedom (DOF) for the optimization significantly and allows fast and accurate estimation. In most applications it is possible to measure the size of the person before the capturing, like in sport motion analysis or in capturing motion for movies or video games.

Our approach incorporates silhouette information and point tracking using the full perspective camera model. Different cues result in different types of optimization equations. Our method minimizes errors, where they are observed and makes no approximations to the movement or projection model. Additionally it allows analytical derivations of the optimization function, which speeds up the calculation by more accuracy and less function evaluations than numerical derivatives. Therefore the approach is fast enough for real-time applications in the near future as we process images already in less than a second.

## 2 Body and Movement Model

Depending on the kind of work different body models are used for the estimation process. The models range from simple stick figures [3] over models consisting of scalable spheres (meta-balls) [12] to linear blend skinned models [2].

We use models with movement capabilities as defined in the MPEG4 standard. However not all 180 DOF are estimated, but a subset of up to 30 parameters. The MPEG4 description allows a simple change of body models and reanimation of other models with the captured motion data. An example of one model used in this work is shown in (1). The model for a specific person is obtained by silhouette fitting of a template model as described in [7]. The MPEG4 body model is a mixture of articulated objects. The movement of a point, e.g. on the hand, may therefore be expressed as a concatenation of rotations [8]. As the rotation axes are known, e.g. the flexion of the elbow, the rotation has only one degree of freedom (DOF), the angle around that axis. In addition to the joint angles there are 6 DOF for the position and orientation of the object within the global world coordinate frame. For an articulated object with  $p$  joints the transformation may be written according to [8] as:



**Fig. 1.** The body model with rotation axes shown as arrows

$$f(\boldsymbol{\theta}, \mathbf{x}) = (\theta_x, \theta_y, \theta_z)^T + (R_x(\theta_\alpha) \circ R_y(\theta_\beta) \circ R_z(\theta_\gamma) \circ R_{\boldsymbol{\omega}, \mathbf{q}}(\theta_1) \circ \dots \circ R_{\boldsymbol{\omega}, \mathbf{q}}(\theta_p))(\mathbf{x}) \tag{1}$$

where  $(\theta_x, \theta_y, \theta_z)^T$  is the global translation,  $R_x, R_y, R_z$  are the rotations around the global  $x, y, z$ -axes with Euler angles  $\alpha, \beta, \gamma$  and  $R_{\boldsymbol{\omega}, \mathbf{q}}(\theta_i), i \in \{1..p\}$  denotes the rotation around the known axis with angle  $\theta_i$ . The axis is described by the normal vector  $\boldsymbol{\omega}_i$  and the point  $\mathbf{q}_i$  on the axis with closest distance to the origin.

Equation (1) gives the position of a point  $\mathbf{x}$  on a specific segment of the body (e.g. the hand) with respect to joint angles  $\boldsymbol{\theta}$  and an initial body pose.

The first derivatives of  $f(\boldsymbol{\theta}, \mathbf{x})$  with respect to  $\boldsymbol{\theta}$  give the Jacobian matrix  $J_{ki} = \frac{\partial f_k}{\partial \theta_i}$ . The Jacobian for the movement of the point  $\mathbf{x}$  on an articulated object is

$$J = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 & \frac{\partial f}{\partial \theta_\alpha} & \frac{\partial f}{\partial \theta_\beta} & \frac{\partial f}{\partial \theta_\gamma} & \frac{\partial f}{\partial \theta_1} & \dots & \frac{\partial f}{\partial \theta_p} \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

with the simplified derivative at zero:

$$\left. \frac{\partial f}{\partial \theta_i} \right|_0 = \left. \frac{\partial R_{\boldsymbol{\omega}, \mathbf{q}}(\theta_i)}{\partial \theta_i} \right|_0 = \boldsymbol{\omega}_i \times (\mathbf{x} - \mathbf{q}_i) = \boldsymbol{\omega}_i \times \mathbf{x} - \boldsymbol{\omega}_i \times \mathbf{p}_i \tag{3}$$

where  $\mathbf{p}_i$  is an arbitrary point on the rotation axis. The term  $\boldsymbol{\omega}_i \times \mathbf{p}_i$  is also called the momentum. The simplified derivative at zero is valid, if relative transforms in each iteration step of the *Nonlinear Least Squares* are calculated and if all axes and corresponding point pairs are given in world coordinates.

### 2.1 Projection

If the point  $\mathbf{x} = (x_x, x_y, x_z)^T$  is observed by a pin-hole camera and the camera coordinate system is in alignment with the world coordinate system, the camera projection may be written as:

$$p(\mathbf{x}) = \begin{pmatrix} s_x \frac{x_x}{x_z} + c_x \\ s_y \frac{x_y}{x_z} + c_y \end{pmatrix} \tag{4}$$

where  $s_x, s_y$  are the pixel scale (focal length) of the camera in x- and y-direction, and  $(c_x, c_y)^T$  is the center of projection in camera coordinates.

We now combine  $f(\boldsymbol{\theta}, \mathbf{x})$  and  $p(\mathbf{x})$  by writing  $g(s_x, s_y, c_x, c_y, \boldsymbol{\theta}, \mathbf{x}) = p(f(\boldsymbol{\theta}, \mathbf{x}))$ .

The partial derivatives of  $g$  can now be easily computed using the chain rule. The resulting Jacobian reads as follows:

$$\begin{aligned}
 J &= \begin{bmatrix} \frac{\partial g}{\partial s_x} & \frac{\partial g}{\partial s_y} & \frac{\partial g}{\partial c_x} & \frac{\partial g}{\partial c_y} & \frac{\partial g}{\partial \theta_x} & \frac{\partial g}{\partial \theta_y} & \frac{\partial g}{\partial \theta_z} & \frac{\partial g}{\partial \theta_\alpha} & \dots & \frac{\partial g}{\partial \theta_p} \end{bmatrix} \\
 &= \begin{bmatrix} \frac{f(\boldsymbol{\theta})_x}{f(\boldsymbol{\theta})_z} & 0 & 1 & 0 & \frac{s_x}{f(\boldsymbol{\theta})_z} & 0 & s_x \frac{-f(\boldsymbol{\theta})_x}{(f(\boldsymbol{\theta})_z)^2} & \frac{\partial g_x}{\partial \theta_\alpha} & \dots & \frac{\partial g_x}{\partial \theta_p} \\ 0 & \frac{f(\boldsymbol{\theta})_y}{f(\boldsymbol{\theta})_z} & 0 & 1 & 0 & \frac{s_y}{f(\boldsymbol{\theta})_z} & s_y \frac{-f(\boldsymbol{\theta})_y}{(f(\boldsymbol{\theta})_z)^2} & \frac{\partial g_y}{\partial \theta_\alpha} & \dots & \frac{\partial g_y}{\partial \theta_p} \end{bmatrix} \tag{5}
 \end{aligned}$$

and

$$\frac{\partial g}{\partial \theta_i} = \begin{pmatrix} \frac{\partial (s_x \frac{f_x}{f_z})}{\partial \theta_i} \\ \frac{\partial (s_y \frac{f_y}{f_z})}{\partial \theta_i} \end{pmatrix} = \begin{pmatrix} \frac{s_x \left( \frac{\partial f_x}{\partial \theta_i} f(\boldsymbol{\theta})_z - f(\boldsymbol{\theta})_x \frac{\partial f_z}{\partial \theta_i} \right)}{(f(\boldsymbol{\theta})_z)^2} \\ \frac{s_y \left( \frac{\partial f_y}{\partial \theta_i} f(\boldsymbol{\theta})_z - f(\boldsymbol{\theta})_y \frac{\partial f_z}{\partial \theta_i} \right)}{(f(\boldsymbol{\theta})_z)^2} \end{pmatrix} \tag{6}$$

The partial derivatives  $\frac{\partial f}{\partial \theta_i}, i \in \{\alpha, \beta, \gamma, 1, \dots, p\}$  are given in equation (2) and  $f(\boldsymbol{\theta}) = (f_x, f_y, f_z)^T$  is short for  $f(\boldsymbol{\theta}, \mathbf{x})$ . Note that  $f(\boldsymbol{\theta})$  simplifies to  $\mathbf{x}$ , if  $\boldsymbol{\theta}$  is zero.

We minimize the distance between the projected 3D model point with its corresponding 2D image point, while in [13] the 3D-difference of the viewing ray and its corresponding 3D point is minimized. The minimization in 3D space is not optimal, if the observed image positions are disturbed by noise, as shown in [15], because for 3D points, which are farther away from the camera, the error in the optimization will be larger as for points nearer to the camera, which leads to a biased pose estimate due to the least squares solution. In [15] a scaling value was introduced, which down weights correspondences according to their distance to the camera, which is in fact very close to the equation (5).

Another relation exists to the work of [1], where the first 10 partial derivatives of Equation (5) are used for estimating the internal and external camera parameters by nonlinear optimization. This allows full camera calibration from (at best) five 2D-3D correspondences or pose from 3 correspondences. An implementation of it with an extension to the *Levenberg-Marquardt* algorithm[4], which ensures an error decrease in each iteration, is available for public in our open-source C++ library [6].

### 3 Estimating Body Pose

Assume a person, whose body model is known, is observed by a pinhole camera with known internal parameters at some time  $t$  resulting in an image  $I_t$ . Let  $X = \{\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_N\}$  be the set of model points and  $X' = \{\mathbf{x}'_0, \mathbf{x}'_1, \dots, \mathbf{x}'_N\}$  the set of their projected image points. Additionally assume that the pose of the person is known at that time, such that the projected body model aligns with the observed image as in the second image of figure 4. If the person now moves

a little and an image  $I_{t+1}$  is taken, it is possible to capture the movement by estimating the relative joint angles of the body between the frames  $I_t$  and  $I_{t+1}$ . If the image points  $\hat{X}'$  in  $I_{t+1}$  that correspond to  $X'$  are found, e.g. by some matching algorithm, the pose estimation problem is to find the parameters  $\hat{\theta}$  that best fit the transformed and projected model points to  $\hat{X}'$ , which can be formulated as follows:

$$\hat{\theta} = \arg \min_{\theta} \sum_{i=1}^N |g(\theta, \mathbf{x}_i) - \hat{\mathbf{x}}'_i|^2 \quad (7)$$

This problem is known as *Nonlinear Least Squares* and can be solved by *Newton's Method* [4]. We use the *Gauss-Newton Method* [4], which doesn't require the the second derivatives of  $g(\theta, \mathbf{x}_i)$ .

The solution is found by iteratively solving the following equation:

$$\theta_{t+1} = \theta_t - (J^T J)^{-1} J^T (G(\theta_t, \mathbf{X}) - \hat{\mathbf{X}}') \quad (8)$$

Here the Jacobian matrix  $J$  consists of all partial derivatives for all  $N$  points, where the Jacobian for a single point is given in equation (5). In case of convergence the final solution  $\hat{\theta}$  is found.

To get the initial pose, the user has to position the model manually. Because the depth is difficult to measure from a single view, markers on the floor give the user helpful information, where to position the model. Small errors in the manual positioning are not crucial, because the silhouette correspondences are correcting small errors.

### 3.1 Tracking of Image Features

For the estimation from above it is necessary to have correspondences between 2D image points and 3D model points. These can be calculated by tracking 2D features from one image to the next. Because we assume that the initial pose of the person is known as in figure 4, it is possible to get the relation between the image of the real person and the 3D model point by intersection of the feature's viewing ray and the 3D model surface using the known projection matrix. Then the same feature point has to be found within the next image and gives the necessary 2D image 3D model point correspondence. We use corners, which are tracked with the KLT feature tracker [14]. Tracking point features allows us to capture motion, which wouldn't be possible from the silhouette alone, e.g. an arm moving in front of the body like in figure 2. However as also visible, the motion estimation is not very accurate, because the assumption, that the correspondence of model and image point is given by projection of the model point, leads to an error accumulation over time. As visible the elbow position drifts away to the left. To stabilize the estimation we combine the corner tracking with silhouette information as described in the next section.

Because the movement of the arms and legs is usually larger than of the torso, we distribute feature points equally on the body of the person, such that there



**Fig. 2.** Tracking of point features (Cross marks). Boxes indicate a tracked corner. The movement of a corner over the last frames is shown as a black line.

are enough correspondences for estimation of the arm joint angles. Limiting the number and distributing the position of the tracked points is also necessary for fast computation. We achieve this by projecting the 3D model into the real image using OpenGL similar to [8], which gives directly the relation of feature points and visible body segments. In this way we can distribute the feature points equally over the visible segments.

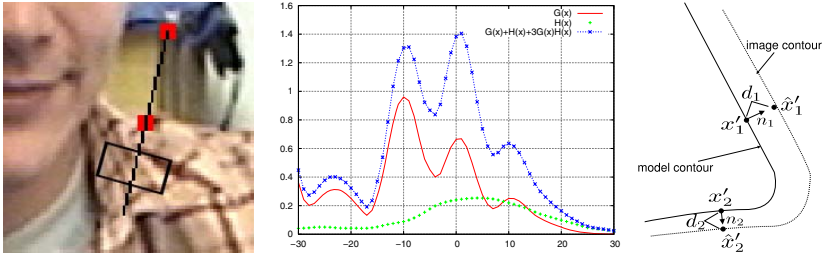
## 4 Correspondences by Silhouette

To compensate the drift we add silhouette information to our estimation. This is achieved by calculating additional 2D-3D correspondences for the model silhouette and the silhouette of the real person. In contrast to [13] we don't utilize explicit segmentation of the images in fore- and background, but use the predicted model silhouette to search for corresponding points on the real silhouette.

Previous work like [9] already took this approach by searching for a maximum grey value gradient in the image in the vicinity of the model silhouette. However we experienced that the gray value gradient alone gives often erroneous correspondences, especially if the background is heavily cluttered and the person wears textured clothes.

Therefore we also take color information into account. As the initial pose is known, it is possible to calculate a color histogram for each body segment. We use the HSL color space to get more brightness invariance. This reference histogram is then compared with a histogram calculated over a small window on the searched normal. In figure 3 the normal is shown and the rectangular window, which is used for histogram and gradient calculation. The expectation is, that the histogram difference changes most rapidly on the point on the normal of the correct correspondence, where the border between person and background is. The type of combination function was chosen by analyzing the developing of gradient and histogram values over 15 normals in different images. The actual values of the combination were then evaluated experimentally trying different values and counting the number of correct correspondences manually for about 100 silhouette points in 4 different images.

A rather difficult case is shown in figure 3 middle, which shows a plot of the maximum search along the normal of figure 3 left. The grey value gradient  $G(x)$  is



**Fig. 3.** Left: Correspondence search along the normal. Middle: The gradient ( $G(x)$ ) and histogram ( $H(x)$ ) values along the normal. Correct correspondence at 0. Right: Silhouette correspondences.

shown as a solid line, the gradient of the histogram differences  $H(x)$  as points and the combination with lines and points. As visible, the grey value gradient alone would give a wrong correspondence, while the combination yields the correct maximum at zero. The correspondences found in this way could be integrated into the estimation the same way as the correspondences from feature tracking. However, for most silhouette parts a 2D-3D point correspondence isn't correct, because of the aperture problem. For parallel lines it isn't possible to measure the displacement in the direction of the lines. Therefore we use a formulation that only minimizes the distance between the tangent at the model silhouette and the target silhouette point, resulting in a 3D-point 2D-line correspondence as visible in figure 3 right.

For a single correspondence the minimization is

$$\min_{\theta} [(g(\theta, \mathbf{x}) - \mathbf{x}')^T \mathbf{n} - d]^2 \tag{9}$$

where  $\mathbf{n}$  is the normal vector on the tangent line and  $d$  is the distance between both silhouettes, which can be computed as  $d = (\hat{\mathbf{x}}' - \mathbf{x}')\mathbf{n}$ . The point on the image silhouette  $\hat{\mathbf{x}}'$  is the closest point to  $\mathbf{x}'$  in direction of the normal. In this formulation a movement of the point perpendicular to the normal will not change the error. We calculate the normal vector as the projected face normal of the triangle, which belongs to the point  $\mathbf{x}'$ .

For a set  $\mathbf{X}$  with  $N$  points and projected image points  $\mathbf{X}'$  the optimal solution is:

$$\hat{\theta} = \arg \min_{\theta} \sum_{i=1}^N [(g(\theta, \mathbf{x}_i) - \mathbf{x}'_i)^T \mathbf{n}_i - d_i]^2$$

This is again a *Nonlinear Least Squares* problem and can be solved as above with the following Jacobian:

$$J_{ik} = \left( \frac{\partial g(\theta, \mathbf{x}_i)}{\partial \theta_k} \right)^T \mathbf{n}_i$$

Note that each of these correspondences gives one row in the Jacobian.

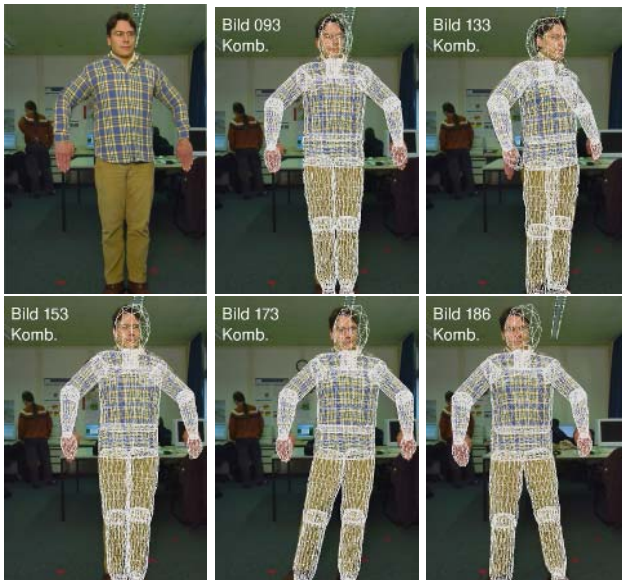


Equation (9) is an implicit description of a 2D line. The same formulation is used in [3], where the normal of the line is the image gradient and the difference  $d$  is the grey value difference. The equations for the articulated object are derived using twists, but lead to the same equations and are also solved with the *Gauss-Newton* method. However in [3] the perspective projection was approximated by a scaled orthography.

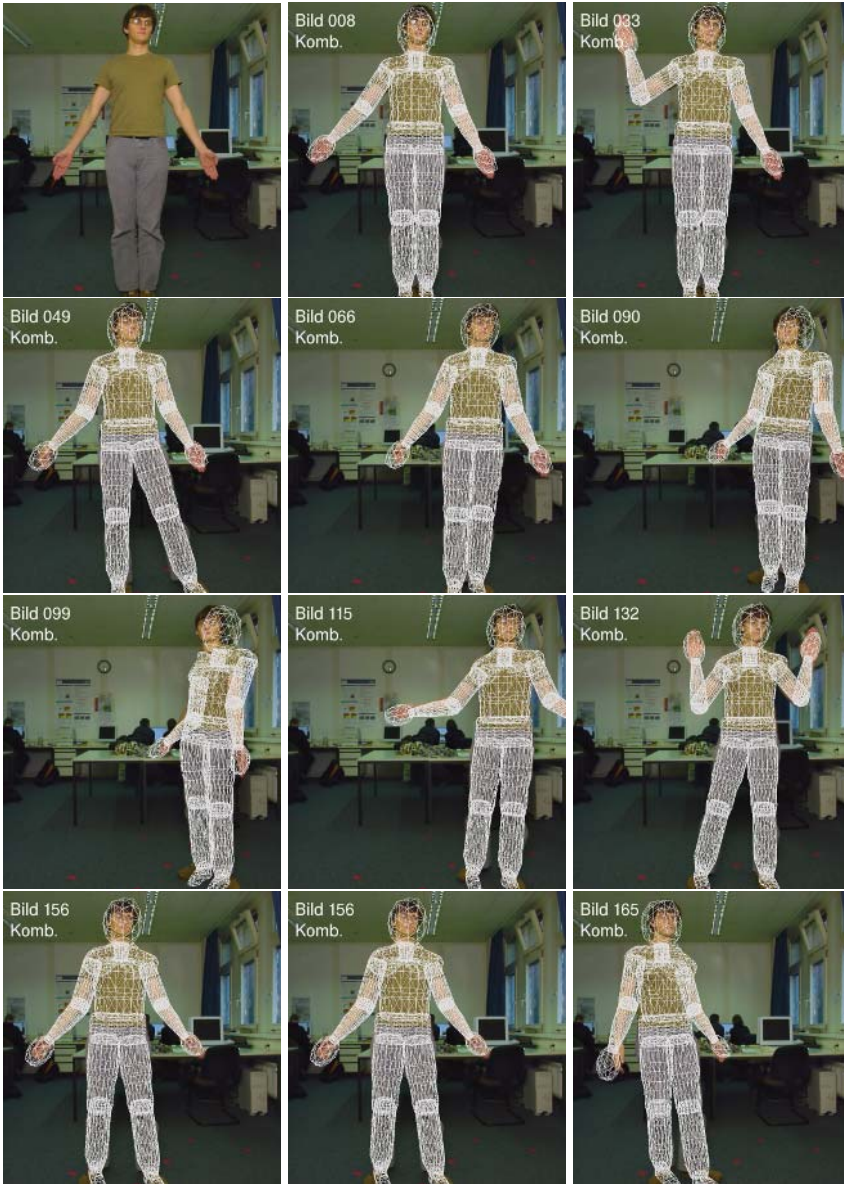
## 5 Results

Correspondences from point tracking and from silhouette difference are combined within the optimization by joining both correspondence sets together. Because we estimate pose with different correspondences, weights are added in the Least Squares steps. That way it is possible to ensure a similar influence of 3D-2D point correspondences and 3D-point 2D-line correspondences. In the following sequences 19 DOF were estimated. Five for the global position and rotation (rotation back and forth is not estimated), one for abduction of the whole shoulder complex, three for each shoulder, one for the elbow and two for each leg (twisting and abduction). Additionally the estimation was damped by a regularization term, such that a large change of joint angles in one iteration is unlikely, if only a few correspondences affect the joint. This way no correspondences for a segment lead to no change for that joint.

Figure 4 shows results for a simple movement, that consists of a rotation of the upper body and stepping aside afterwards. The person is wearing a checkered



**Fig. 4.** Original image and estimated model pose with 19 DOF



**Fig. 5.** Original image and estimated sequence with 19 DOF

shirt that exhibits lots of disturbing gray value gradients. The estimated body pose is shown in white as superimposed on the real camera image. As visible, the movement could be captured successfully from a single camera view in spite of the unknown cluttered background and the inhomogeneous clothing.

Results for a more complex movement for a different person are shown in figure 5. The person is wearing a T-shirt and the background is non-static and cluttered again. Movement between frames is quite large, because capturing was done with 7 fps, while the person was moving at normal speed. Even though the shoulder and the upper arm are completely hidden during some frames, the movement could be captured correctly.

## 6 Conclusions

We showed how estimation of human movement can be derived from point transformations of an articulated object. Our novel approach uses a full perspective camera model and minimizes errors where they are observed, i.e. in the image plane. That way we overcome limitations and approximations of previous work. No explicit segmentation of the images is needed. Correct correspondences are found in spite of cluttered non-static background and normal clothing. Motion with 19 DOF could be estimated that even contained partially hidden body parts. Movements parallel to the optical axis of the camera are not possible to estimate accurately from a single view, e.g. movement of the arms back and forth.

The estimation method is fast enough to fulfill real-time conditions in the near future as processing of one frame is done in less than a second. Ongoing work is to combine this approach with body pose estimation from depth images.

## References

1. H. Araujo, R. Carceroni, and C. Brown. A Fully Projective Formulation to Improve the Accuracy of Lowe's Pose Estimation Algorithm. *Comp. Vis. and Image Understanding*, 70(2), 1998.
2. M. Bray, E. Koller-Meier, P. Mueller, L. Van Gool, and N. N. Schraudolph. 3D Hand Tracking by Rapid Stochastic Gradient Descent Using a Skinning Model. In *CVMP*. IEE, March 2004.
3. C. Bregler and J. Malik. Tracking people with twists and exponential maps. In *Proceeding IEEE CVPR*, pages 8–15, 1998.
4. Edwin K.P. Chong and Stanislaw H. Zak. *An Introduction to Optimization, Second Edition*, chapter 9. Wiley, 2001.
5. Lars Mündermann et al. Validation of a markerless motion capture system for the calculation of lower extremity kinematics. In *Proc. American Society of Biomechanics*, Cleveland, USA, 2005.
6. J.-F. Evers-Senne, J.-M. Frahm, D. Grest, K. Köser, B. Streckel, J. Woetzel, and J.-F. Woelk. Basic Image AlgorithmS (BIAS) open-source-library, C++. [www.mip.informatik.uni-kiel.de/Software/software.html](http://www.mip.informatik.uni-kiel.de/Software/software.html), 2005.
7. D. Grest, D. Herzog, and R. Koch. Human Model Fitting from Monocular Posture Images. In *Proc. of VMV*, Nov. 2005.
8. D. Grest, J. Woetzel, and R. Koch. Nonlinear Body Pose Estimation from Depth Images. In *Proc. of DAGM*, Vienna, Sept. 2005.
9. I. Kakadiaris and D. Metaxas. Model-Based Estimation of 3D Human Motion. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 22(12), 2000.

10. R.M. Murray, Z. Li, and S.S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.
11. M. Niskanen, E. Boyer, and R. Horaud. Articulated motion capture from 3-D points and normals. In *CVMP*, London, 2005.
12. Ralf Plaenkers and Pascal Fua. Model-Based Silhouette Extraction for Accurate People Tracking. In *Proc. of ECCV*, pages 325–339. Springer-Verlag, 2002.
13. B. Rosenhahn, U. Kersting, D. Smith, J. Gurney, T. Brox, and R. Klette. A System for Marker-Less Human Motion Estimation . In W. Kropatsch, editor, *DAGM*, Wien, Austria, Sept. 2005.
14. C. Tomasi and T. Kanade. Detection and tracking of point features. Technical report, Carnegie Mellon University, Pittsburg, PA, 1991.
15. B. Wetegren, L.B. Christensen, B. Rosenhahn, O. Gran ert, and N. Krüger. Image uncertainty and pose estimation in 3d euclidian space. *Proceedings DSAGM*, pages 76–84, 2005.

# Pose Estimation from Uncertain Omnidirectional Image Data Using Line-Plane Correspondences

Christian Gebken, Antti Tolvanen, and Gerald Sommer

Institut für Informatik, CAU Kiel  
Christian-Albrechts-Platz 4, 24118 Kiel, Germany  
{chg, ant, gs}@ks.informatik.uni-kiel.de

**Abstract.** Omnidirectional vision is highly beneficial for robot navigation. We present a novel perspective pose estimation for omnidirectional vision involving a parabolic central catadioptric sensor using line-plane correspondences. We incorporate an appropriate and approved stochastic method to deal with uncertainties in the data.

## 1 Introduction

Roughly speaking, rigidly moving an object in 3D such that it comes into agreement with 2D-sensory data of a camera, is called 2D-3D pose estimation [3]. It is a well-studied subject in the case of pinhole cameras for which sophisticated methods exist, see e.g. [13].

Single viewpoint catadioptric vision sensors combine a conventional camera with one or two mirrors and provide a panoramical view of  $360^\circ$ . Our device is a folded system consisting of two parabolic mirrors and a lens to provide a scaled, approximately orthographic projection from the main mirror. It can equivalently be treated as a single mirror device, see Nayar et al [10].

The most significant advantages of omnidirectional vision are related to navigation. For example, methods of movement estimation from triangulation, topology map and feature flow based methods [1,4,6] for localization give good results on the estimation of movements between frames and the localization from the visual information. Apart from those methods, 2D-3D pose estimation gives the complete pose information, that is more than a 2D-position in a plane. Since it includes all six possible degrees of freedom (DOF), it can account for effects like pitch, roll and yaw. Therefore, it represents an appropriate method for navigation, also on uneven surfaces. Furthermore, in the case of omnidirectional pose estimation, the object does not need to be observed within some narrow spatial angle, but may surround the visual sensor itself. This implies a number of advantages. First, an object remains on the image plane under most movements, which is desirable for tasks such as tracking. Second, the accuracy of the estimated pose should be superior, as for example in triangulation, which performs best if the used landmarks are seen at right angles. Still, surprisingly little research was done on omnidirectional pose estimation.

Our objective was to develop accurate pose estimation for omnidirectional vision given imprecise image features, i.e. 2D-sensory data. The motivation was

to take the opportunity to extend approved pinhole methods to the omnidirectional case by exploiting simple existing geometrical relations for parabolic mirrors. The stochastic is one of the fundamental aspects of this work; to account for invariable uncertainties in observational data we consequently decided on a least squares adjustment parameter estimation. The concept of our approach is a well-tried amalgamation of geometry with stochastic via Geometric Algebra.

One assumption we make is to have 3D-models of the interesting objects we observe in the images. This can be an ordinary object like a table or it is a model describing the environment. Secondly, we assume to know the one-to-one correspondences between the model features and the image features.

Note that with this contribution we extend our previous work by using line models instead of point models. The matching image entities are therefore lines. Recognition and localization is simpler for lines than for points, since those are intrinsically higher-dimensional structures. Localization is more precise for lines, as well. Regarding regular structures, like a skyscraper, it is more efficient to have line models than to store the corners of each single window. We can state that the existence of key points, e.g. corners, mostly inheres with the existence of lines which are then the preferable entities.

In the next section, we discuss the pose estimation and all related topics in some detail. In section 3 we present experimental results. Finally, we give conclusions in section 4.

## 2 Omnidirectional 2D-3D Pose Estimation

In general, perspective 2D-3D pose estimation consists of determining the orientation and position of an internally calibrated camera [5], given a 3D-model of an object in a scene and a set of 2D-correspondence features (points, lines, curves) from an image of that scene. The model serves as a reference to an external (world) coordinate system. If we determine the model's position and orientation with respect to the camera coordinate system, we are able to infer the pose of the camera, given by a rigid body motion (RBM). Specifically, we estimate the RBM, such that the model lines come to lie on the projection planes of the underlying image lines.

We use the Geometric Algebra  $G_{4,1}$  of the conformal embedding of Euclidean 3D-space as introduced in [2,8]. A similar pose estimation could also be done solely in Euclidian 3D-space, but we obtain certain advantages when working in  $G_{4,1}$ : geometric entities as points, spheres, planes or lines and geometric operators as an inversion or an RBM are basic elements of  $G_{4,1}$ . They have thus a natural representation in terms of (sparse) vectors of  $\mathbb{R}^{25}$ . Moreover, incidence relations, as needed to decide whether a line lies on a projection plane, can be evaluated by means of bilinear algebra products. Nevertheless, for understandability to unfamiliar readers, and since it is not the main subject of this work, we make explicit use of Geometric Algebra in just one passage. In practice we employ the framework of Geometric Algebra throughout all steps of our method. A

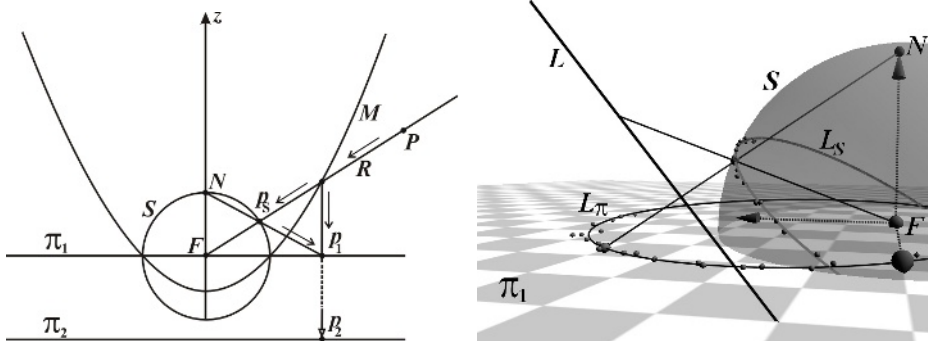
general introduction to the estimation of geometric entities and operators from uncertain data using Geometric Algebra can be found in [12].

Our method consists of three steps: from those pixels corresponding to model lines, we compute the projection planes with associated uncertainties. In a second step, a simple algorithm is used to do prior rotation estimation being a first and rough guess at the rotational part of the desired RBM. As a result the model will be aligned such that its lines are nearly parallel to the respective projection planes. Next, an iterative method estimates the entire pose now taking also the plane uncertainties into account.

Before we explain those steps we give an overview regarding catadioptric imaging with a parabolic mirror.

### 2.1 Omnidirectional Imaging

Despite our interest in the mapping of lines to the image we begin with the point case. The omnidirectional camera setup we consider consists of a camera focused at infinity, which looks at a parabolic mirror centered on its optical axis. This setup is shown in figure 1. A light ray emitted from point  $P$  that

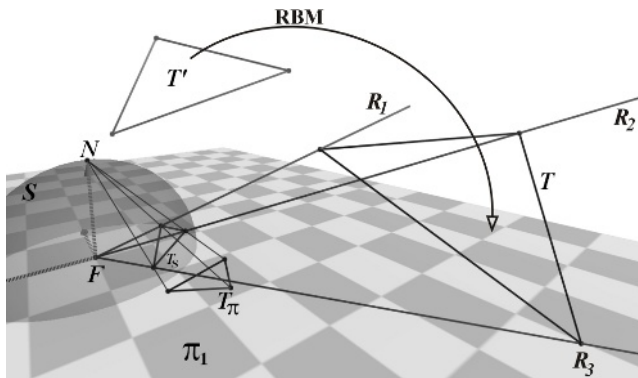


**Fig. 1.** **Left:** Mapping (cross-section) of a point  $P$ : the image planes  $\pi_1$  and  $\pi_2$  are identical. **Right:** Mapping of line  $L$  to  $L_\pi$  via great circle  $L_S$  on  $S$ . As an example, scattered image data belonging to  $L_\pi$  is shown.

would pass through the focal point  $F$  of the parabolic mirror  $M$ , is reflected parallel to the central axis of the parabolic mirror, to give point  $p_2$ . Since all such reflected rays are parallel, a camera placed beneath the mirror focused at infinity will generate a sharp image on plane  $\pi_2$ . Here, we use the simplification that a projection to sphere  $S$  with a subsequent stereographic projection to  $\pi_1$  produces an identical image on  $\pi_1$ . Accordingly, point  $P$  maps to  $p_s$  and further to  $p_1$ , see figure 1. Together with the right side of figure 1 it is intuitively clear that infinitely extended lines form great circles on  $S$ . Moreover, a subsequent stereographic projection, being a conformal mapping, results in circles<sup>1</sup> on the image plane, which are then no more concentric. For details refer to [7,14].

<sup>1</sup> An often occurring special case is a (vertical) line parallel to the optical axis, which is mapped to an image line, i.e. a circle with infinite radius, passing the origin  $F$ .

Note that given image points we can apply all mentioned steps in reverse order to obtain the corresponding projection rays. Similarly, we can compute the corresponding projection plane from two image points, since their back-projections on sphere  $S$  can always be taken to form a great circle, the plane of which represents the projection plane.



**Fig. 2.** Pose estimation: fitting a triangle model to respective projection rays/planes in 3D

To illustrate the whole pose estimation process consider figure 2: three image points build the triangle-like<sup>2</sup> object  $T_\pi$ , which is stereographically back-projected to  $T_S$  on  $S$ . In conjunction with  $F$ , we obtain the projection rays or planes, respectively. The correct RBM then moves the model triangle  $T'$ , so that either the model points comes to lie on the projection rays or in our case, the model lines come to lie on the projection planes.

## 2.2 Estimating Projection Planes

In order to perform the line-plane fitting artificial plane observations are built from our initial image point observations: for each set of image points that corresponds to a model line, the respective projection plane is evaluated. This is done in very much the same way as the circle estimation described in [12], where the stochastic estimation method underlying this work applies as well. A circle can be defined in terms of a plane, a center and a radius; the plane estimation can thus be restricted to three components representing the normal of the circle plane.

We assume that all image points initially have the same 2D-uncertainty given by a  $2 \times 2$  identity covariance matrix, i.e. we assume an pixel error of one in row

<sup>2</sup> In the figure  $T_\pi$  and  $T_S$  are drawn as triangles, although their sides are supposed to be arcs rather than lines.



and column. Since the planes have to be estimated from the stereographically back-projected image points, see figure 1, we must move the involved image points to the projection sphere  $S$ . This is done by an inversion of the image points in a certain sphere. The points thereby obtain distinct 3D-uncertainties accounting for the imaging geometry. The mapping of a far image point to a point close to the North Pole  $N$  of  $S$ , for example, is less affected by noise and will thus inhere with a higher confidence, see figure 1. Mathematically, the uncertainties are computed using standard error propagation, where we profit from the inversion being an element of  $G_{4,1}$ .

Since our estimation method is capable of providing a covariance matrix regarding the estimated entity, we obtain a  $3 \times 3$  covariance matrix for each plane. Those are then to be reinput to our pose estimation algorithm.

### 2.3 Prior Model Alignment

Estimation problems mostly require a linearization of condition or constraint functions and one usually ends up with an iterative method, as we do. This raises the need for a sufficiently good initial estimate regarding the iterations. The prior model alignment provides such a starting point at very low costs. Moreover, it shortens the overall computation time. We like to rotate the model such that the unit direction vectors  $\{\hat{r}_{1..N}\}$  of its lines lie on the respective planes. Here, a prerequisite is to have the normal vectors  $\{\hat{n}_{1..N}\}$  of all planes belonging to visible model lines. We search for a rotation matrix  $R$  such that  $(\forall i): \hat{n}_i^T R \hat{r}_i = 0$ .

By Rodrigues's formula (1840) we know that the rotation matrix  $R$  regarding a rotation of angle  $\theta$  around unit vector  $\hat{a} = (a_1, a_2, a_3)^T$  can be expressed by an exponential map of  $A = ((0, a_3, -a_2)^T (-a_3, 0, a_1)^T (a_2, -a_1, 0)^T)$ :  $R = \exp(\theta A)$  which is  $R = I_3 + \sin \theta A + (1 - \cos \theta) A^2$ . For small angles we obtain  $R = I_3 + \theta A$ . With this relation and due to the skew symmetric structure of  $A' = \theta A$  it is possible to solve for  $a' = (\theta a_1, \theta a_2, \theta a_3)^T$ , where each line-plane pair gives one line  $\hat{n}_i^T A' \hat{r}_i = -\hat{n}_i^T \hat{r}_i$  in an overdetermined system of linear equations. Every run of this procedure yields a rotation matrix, the concatenation of which gives the desired rotation matrix  $R$ . Once, the rotated lines are close enough to the planes w.r.t. some threshold the procedure can be stopped.

### 2.4 Stochastic Estimation Method

In this section we concisely introduce our two parameter estimation methods, the common *Gauss-Markov* method and the most generalized case of *least squares adjustment*, the *Gauss-Helmert* method. Both are founded on the respective homonymic linear models, cf. [9]. The word 'adjustment' puts emphasis on the fact that an estimation has to handle redundancy in observational data appropriately, e.g. to weight unreliable data to a lesser extend. The principle of least squares adjustment, i.e. to minimize the sum of squared weighted errors  $\Delta y_i$ , is often denoted as

$$\sum_i \Delta y_i^T \Sigma_{y_i}^{-1} \Delta y_i \longrightarrow \min, \tag{1}$$

where  $\Sigma_{y_i}$  is a covariance matrix assessing the confidence of  $y_i$ .

Let  $\{\mathbf{b}_{1..N}\}$ <sup>3</sup> denote a set of  $N$  observations. Each observation  $\mathbf{b}_i$  is associated with an appropriate covariance matrix  $\Sigma_{\mathbf{b}_i}$  denoting the confidence. An entity, parameterized by a vector  $\mathbf{p}$ , is to be fitted to the observational data. Consequently, we define a condition function  $\mathbf{g}(\mathbf{b}_i, \mathbf{p})$ , which is supposed to be zero if the observations and the entity in demand fit. If we know an already good estimate  $\hat{\mathbf{p}}$  we can make a linearization yielding  $(\partial_{\mathbf{p}} \mathbf{g})(\mathbf{b}_i, \hat{\mathbf{p}}) \Delta \mathbf{p} + \mathbf{g}(\mathbf{b}_i, \hat{\mathbf{p}}) \approx 0$ , hence with  $\mathbf{U}_i = (\partial_{\mathbf{p}} \mathbf{g})(\mathbf{b}_i, \hat{\mathbf{p}})$  and  $\mathbf{y}_i = -\mathbf{g}(\mathbf{b}_i, \hat{\mathbf{p}})$ :  $\mathbf{U}_i \Delta \mathbf{p} = \mathbf{y}_i + \Delta y_i$ , which exactly matches the linear *Gauss-Markov* model. The minimization of equation (1) in conjunction with the Gauss-Markov model leads to the *best linear unbiased estimator*<sup>4</sup>. Note that we have to leave the weighting out in equation (1), since our covariance matrices  $\Sigma_{\mathbf{b}_i}$  do not match the  $\Sigma_{y_i}$ . Subsequently, we derive a model which includes the weighting.

If we take our observations as estimates, i.e.  $\{\hat{\mathbf{b}}_{1..N}\} = \{\mathbf{b}_{1..N}\}$ , we can make a complete Taylor series expansion of first order at  $(\hat{\mathbf{b}}_i, \hat{\mathbf{p}})$  yielding

$$(\partial_{\mathbf{p}} \mathbf{g})(\hat{\mathbf{b}}_i, \hat{\mathbf{p}}) \Delta \mathbf{p} + (\partial_{\mathbf{b}} \mathbf{g})(\hat{\mathbf{b}}_i, \hat{\mathbf{p}}) \Delta \mathbf{b}_i + \mathbf{g}(\hat{\mathbf{b}}_i, \hat{\mathbf{p}}) \approx 0.$$

Similarly, with  $\mathbf{V}_i = (\partial_{\mathbf{b}} \mathbf{g})(\hat{\mathbf{b}}_i, \hat{\mathbf{p}})$  we obtain  $\mathbf{U}_i \Delta \mathbf{p} + \mathbf{V}_i \Delta \mathbf{b}_i = \mathbf{y}_i$ , which exactly matches the linear *Gauss-Helmert* model. Note, that the error term  $\Delta y_i$  has been replaced by the linear combination  $\Delta y_i = -\mathbf{V}_i \Delta \mathbf{b}_i$ : the Gauss-Helmert differs from the Gauss-Markov model, because the observations have become random parameters and are thus allowed to undergo small changes  $\Delta \mathbf{b}_i$  to compensate for errors. But changes have to be kept minimal, as observations represent the best available. This is achieved by replacing equation (1) with

$$\sum_i \Delta \mathbf{b}_i^T \Sigma_{\mathbf{b}_i}^{-1} \Delta \mathbf{b}_i \longrightarrow \min, \tag{2}$$

where  $\Delta \mathbf{b}_i$  is now considered as error vector. The minimization of (2) subject to the Gauss-Helmert model can be done using Lagrange multipliers, cf. [9].

Due to outstanding convergence properties we start iterating with the Gauss-Markov method. At the optimum we start the slower Gauss-Helmert method which ultimately adjusts the estimate according to the given uncertainties  $\Sigma_{\mathbf{b}_i}$ .

## 2.5 Perspective Line-Plane Pose Estimation

Here we derive geometric constraint equations for the stochastic estimation methods presented in the previous section. The respective expressions come from the Geometric Algebra of conformal space  $G_{4,1}$ . A similar methodology was

<sup>3</sup> We use the abbreviation  $\{\mathbf{b}_{1..N}\}$  for a set  $\{\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_N\}$ .

<sup>4</sup> It has been shown in [9] that different approaches, namely *least squares*, *maximum likelihood* and the linear approach, equally lead to the best linear unbiased estimator.

chosen by Rosenhahn et al [13]. The products used in the following are the *geometric product*, which is the main algebra product, and the *outer product*, which is in no way related to the outer product of matrices. The geometric product is denoted by juxtaposition and the outer product by  $\wedge$ .

Let  $P$  be a projection plane, see section 2.2. For any line  $L$  lying on  $P$ , we have  $P \wedge L = 0 \in G_{4,1}$ . A model line  $L'$  is transformed by an RBM represented by  $V$ , say, via the operation  $V L' \tilde{V}$ , where the *reverse*  $\tilde{V}$  is similar to conjugation in quaternions. Therefore, if we have estimated the correct RBM  $V$ , a model line  $L'$  with corresponding projection plane  $P$  has to satisfy  $P \wedge (V L' \tilde{V}) = 0$ .

Due to the numerical representation of  $G_{4,1}$ , we can identify our elements  $P$ ,  $L'$  and  $V$  with particular vectors  $\mathbf{p} \in \mathbb{R}^3$ ,  $l' \in \mathbb{R}^6$  and  $\mathbf{v} \in \mathbb{R}^8$ . For example,  $p$  simply denotes the normal vector of the plane represented by  $P$ . Moreover, each algebra product is a bilinear function and can be formulated equivalently using a certain tensor, cf. [12]. By contracting the constituent tensors the condition function  $\mathbf{g}$  of the previous section can be written in the following way

$$\mathbf{g}^t(\mathbf{p}, \mathbf{v}) := \sum_{i,j,k,l} v^i v^j p^k l'^l Q^t_{ijkl} = 0, \quad t \in \{1 \dots 4\}. \quad (3)$$

Algebraically, the constraint  $P \wedge L$  may only be nonzero in four of its  $2^5 = 32$  components, i.e. we have  $t \in \{1 \dots 4\}$ . The observations and parameters are  $\mathbf{p}$  and  $\mathbf{v}$ , respectively. Hence, differentiating would yield the matrices  $\mathbf{V}$  and  $\mathbf{U}$  required in section 2.4. Note that the eight components of  $V$  are an overparameterization of the six DOF of an RBM, such that we need to include the RBM-constraint  $V \tilde{V} = 1$  in the minimization process, which also turns out to be a bilinear function of the components of  $V$ . Such additional constraints can be readily included in our parameter estimation methods.

### 3 Experimental Results

Two real world experiments were performed using an imaging system consisting of a *Kamerawerk Dresden Loglux i5* camera and *Remote Reality Netvision 360* catadioptric sensor with a parabolic mirror. The aim of the experiments was to test object pose estimation and navigation and the robustness of the used methods in these tasks. As intrinsic calibration parameters we used the 40 mm mirror radius and 16.7 mm focal length for the main mirror given by the manufacturer. The projection of the sensor was assumed to be exactly orthographic and the whole mirror was assumed to be visible in the image. Images were acquired in  $1280 \times 1024$  resolution where the actual size of the omnidirectional image is the area of a circle with 492 pixel radius corresponding to the 40 mm mirror radius. The radius and the center of the image were determined from the sum of images used in the experiments. No other calibration was done. The image lines were extracted manually with seven points/line.

In the first experiment a model house was moved with a robot arm to 21 different locations. The robot arm gives ground truth of the translations between

the different locations with millimeter accuracy. The magnitude of these translations was between 7.7 cm and 62.4 cm and the distance of the model house to the optical center of the catadioptric sensor was between 31.4 cm and 82.8 cm. The house dimensions in cm are  $21 \times 15 \times 21$ . From the 21 acquired images the RBMs of the model house from the optical center were estimated. These estimates were used to get the relative translation estimates between the different model house positions. The results are given in table 1.

**Table 1.** The errors of the house pose estimation

	Abs. error [mm]	Rel. error [%]	Angle error [°]
mean	10.4	3.5	0.9
std	4.8	1.7	0.4
min	0.9	0.4	0.12
max	21.3	11.5	2.4

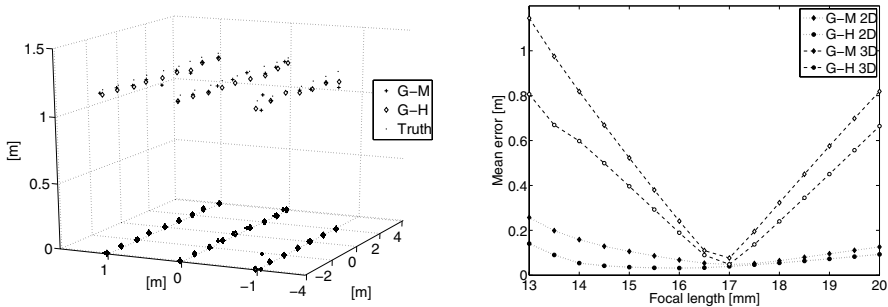
In the second experiment the sensor was moved to 25 different positions in a hallway. The model was defined by lines clearly visible in most of the images. The other criterion was reasonable measurability needed to create the model. The walls were assumed to be perpendicular to the floor and all corners to be right angled. With these assumptions we get roughly 2 cm accuracy for positions of the model lines. The model consisted of total 51 lines from which on average 20 lines were visible in an image. The maximum orthogonal distance of these lines was 18.1 m, minimum 3.8 m and the sensor movements were made on  $8 \times 2$  m<sup>2</sup> area inside the model. The results for the error in the position are given in table 2 for Gauss-Markov (G-M) and Gauss-Helmert (G-H) methods in 2D and 3D. Figure 3 on left represents the results for G-M and G-H methods and the groundtruth (Truth) in 3D.

In addition to the pose estimation with the given parameters we studied the robustness of the used methods in respect to the change of the focal length of the mirror (see figure 3 on right). It can be seen that the G-H method is always better than G-M and slightly more robust. 2D estimation works always better as the error source is one dimension smaller and the estimation relies mostly on the vertical world lines whose image remains almost unchanged with the change of focal length. Using 0.1 mm steps for the focal length gives the most accurate results for G-M 3D 5.8 cm with focal length 16.8 mm and for G-H 3D 4.2 cm with focal length 16.9 mm.

Comparisons for 3D results are hard due to the limited number of usable publications. The results in 2D are comparable results to those given by Aliaga [1]; he obtained an average planar error of 2.8 cm within a room of 5 meters diameter using a triangulation method and with exact calibration of the system. Cauchois et al [4] reached about 1 cm accuracy in 2D using an image database method with a conical mirror and a room of  $2 \times 3$  m<sup>2</sup>.

**Table 2.** The errors of the navigation

	Mean error [cm]	RMS error [cm]	min [cm]	max [cm]
G-M 3D	7.6	9.4	3.6	32.2
G-M 2D	5.1	7.7	0.4	32.0
G-H 3D	6.4	6.5	2.7	8.3
G-H 2D	3.5	3.9	0.5	5.7



**Fig. 3. Left:** navigation results. The 3D positions are also projected to plane for clarity. **Right:** focal length vs. mean error.

## 4 Conclusions

The objective of this work was to realize 2D-3D pose estimation for omnidirectional vision using line-plane correspondences. The pose was computed by a stochastic estimation method, which accounts for uncertainties in the image data.

The experimental results clearly demonstrate that our combination of 2D-3D pose estimation with omnidirectional vision does provide exact results for navigation within relatively big environments. The results of our house experiments show that we still obtain good results, if we utilize our method for conventional 2D-3D object pose estimation.

Especially the 2D-navigation was found out to be very robust in respect to changes of focal length. The change of the focal length scales the image radially. Since the images of vertical world lines are radial lines in the image they are invariant in this scaling. On the other hand the positions of image points on the radial lines are not invariant. This motivates studies on the differences in the robustness of point-line and line-plane pose estimation in 2D-navigation.

In the future we would like to automate the point extraction from the image in order to construct a ready to use method for robotics. This is plausible as the calculation time for the pose estimation (including 3D-visualization) is under 1

second using a scripting language (CLUCalc, see [11]) on a 3 GHz Pentium 4 computer.

## References

1. Daniel G. Aliaga. Accurate catadioptric calibration for real-time pose estimation of room-size environments. In *International Conference on Computer Vision (ICCV)*, pages 127–134, 2001.
2. Pierre Angles. Construction de revêtements du groupe conforme d'un espace vectoriel muni d'une «métrique» de type  $(p, q)$ . *Ann. Inst. Henri Poincaré*, 33(1):33–51, 1980.
3. Adnan Ansar and Konstantinos Daniilidis. Linear pose estimation from points or lines. In *7th European Conference on Computer Vision (ECCV)*, Copenhagen, Denmark, pages 282–296, 2002.
4. Cyril Cauchois, Eric Brassart, Laurent Delahoche, and Cyril Drocourt. Spatial localization method with omnidirectional vision. In *11th IEEE International Conference on Advanced Robotics (ICAR)*, Coimbra, Portugal, pages 287–292, 2003.
5. Olivier Faugeras. *Three-Dimensional Computer Vision*. MIT Press, 1993.
6. Jos Gaspar and Jos Santos-Victor. Vision-based navigation and environmental representations with an omni-directional camera. *IEEE Transactions on Robotics and Automation*, 16(6):890–898, 2000.
7. Christopher Geyer and Kostas Daniilidis. Catadioptric projective geometry. *International Journal of Computer Vision*, 45(3):223–243, 2001.
8. D. Hestenes and G. Sobczyk. *Clifford Algebra to Geometric Calculus: A Unified Language for Mathematics and Physics*. Reidel, Dordrecht, 1984.
9. K.-R. Koch. *Parameter Estimation and Hypothesis Testing in Linear Models*. Springer, 1997.
10. Shree K. Nayar and Venkata Peri. Folded catadioptric cameras. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, Ft. Collins, CO, USA, pages 2217–, 1999.
11. C. Perwass, C. Gebken, and D. Grest. CLUCalc. <http://www.clucalc.info/>, 2006.
12. C. Perwass, C. Gebken, and G. Sommer. Estimation of geometric entities and operators from uncertain data. In *27. Symposium für Mustererkennung, DAGM 2005, Wien, 29.8.-2.9.005*, number 3663 in LNCS. Springer-Verlag, Berlin, Heidelberg, 2005.
13. B. Rosenhahn and G. Sommer. Pose estimation in conformal geometric algebra, part I: The stratification of mathematical spaces, part II: Real-time pose estimation using extended feature concepts. *Journal of Mathematical Imaging and Vision*, 22:27–70, 2005.
14. Antti Tolvanen, Christian Perwass, and Gerald Sommer. Projective model for central catadioptric cameras using clifford algebra. In *27. Symposium für Mustererkennung, DAGM 2005, Wien, 29.8.-2.9.005*, volume 3663 of LNCS, pages 192–199. Springer-Verlag, Berlin, Heidelberg, 2005.

# Kernel Particle Filter for Visual Quality Inspection from Monocular Intensity Images

Dirk Stöbel and Gerhard Sagerer

Applied Computer Science, Faculty of Technology,  
Bielefeld University, P.O. Box 100131, 33501 Bielefeld, Germany  
{dstoesse, sagerer}@techfak.uni-bielefeld.de

**Abstract.** Industrial part assembly has come a long way and so has visual quality inspection. Nevertheless, the key issue in automated industrial quality inspection, i.e. the pose recovery of the objects under inspection, is still a challenging task for assemblies with more than two rigid parts. This paper presents a system for the pose recovery of assemblies consisting of an arbitrary number of rigid subparts. In an offline stage, the system extracts edge information from CAD models. Online, the system uses a novel kernel particle filter to recover the full pose of the visible subparts of the assembly under inspection. The accuracy of the pose estimation is evaluated and compared to state-of-the-art systems.

## 1 Introduction

Visual inspection is an important part of most manufacturing work flows. In the context of large scale production, great effort is being put into automating this task. Especially car manufacturers regard automated visual inspection as a vital part of their quality assurance programs. This paper proposes a system targeted to be part of an industrial production environment in which assemblies are handled by industrial robots and inspected with a monocular camera. The system's purpose is to measure the pose parameters of a group of parts that have been assembled to a complex aggregate. The resulting aggregate pose might afterwards be fed into a detection stage to pick out fault configurations.

The pose estimation problem lies at the very heart of any automated visual inspection task. The more accurate aggregate pose parameters can be determined, the more sensitive and specific subsequent fault detection stages can distinguish correct from unwanted assembly configurations. Presently, the problem can be solved fast and accurately [1], [2], given the correspondence between a set of characteristic object features and measurements such as intensity images, laser-range data, etc. Unfortunately, this kind of information is usually not available beforehand. Establishing correspondences, e.g. by means of Iterative Closest Point approaches [3], becomes increasingly complex when dealing with *articulated objects* like assemblies, i.e. multi-body systems whose rigid parts are connected by joints. With an increasing number of parts, complexity arises from inter-part occlusion and growing ambiguity in the assignment of observed image features to specific assembly parts. This might be one reason why, to our

knowledge, it has not been attempted yet to estimate the pose parameters of multi-part assemblies from monocular images in the context of automated visual inspection. The approach presented here employs a kernel particle filter that doesn't need a priori correspondence information. The tradeoff is reduced accuracy, though it will be shown in Sect. 5 that it does not suffer too much compared to results reported for state-of-the-art systems which recover the pose of single rigid bodies.

*Kernel particle filters* (KPFs) have recently been introduced by Chang & Ansari [4] who use it in the field of visual tracking. KPFs combine particle filters with a mean shift algorithm. The particle filter first creates a sample set representation of a posterior probability density. Afterwards, mean shift iterations move the particles towards the modes of the posterior probability density. It is shown in [4] that KPFs maintain a compact representation of the modes of the posterior, especially in high dimensional state spaces.

The work presented here is based on the approach of Schmidt et al. [5] who use a KPF to estimate ten pose parameters of a human's upper body model from monocular color images. Unfortunately, their KPF operates on an unnormalized state space of  $d$  dimensions. Accordingly, for the inherent kernel density estimation,  $d$  bandwidth parameters must be specified manually. Within industrial applications each manually set parameter increases the complexity of system operation. This is why the KPF presented in this paper uses a state space that is variance normalized as it was suggested in [6]. This strategy reduces the number of bandwidth parameters to one. Nevertheless, preliminary experiments revealed that the remaining bandwidth parameter was notoriously hard to specify. The system proposed in this paper therefore enhances the KPF with a variable bandwidth selection technique. The technique was introduced by Comaniciu et al. [7] and was originally used in the context of mean shift image segmentation. As it will be shown in the evaluation section, this enhancement allows to use the same bandwidth parameter setting for different measurement scenarios.

This paper makes three major contributions: First, the proposed system uses a unique kernel particle filter that operates on a variance normalized state space and further enhances the approach of Schmidt et al. [5] with a variable bandwidth selection scheme. Second, the system is the first to employ kernel particle filtering in the field of automated visual inspection. It recovers the pose of articulated objects without explicitly solving the correspondence problem. Third, the paper provides a detailed evaluation of the system's measurement accuracy and characteristics based on real world images. The system is shown to be competitive to state-of-the-art systems that focus on the single rigid body case.

The remainder of the paper is organized as follows: The next section presents a brief system overview. Section 3 then specifies how features extracted from CAD descriptions are used to model the appearance of assemblies under varying pose parameterizations. Afterwards, the new kernel particle filter is detailed in Sect. 4. A thorough evaluation of the system's measuring accuracy and a comparison to state-of-the-art systems is given in Sect. 5. Finally, an outlook concludes this contribution.



## 2 System Overview

Our system operates in two stages. During an *offline* stage, contour edge features are automatically extracted from 3D CAD data. This is done for each rigid assembly part. The resulting feature models are manually put together to a *kinematic tree*, i.e. a tree-like representation of rigid parts that are connected by joints. Within the system, kinematic trees represent ideal assemblies and the joint parameter ranges that reflect possible pose changes of rigid parts.

The second stage of system operation is an *online* stage. A monocular camera takes images of an assembly under inspection. The assembly is presented to the camera by an industrial robot. Alternatively, it is possible to have the assembly attached to a fixture and the camera being mounted to an industrial robot. In either case, the coordinate system transformation between the camera and the root node of the kinematic tree is known. The image is preprocessed with a SUSAN filter [8], yielding an edge image from which the approximate Euclidean distance transform is calculated [9]. Finally, the system uses the new KPF to find the pose parameters which register the rigid part models best to the edges observed in the image.

## 3 Assembly Model

Within manufacturing work flows, 3D CAD descriptions of processed objects are generally available. Accordingly, the system described in this paper automatically extracts edge features from 3D CAD data. The results are manually grouped to kinematic trees. The approach is based on our framework presented in [10]. A brief overview is provided here for clarity.

The automatic feature extraction determines *contour edges*, i.e. edges which possibly occur as part of the object's silhouette against an arbitrary background. It proceeds as follows: First, the set  $E_c$  of a CAD model's potential contour edges is found by analyzing the angle between all of the model's adjacent triangles:

$$E_c = \{E | \text{isconvex}(E, \mathbf{N}_E^1, \mathbf{N}_E^2) \wedge \alpha_E = \angle(\mathbf{N}_E^1, \mathbf{N}_E^2) > 0\} \quad (1)$$

where  $\mathbf{N}_E^1, \mathbf{N}_E^2$  represent the normals of two adjacent triangles and  $E$  the edge shared between them. The angle  $\alpha_E$  allows to assign a score to each element of  $E_c$ , because more acute angles yield a more frequent appearance of edges under different perspective projections. All elements of  $E_c$  with a certain minimum score are taken as contour edges that together represent an assembly part.

The feature extraction stage precomputes the visibility of all elements in  $E_c$  w.r.t. all possible discrete view-angles upon them and stores the results in a run-length encoded *visibility map*. Furthermore, oriented bounding boxes (OBBs) are fitted automatically to the vertex data contained in the originating CAD models. By indexing into the visibility maps and performing occlusion checks on the OBBs, the system can later infer the visible contour edges of the assembly for a specific pose parameter assignment in a very fast and efficient manner. The resulting online edge contour prediction has a worst time complexity that is linear in the number of visible contour edge elements and OBBs.

## 4 Kernel Particle Filter

This section details the new kernel particle filter for assembly pose recognition. First, it provides a brief description of the particle filter which generates a sample set representation of a posterior probability density. The posterior relates the space of assembly poses to an intensity image of the inspected aggregate. Second, the iterative mean shift procedure that locates posterior modes is described. Third, the variable bandwidth selection inherent to the mean shift mode finding is presented.

### 4.1 Particle Filter

Let the  $d$  degrees of freedom of an assembly pose be represented by a configuration state vector  $\mathbf{x} \in \mathbb{R}^d$ . Furthermore, let an observation of the assembly be denoted as  $\mathbf{y}$ . Assume, too, that a prior  $p(\mathbf{x})$  is given from which state vectors can be sampled. In our case,  $p(\mathbf{x})$  is uniformly distributed over all physically possible assembly poses. The latter are specified manually during the system's offline stage and describe physical limits of the assembly pose parameters. A sample set representation of the posterior probability density  $p(\mathbf{x}|\mathbf{y})$  can then be obtained by particle filtering [11], which in our case degenerates to factored sampling because there is no sequence of observations over time but only one image measurement. Factored sampling draws samples  $\{\mathbf{s}^{(n)}\}_{n=1}^N$  from the prior and assigns each sample a weight  $\mathbf{w}^{(n)} = p(\mathbf{y}|\mathbf{s}^{(n)})$  corresponding to a measurement density. Afterwards, the weights are normalized such that they integrate to one. The set  $\{\mathbf{s}^{(n)}, \mathbf{w}^{(n)}\}_{n=1}^N$  is now a sample set approximation to the posterior.

To evaluate the measurement density  $p(\mathbf{y}|\mathbf{s}^{(n)})$ , the assembly model is first used to determine the contour edges visible under pose parameterization  $\mathbf{s}^{(n)}$ . The edge elements are then projected to the 2D image space by applying the model of the fully calibrated camera that captured the image. Afterwards, a set  $M$  of 2D control points is positioned equidistantly along the projected edge elements. Let  $I$  denote the set of edge pixels extracted from  $\mathbf{y}$ . The *partial directed Hausdorff distance* between  $M$  and  $I$  is then defined as [12]

$$h^f(M, I) = f_{m \in M}^{\text{th}} \min_{i \in I} \|i - m\| \quad (2)$$

where  $f_{z \in Z}^{\text{th}} g(z)$  denotes the  $f$ -th quantile of  $g(z)$  over an ordered set  $Z$ , for some value of  $f$  between zero and one. According to [12],  $h^f(M, I)$  defines a measure for the similarity between  $M$  and  $I$  that is quite robust against outliers. The term  $\min_{i \in I} \|i - m\|$  is pre-calculated once per observation  $\mathbf{y}$ , using a chamfering technique [9]. Once the partial directed Hausdorff distance is computed, the measurement density is expressed as a Gaussian weighting function

$$p(\mathbf{y}|\mathbf{s}^{(n)}) = \exp\left(-\frac{h^f(M, I)^2}{2\sigma^2 c^2}\right) \quad (3)$$

where  $\sigma$  is the standard deviation and  $c$  a constant normalizing  $h^f(M, I)$  to  $[0, 1]$ . Equation 3 converts  $h^f(M, I)$  into a likelihood that can be combined with other cue likelihoods as suggested in [5]. The value of  $\sigma$  was chosen experimentally.

### 4.2 Mean Shift Based Mode Estimation

The system described in this paper recovers the pose of an assembly by locating the most prominent mode of  $p(\mathbf{x}|\mathbf{y})$ . However, sampling the posterior exhaustively in a high-dimensional space would demand excessive numbers of samples. To alleviate this problem, [4], [6] and [5] successfully use mean shift based mode detection. The procedure helps in maintaining a compact representation of the posterior modes and thus significantly reduces the number of samples needed to represent  $p(\mathbf{x}|\mathbf{y})$ .

Mean shift based mode detection is based on kernel density estimation, which is also known as the Parzen window method [13]. Given the set of  $d$ -dimensional samples and weights resulting from Sect. 4.1, the kernel density estimate (KDE) of the posterior with kernel  $K$  and bandwidth parameter  $b$  can be written as

$$\hat{p}(\mathbf{x}|\mathbf{y}) = \frac{1}{Nb^d} \sum_{n=1}^N K\left(\frac{\mathbf{x} - \mathbf{s}^{(n)}}{b}\right) \mathbf{w}^{(n)} . \tag{4}$$

In this paper,  $K$  is the radially symmetric Epanechnikov kernel:

$$K(\mathbf{x}) = \begin{cases} \frac{1}{2}c_d^{-1}(d+2)(1 - \|\mathbf{x}\|) & : \|\mathbf{x}\| \leq 1 \\ 0 & : otherwise \end{cases} \tag{5}$$

where  $c_d$  is the volume of the unit  $d$ -dimensional sphere. The gradient of  $\hat{p}(\mathbf{x}|\mathbf{y})$  can be estimated and the samples  $\{\mathbf{s}^{(n)}\}_{n=1}^N$  shifted towards stationary points in the posterior probability density. This technique is known as the *mean shift procedure* [14]. In short, it proceeds by iteratively shifting each sample to the mean position calculated by

$$\mathbf{m}(\mathbf{s}^{(n)}) = \frac{\sum_{l=1}^N K\left(\frac{\mathbf{s}^{(n)} - \mathbf{s}^{(l)}}{b}\right) \mathbf{w}^{(l)} \mathbf{s}^{(l)}}{\sum_{l=1}^N K\left(\frac{\mathbf{s}^{(n)} - \mathbf{s}^{(l)}}{b}\right) \mathbf{w}^{(l)}} . \tag{6}$$

In (4), the KDE is constructed from a radially symmetric kernel. However, the  $\mathbf{s}^{(n)}$  encode translational and rotational pose parameters from different scales for which a radially symmetric kernel might not be appropriate. Thus, PCA is used as a whitening step to obtain a set  $\{\mathbf{s}_v^{(n)}\}_{n=1}^N$  of variance normalized samples.

In summary, the mean shift based mode estimation proceeds by initially generating a sample set with weights  $\{\mathbf{s}^{(n)}, \mathbf{w}^{(n)}\}_{n=1}^N$ . After variance normalization, the samples are moved to their mean positions calculated from (6). The shifted samples are then reprojected to the original space by applying the inverse whitening matrix, resulting in updated samples  $\mathbf{s}^{(n)'}$ . This in turn allows to update the weights  $\mathbf{w}^{(n)}$  by evaluating the measurement density as described in Sect. 4.1. However, the mean shift procedure was originally designed not to shift the samples but only some mode descriptors. Shifting the samples affects the KDE. To alleviate this effect the weights must be normalized by computing

$$\mathbf{w}^{(n)'} = \frac{p(\mathbf{y}|\mathbf{s}^{(n)'})}{q(\mathbf{s}^{(n)'})} \tag{7}$$

where  $q(\mathbf{s}^{(n)'})$  is the new proposal density determined by

$$q(\mathbf{s}^{(n)'}) = \frac{1}{Nb^d} \sum_{l=1}^N K \left( \frac{\mathbf{s}^{(n)' } - \mathbf{s}^{(l)'}}{b} \right). \tag{8}$$

Once the sample weights have been normalized, the mean shift procedure can be repeated. Within our experiments, five iterations were usually sufficient to locate the posterior modes. The modes can be determined by calculating

$$\hat{\mathbf{x}}(\mathbf{s}^{(n)'}) = \sum_{\mathbf{s}^{(l)' } \in S_h(\mathbf{s}^{(n)'})} \mathbf{w}^{(l)' } \mathbf{s}^{(l)' } \tag{9}$$

with  $S_h(\mathbf{s}^{(n)'})$  denoting a hyper-sphere around a certain sample. The most prominent mode is the  $\hat{\mathbf{x}}(\mathbf{s}^{(n)'})$  with the largest accumulated weight.

### 4.3 Variable Bandwidth Selection

Experiments with real data revealed that the bandwidth parameter  $b$  from Sect. 4.2 is hard to specify. Fixed settings tend to over- or undersmooth the posterior after a few iterations of mean shift. In the context of mean shift image segmentation, [7] proposed a variable bandwidth selection scheme that we adapted to our kernel particle filter. It is based on the concept of a *sample point* density estimator that selects a different bandwidth  $b = b(\mathbf{s}^{(n)})$  for each sample  $\mathbf{s}^{(n)}$ :

$$\hat{p}_{\text{sp}}(\mathbf{x}|\mathbf{y}) = \frac{1}{N} \sum_{n=1}^N \frac{1}{b(\mathbf{s}^{(n)})^d} K \left( \frac{\mathbf{x} - \mathbf{s}^{(n)}}{b(\mathbf{s}^{(n)})} \right) \mathbf{w}^{(n)} \tag{10}$$

where  $\hat{p}_{\text{sp}}(\mathbf{x}|\mathbf{y})$  is the sample point density estimate of the posterior. It can be shown that the following choice of  $b(\mathbf{s}^{(n)})$  significantly reduces the bias of the density estimator [7]:

$$b(\mathbf{s}^{(n)}) = b_0 \left( \frac{\lambda}{f(\mathbf{s}^{(n)})} \right)^{\frac{1}{2}} \tag{11}$$

where  $\lambda$  is a proportionality constant. The true density  $f(\mathbf{s}^{(n)})$  in (11) is unknown. However, a *pilot density*  $\tilde{f}$  can be used that is obtained with the initial bandwidth parameter  $b_0$  and the fixed bandwidth KDE as in (8).

The proportionality constant  $\lambda$  divides the range of density values into low and high densities [7]. For densities that are locally low, i.e. smaller than  $\lambda$ ,  $b(\mathbf{s}^{(n)})$  is increased w.r.t.  $b_0$ . Locally high densities result in a decreased local bandwidth. A good choice of  $\lambda$  is the geometric mean of the pilot density values. Effectively, the adaptive bandwidth selection scheme still depends on a bandwidth parameter. However,  $b_0$  only serves as a reference bandwidth that is changed according to local density fluctuations. The approach enabled us to use the same bandwidth setting for all experiments reported in the next section.

## 5 System Evaluation

In order to evaluate the system's measurement accuracy and precision, experiments with two different assemblies were carried out. Both were constructed from wooden toy building blocks like screws, nuts and bars. Concerning the pose estimation problem, these parts are rather challenging because they have coated surfaces which reflect light strongly and provide little structure.

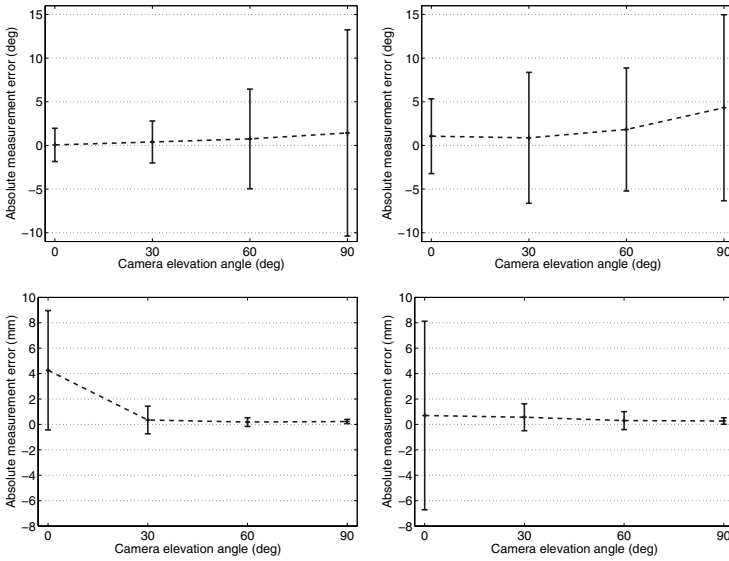
The first assembly was used to study a simple scenario in detail and is illustrated in Fig. 1. It consisted of a single screw with hexagonal head that was screwed into a block. A static camera was calibrated to the block and the assembly captured under four different camera elevation angles ( $0^\circ$ ,  $30^\circ$ ,  $60^\circ$  and  $90^\circ$ ). For each angle, 125 images were captured, measuring five different screw positions. The whole procedure was carried out for a large and a small zoom setting, resulting in image scales of 0.1 and 0.3 mm per pixel and a total of 1000 image measurements taken. Afterwards, the new KPF described in this paper determined the six pose parameters of the screw w.r.t. the block within the captured images (using 500 particles and five iterations of mean shift). Finally, the screw's rotation and translation around the screw axis were compared to the ground truth that had been measured manually with Vernier calipers and a protractor.



**Fig. 1.** A simple screw-block assembly. *Left:* Elevation  $60^\circ$ , image scale 0.1 mm per pixel. *Right:* Elevation  $0^\circ$ , image scale 0.3 mm per pixel.

The results of the first experiment are illustrated in Fig. 2. The graphs in the top row depict the mean absolute pose estimation error and standard deviation for the measurement of the rotation around the screw axis. They show that the screw rotation is measured most accurately and precisely under a small elevation angle around  $0^\circ$ , i.e. when the hexagonal screw head is seen straight from above. Under these ideal conditions, the screw rotation can still be measured with reasonable precision even if the size of the screw within the image is comparatively small. However, it can be seen as well that the measurement precision quickly decays with increasing elevation angle. For elevations much larger than  $30^\circ$ , rotation measurements might be too unprecise even when the screw covers large parts of the image. On the other hand, the bottom row of graphs in Fig. 2

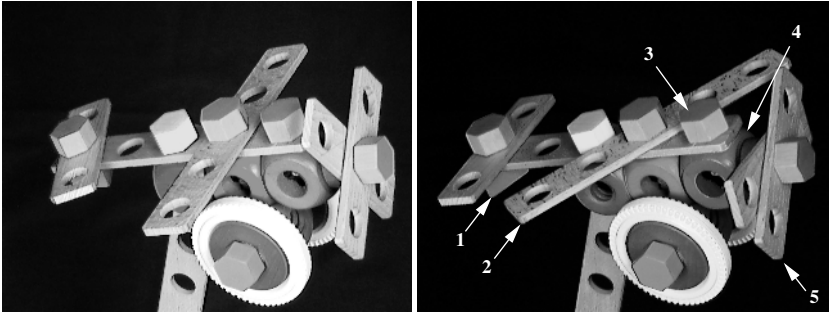
illustrates that the screw translation is measured best under a large camera elevation angle around  $90^\circ$ . It is thus impossible to measure screw translation and rotation most accurately and precisely from the same camera perspective.



**Fig. 2.** Mean absolute pose estimation error and standard deviation. *Top row:* Screw rotation measurements. *Top left:* Image scale was 0.1 mm per pixel. *Top right:* Image scale was 0.3 mm per pixel. *Bottom row:* Screw translation measurements. *Bottom left:* Image scale was 0.1 mm per pixel. *Bottom right:* Image scale was 0.3 mm per pixel.

The second assembly was a complex model airplane built from 20 parts which is illustrated in Fig. 3. To measure the pose parameter ground truth for all 20 parts manually would have been impossible, so we focused on the five most accessible parts that are shown on the right hand side of Fig. 3. As ground truth, we manually measured the most strongly varying rotation parameter of parts 1, 2, 3 and 5 and the most strongly varying translational parameter of parts 1 and 4 w.r.t. their parent within the kinematic tree. A static camera was calibrated to the fixture to which the airplane was mounted. Afterwards, 50 images of the assembly were captured in which the pose parameters of the five parts were changed systematically.

Recovering all pose parameters of the assembly simultaneously would have failed because the sample space would simply have been too large. Instead, the kinematic tree was recursively traversed depth-first and the six pose parameters of each rigid part recovered individually. This approach is potentially less robust but did not lead to any problems here. As for the first assembly, the new KPF was parameterized with 500 particles and five iterations of mean shift. The results are summarized in Tables 1 and 2, together with results from two other recent



**Fig. 3.** An airplane assembled from 20 parts in two different poses. The ground truth was manually measured for the numbered parts.

inspection systems [15], [16]. In [16], hierarchical template matching was used to recover the pose of an oil cap. In a second experiment, an ignition plug was inspected. Unfortunately, the authors only specify mean absolute measurement errors. In [15], a 2D-3D pose estimation procedure for the visual inspection of single rigid bodies is reported together with a detailed analysis of the system's measurement accuracy and precision. Tables 1 and 2 only state the best-case results published there regarding the inspection of an oil cap.

Comparing the results, it can be seen that our system recovers pose parameters with a competitive mean absolute error and standard deviation, at least in the case of translational pose parameters. Concerning the recovery of rotational parameters, the absolute error has on average a slightly higher standard deviation than the system reported in [15].

**Table 1.** Mean absolute error and standard deviation of angle measurements

	part 1	part 2	part 3	part 5	Kölzow [15]	Bank et al. [16]
$\mu$ [°]	-0.4	-0.4	-1.4	-0.3	-0.5	1.0
$\sigma$ [°]	3.1	0.6	3.1	2.1	1.6	-

**Table 2.** Mean absolute error and standard deviation of translation measurements

	part 1	part 4	Kölzow [15]	Bank et al. [16]
$\mu$ [mm]	0.2	0.3	-0.3	0.5
$\sigma$ [mm]	0.5	0.6	0.5	-

## 6 Conclusion and Future Work

We have presented a unique system that recovers the pose parameters of assemblies by using a new kernel particle filter. Regarding the mean absolute error and standard deviation of the measurements, our system performs comparable

to state-of-the-art systems that focus on the single rigid body case. However, the measurement density evaluation within the new KPF is very expensive in terms of computational load. This is why, for the assembly with 20 parts, inspection took about five to ten seconds per part on a Pentium IV 2.0 GHz processor. As a next step, we will therefore implement the edge prediction in OpenGL. This should increase the speed by up to two orders of magnitude.

## References

1. Goddard, J.S.: Pose and Motion Estimation from Vision using Dual Quaternion-Based Extended Kalman Filtering. PhD thesis, Univ. of Tennessee, Knoxville (1997)
2. Rosenhahn, B., Sommer, G.: Pose Estimation in Conformal Geometric Algebra. *Journal of Mathematical Imaging and Vision* **22** (2005) 27–70
3. Besl, P.J., McKay, N.D.: A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **14**(2) (1992) 239–256
4. Chang, C., Ansari, R.: Kernel Particle Filter: Iterative Sampling for Efficient Visual Tracking. In: *ICIP 2003*. (2003) 977–980
5. Schmidt, J., Kwolek, B., Fritsch, J.: Kernel Particle Filter for Real-Time 3D Body Tracking in Monocular Color Images. In: *Proc. of Automatic Face and Gesture Recognition*, Southampton, UK, IEEE (2006) 567–572
6. Chang, C., Ansari, R.: Kernel Particle Filter for Visual Tracking. *IEEE Signal Processing Letters* **12**(3) (2005) 242–245
7. Comaniciu, D., Ramesh, V., Meer, P.: The Variable Bandwidth Mean Shift and Data-Driven Scale Selection. In: *ICCV 2001*. Volume 1., IEEE (2001) 438–445
8. Smith, S.M., Brady, J.M.: SUSAN - a new approach to low level image processing. *International Journal of Computer Vision* **23**(1) (1997) 45–78
9. Borgefors, G.: Distance Transformations in Digital Images. *Computer Vision, Graphics, and Image Processing* **34** (1986) 344–371
10. Stöbel, D., Hanheide, M., Sagerer, G., Krüger, L., Ellenrieder, M.: Feature and Viewpoint Selection for Industrial Car Assembly. In Rasmussen, C.E., Bülthoff, H.H., Giese, M.A., Schölkopf, B., eds.: *Pattern Recognition: 26th DAGM Symposium*. Volume 3175 of *Lecture Notes in Computer Science.*, Springer Berlin / Heidelberg, Germany, Springer-Verlag (2004) 528–535
11. Isard, M., Blake, A.: ICONDENSATION: Unifying low-level and high-level tracking in a stochastic framework. *Lecture Notes in Computer Science* **1406** (1998) 893–908
12. Rucklidge, W.: Efficient Visual Recognition Using the Hausdorff Distance. Volume 1173 of *Lecture Notes in Computer Science*. Springer-Verlag (1996)
13. Duda, R.O., Hart, P.E.: *Pattern Classification and Scene Analysis*. New York: Wiley & Sons (1973)
14. Comaniciu, D., Meer, P.: Mean Shift: A Robust Approach Toward Feature Space Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**(5) (2002) 603–619
15. Kölzow, T.: System zur Klassifikation und Lokalisation von 3D-Objekten durch Anpassung vereinheitlichter Merkmale in Bildfolgen. PhD thesis, Bielefeld University (2002) in german.
16. von Bank, C., Gavrilu, D.M., Wöhler, C.: A Visual Quality Inspection System Based on a Hierarchical 3D Pose Estimation Algorithm. In Michaelis, B., Krell, G., eds.: *Pattern Recognition: 25th DAGM Symposium*. Volume 2781 of *Lecture Notes in Computer Science.*, Springer-Verlag (2003) 179–186



# Monocular 3D Scene Reconstruction at Absolute Scales by Combination of Geometric and Real-Aperture Methods

Annika Kuhl<sup>1,2</sup>, Christian Wöhler<sup>1</sup>, Lars Krüger<sup>1</sup>,  
Pablo d'Angelo<sup>1</sup>, and Horst-Michael Groß<sup>2</sup>

<sup>1</sup> DaimlerChrysler AG, Group Research, Machine Perception  
P. O. Box 2360, D-89013 Ulm, Germany

<sup>2</sup> Technical University of Ilmenau, Faculty of Computer Science and Automation  
P. O. Box 100565, D-98684 Ilmenau, Germany

**Abstract.** We propose a method for combining geometric and real-aperture methods for monocular 3D reconstruction of static scenes at absolute scales. Our algorithm relies on a sequence of images of the object acquired by a monocular camera of fixed focal setting from different viewpoints. Object features are tracked over a range of distances from the camera with a small depth of field, leading to a varying degree of defocus for each feature. Information on absolute depth is obtained based on a Depth-from-Defocus approach. The parameters of the point spread functions estimated by Depth-from-Defocus are used as a regularisation term for Structure-from-Motion. The reprojection error obtained from Bundle Adjustment and the absolute depth error obtained from Depth-from-Defocus are simultaneously minimised for all tracked object features. The proposed method yields absolutely scaled 3D coordinates of the scene points without any prior knowledge about the structure of the scene. Evaluating the algorithm on real-world data we demonstrate that it yields typical relative errors between 2 and 3 percent. Possible applications of our approach are self-localisation and mapping for mobile robotic systems and pose estimation in industrial machine vision.

## 1 Introduction

The knowledge of three-dimensional structure plays an important role in many fields of research such as navigation, obstacle avoidance, and object detection. Depth-from-Stereo [1] was one of the first methods for recovering depth information as it is inspired by human vision. Hereby the known geometry of the cameras is used to triangulate the spatial position of corresponding points from two images that are acquired from different viewpoints. The disadvantage of stereo vision is its need for a pair of precisely calibrated cameras, making it complex and costly for many applications. Therefore spatial scene reconstruction using monocular camera systems is often a preferable solution. Structure-from-Motion is such an alternative: From corresponding points in at least two images acquired sequentially at different camera positions the spatial positions of the

points are recovered. The problem is that the scene can be reconstructed only up to a scaling factor as long as the camera positions are unknown.

Methods to establish point correspondences from different images require the detection and assignment of salient object features. In [2] image features are proposed that serve well for tracking algorithms. Widely used methods are SIFT features [3], involving the extraction of scale invariant features using a staged filtering approach, or the Kanade-Lucas-Tomasi (KLT) feature detector described in [4] which is based on the Harris corner detector and takes into account affine motion.

A different approach to scene reconstruction utilises position variant appearance, e.g. Shape-from-Shading [5], Depth-from-Defocus [6], and Depth-from-Focus [7]. Depth-from-Defocus methods rely on the fact that a real lens blurs the observed scene before the imaging device records it. The amount of blurring depends on the actual lens, but also on the distance of the observed object to the lens. In [8] this property is used to estimate depth simultaneously for all scene points from only one or two images. Depth information is extracted out of a single image showing sharp discontinuities (edges) [9]. A survey of existing methods is given in [6]. In [10] a method is proposed that computes Depth-from-Defocus in real-time using structured lighting. Depth-from-Focus uses images taken by a single camera at different focus settings to compute depth. The focus settings for the image depicting a point with minimal blurring are used to compute the absolute depth [11]. Further work in this field includes Shape-from-Focus [12] and Inverse Optics [13].

So far, no attempt has been made to combine the precise relative scene reconstruction of Structure-from-Motion with the absolute depth data of Depth-from-Defocus. A work related to this paper was published in [14], where a method to recover affine motion and defocus simultaneously is proposed. However, the spatial extent of the scene is not reconstructed in [14], since planar objects are a requirement for the described method.

The main contribution of this paper consists of a novel combination of Structure-from-Motion (a geometric method) with Depth-from-Defocus (a real-aperture method). We will show that the combination of these methods yields a 3D scene reconstruction at absolute scales based on an image sequence acquired with a monocular camera.

## 2 Structure-from-Motion and Depth-from-Defocus

Structure-from-Motion recovers the spatial scene structure using a monocular camera. A pre-requisite for Structure-from-Motion is the geometric calibration of the camera in terms of estimating the internal parameters such as focal length, distortion parameters, etc. [15]. Subsequently, salient feature points are extracted and tracked across the sequence. The motion of these features relative to the camera is then used to minimise the Bundle Adjustment [16] error term

$$E_{\text{SfM}}(\{T_j\}, \{X_i\}) = \sum_{i=1}^N \sum_{j=1}^M [\mathcal{P}(T_j X_i) - x_{ij}]^2 \quad (1)$$

with respect to the  $M$  camera transforms  $T_j$  and the  $N$  scene points  $X_i$ . Here,  $x_{ij}$  denotes the 2D pixel coordinates of feature  $i$  in image  $j$ . The function  $\mathcal{P}$  denotes the projection of 3D scene points to image coordinates and  $T_j$  the transform of the camera coordinate system of image  $j$  with respect to an arbitrary world coordinate system. To facilitate the integration of defocus information into the Structure-from-Motion framework, the image sequences are acquired such that the object is blurred in the first image of the sequence, becoming increasingly focused in the middle and blurred again in the last images. The focal settings of the camera are adjusted according to the maximal and minimal distance of the object. It may be necessary to fully open the aperture in order to obtain a small depth of field.

Depth-from-Defocus directly recovers the spatial scene structure using a monocular camera. The depth  $D$  of the tracked feature points is calculated by measuring the amount of defocus, expressed e.g. by the standard deviation  $\sigma$  of the Gaussian-shaped point spread function (PSF) that blurs the image. An exact description of the PSF due to diffraction of light at a circular aperture is given by the radially symmetric Airy pattern  $A(r) \propto [J_1(r)/r]^2$ , where  $J_1(r)$  is a Bessel function of the first kind [17]. For practical purposes, however, when a variety of additional lens-specific influencing quantities (e.g. chromatic aberration) is involved, the Gaussian function is a reasonable approximation to the PSF [6]. In the following,  $\sigma$  will be referred to as the “radius” of the PSF.

Measuring  $\sigma$  is the most important part of the depth estimation. The classical Depth-from-Defocus approach uses two images of the same object taken at two different focal settings [6]. In [9] it is shown that a-priori information about the image intensity distribution, e.g. the presence of sharp discontinuities (edges), allows the computation of the PSF radius  $\sigma$  based on a single image. This is achieved by estimating the value of  $\sigma$  that generates the observed intensity distribution from the known ideal intensity distribution. Since in our scenario no such a-priori information is available, we suggest the empirical determination of the so-called Depth-Defocus-Function, expressing the standard deviation  $\sigma$  of the Gaussian PSF as a function of depth  $D$ , based on a calibration procedure.

### 3 Spatial Scene Reconstruction by Combining Structure-from-Motion and Depth-from-Defocus

#### 3.1 The Depth-Defocus-Function and Its Calibration

The Depth-Defocus-Function  $\mathcal{S}(D) = \sigma$  expresses the radius  $\sigma$  of the Gaussian PSF as a function of depth  $D$ , i.e. the distance between the object and the lens plane. It is based upon the lens law  $v^{-1} + D^{-1} = f^{-1}$  [17]. An object at distance  $D$  is focused if the distance between lens and image plane is  $v$ , with  $f$  denoting the focal length of the lens. Varying the image plane distance  $v$  by a small amount  $\Delta v$  causes the object to be defocused as the light rays intersect before or behind the image plane. In the geometric optics approximation, a point in the scene is transformed into a so-called circle of confusion of diameter  $|\Delta v|/\kappa$

in the image plane, where  $\kappa$  is the f-stop number expressing the focal length in terms of the aperture diameter. Empirically, we found that for small  $|\Delta v|$  the resulting amount  $F$  of defocus can be modelled by a zero-mean Gaussian, which is symmetric in  $\Delta v$ :

$$F(\Delta v) = \frac{1}{\phi_1} e^{-\frac{1}{\phi_2} \Delta v^2} + \phi_3 . \quad (2)$$

Here, the amount of defocus is described in terms of the radius  $\sigma$  of the Gaussian PSF. But since the Depth-Defocus-Function expresses the relation between the depth of an object and its defocus, the image plane is assumed to be fixed while the distance  $D$  of the object varies by the amount  $\Delta D$ , such that  $\Delta D = 0$  refers to an object that is well focused. But since neither  $D$  nor  $\Delta D$  are known, the functional relation needs to be modelled with respect to  $\Delta v$ :

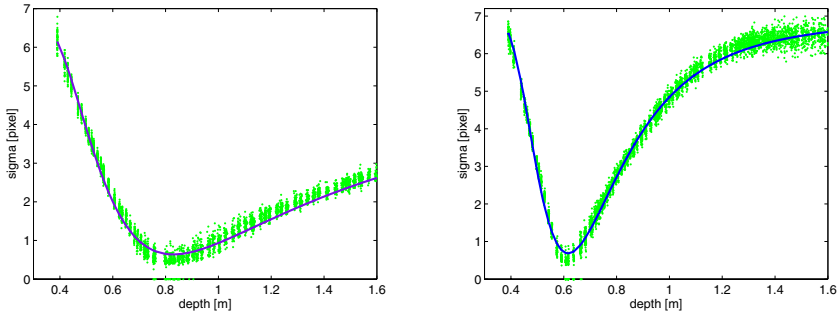
$$\frac{1}{v + \Delta v} + \frac{1}{D} = \frac{1}{f} . \quad (3)$$

A value of  $\Delta v \neq 0$  refers to a defocused object point. Solving Eq. (3) for  $\Delta v$  and inserting  $\Delta v$  in Eq. (2) yields the Depth-Defocus-Function

$$\mathcal{S}(D) = \frac{1}{\phi_1} e^{-\frac{1}{\phi_2} \left(\frac{fD}{D-f} - v\right)^2} + \phi_3 . \quad (4)$$

Calibrating the Depth-Defocus-Function  $\mathcal{S}(D)$  for a given lens corresponds to determining the parameters  $\phi_1$ ,  $\phi_2$ , and  $\phi_3$  in Eq. (4). This is achieved by obtaining a large set of measured  $(\sigma, D)$  data points and perform a least mean squares fit to Eq. (4), where  $D$  is the distance from the camera and  $\sigma$  the radius of the Gaussian PSF  $G$  used to blur the well focused image according to  $I_{ij} = G(\sigma) * I_{if_i}$ . Here,  $I_{if_i}$  represents a small region of interest (ROI) around feature  $i$  in the image  $f_i$  in which this feature is best focused, and  $I_{ij}$  a ROI of equal size around feature  $i$  in image  $j$ .

For calibration, an image sequence is acquired while the camera approaches at uniform speed a calibration rig displaying a checker board. The sharp black-and-white corners of the checker board are robustly and precisely detectable [15] even in defocused images. Small ROIs around each corner allow the estimation of defocus using their greyvalue variance  $\chi$ . The better focused the corner, the higher is the variance  $\chi$ . We found experimentally that the parameterised defocus model according to Eq. (4) is also a reasonable description of the dependence of  $\chi$  on the depth  $D$ . For our calibration sequence the camera motion is uniform and the image index  $j$  is strongly correlated with the object distance  $D$ . Hence, Eq. (4) is fitted to the measured  $(\chi, j)$  data points for each corner  $i$ , such that the location of the maximum of  $\mathcal{S}$  yields the index  $f_i$  of the image in which the ROI around corner  $i$  is best focused. This ROI corresponds to  $I_{if_i}$ . The fitting procedure is applied to introduce robustness with respect to pixel noise. For non-uniform camera motion the index  $f_i$  can be obtained by a parabolic fit to the values of  $\chi$  around the maximum or by directly selecting the ROI with maximal  $\chi$ . The depth  $D$  of each corner is reconstructed from the pose of the complete rig according to [18].



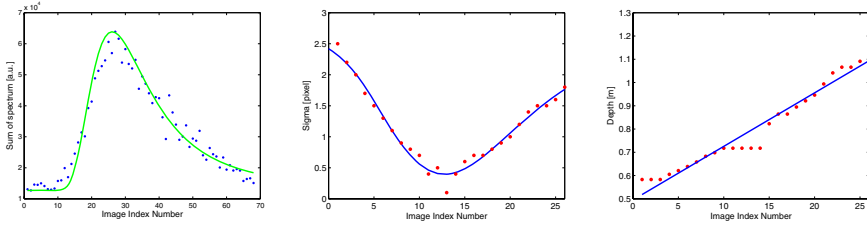
**Fig. 1.** Depth-Defocus-Functions of two lenses with  $f = 12$  mm (left) and  $f = 20$  mm (right), fitted to the measured data points according to Eq. (4), respectively

For each tracked corner  $i$ , we compute for each ROI  $I_{ij}$  the amount of defocus, i.e. the  $\sigma$  value relative to the previously determined best focused ROI  $I_{if_i}$ . By employing the bisection method, we determine the value of  $\sigma$  for which the root mean square deviation between  $G(\sigma) * I_{if_i}$  and  $I_{ij}$  becomes minimal. The Depth-Defocus-Function is then obtained by a least mean squares fit to all determined  $(\sigma, D)$  data points. Two examples are shown in Fig. 1 for lenses with focal lengths of 12 mm and 20 mm and f-stop numbers of 1.4 and 2.4, respectively. Objects at a distance of about 0.8 m and 0.6 m, respectively, are in focus, corresponding to the minimum of the curve.

### 3.2 Combining Motion, Structure, and Defocus

The Structure-from-Motion analysis involves the extraction of salient features from the image sequence which are tracked using the KLT technique [4]. To facilitate the integration of defocus information, a ROI of constant size is extracted around each feature point at each time step. For each tracked feature, the best focused image has to be identified in order to obtain the increase of defocus for the other images. We found that the greyvalue variance as a measure for defocus does not perform well on features other than black-and-white corners. Instead we make use of the amplitude spectrum  $|\mathcal{F}_I(\omega)|$  of the ROI extracted around the feature position. High-frequency components of the amplitude spectrum denote sharp details, whereas low-frequency components refer to large-scale features. Hence, the integral over the high-frequency components serves as a measure for the sharpness of a certain tracked feature. However, since the highest-frequency components are considerably affected by pixel noise and defocus has no perceivable effect on the low-frequency components, a frequency band between  $\omega_0$  and  $\omega_1$  is taken into account according to  $H = \int_{\omega_0}^{\omega_1} |\mathcal{F}_I(\omega)| d\omega$

with  $\omega_0 = \frac{1}{4}\omega_{\max}$  and  $\omega_1 = \frac{3}{4}\omega_{\max}$ , where  $\omega_{\max}$  is the maximum frequency. The amount of defocus increases with decreasing value of  $H$ . The defocus measure  $H$



**Fig. 2.** From the left: Image index vs. defocus measure  $H$  for a tracked image feature; image index vs. PSF radius  $\sigma$ ; image index vs. inferred depth  $D$

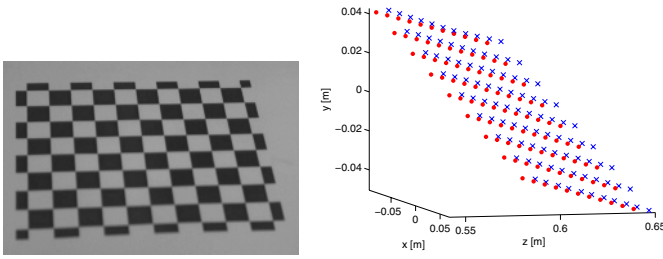
is used to determine the index of the best focused ROI for each tracked feature in the same manner as the greyvalue variance  $\chi$  in Section 3.1. The value of  $H$  cannot be used for comparing the amount of defocus among different feature points since the maximum value of  $H$  depends on the image content. The same is true for the greyvalue variance. Hence, both the integral  $H$  of the amplitude spectrum as well as the greyvalue variance are merely used for determining the index of the image in which a certain feature is best focused.

The defocus, i.e. the radius  $\sigma$  of the Gaussian PSF, is then computed relative to the best focused ROI according to Section 3.1. The depth  $D$  is obtained by inverting the Depth-Defocus-Function  $\mathcal{S}(D)$  according to Eq. (4). The encountered two-fold ambiguity is resolved by using information about the direction of camera motion, which is obtained either based on a-priori knowledge or by performing a Structure-from-Motion analysis according to Eq. (1), yielding information about the path of the camera. If the estimated value of  $\sigma$  is smaller than the minimum of  $\mathcal{S}(D)$ , the depth is set to the value at which  $\mathcal{S}(D)$  is minimal. For an example feature, the calculated defocus and the inferred depth values are shown in Fig. 2.

A general property of the KLT algorithm is that the accuracy of the feature tracker decreases with increasing defocus of the reference pattern. Hence, the feature positions are refined by repeating the tracking procedure for all features, starting from the “sharpest” image located near the middle of the sequence which displays the largest value of  $H$  averaged over all features, proceeding towards either end of the sequence and using the ROIs extracted from this image as reference patterns. The 3D coordinates  $X_i$  of the scene points are then computed by searching for the minimum of the combined error term

$$E_{\text{comb}}(\{T_j\}, \{X_i\}) = \sum_{i=1}^N \sum_{j=1}^M \left[ (\mathcal{P}(T_j X_i) - x_{ij})^2 + \alpha (\mathcal{S}([T_j X_i]_z) - \sigma_{ij})^2 \right] \quad (5)$$

with respect to the  $M$  camera transforms  $T_j$  and the  $N$  scene points  $X_i$ . The value of  $\sigma_{ij}$  corresponds to the estimated PSF radius for feature  $i$  in image  $j$ ,  $\alpha$  is a weighting factor,  $\mathcal{S}$  the Depth-Defocus-Function that calculates the expected defocus of feature  $i$  in image  $j$ , and  $[\cdot]_z$  the  $z$  coordinate, i.e. the depth  $D$ , of a scene point. The correspondingly estimated radii  $\sigma_{ij}$  of the Gaussian PSFs define



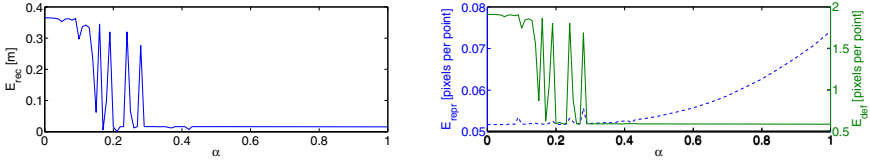
**Fig. 3.** True (dots) and reconstructed (crosses) 3D pose of the checker board ( $\alpha = 0.42$ )

a regularisation term in Eq. (5), such that absolutely scaled 3D coordinates  $X_i$  of the scene points are obtained. The values of  $X_i$  are initialised according to the depth values estimated based on the Depth-from-Defocus approach. To increase the accuracy of the reconstructed 3D scene points, we only make use of feature positions extracted from images in which the feature is not strongly blurred. To minimise the error term  $E_{\text{comb}}$  the Levenberg-Marquardt algorithm [19] is employed.

## 4 Experimental Evaluation

In order to validate our approach we first reconstructed a planar object with known ground truth, using a Baumer  $1032 \times 776$  pixels CCD camera. A checker board as shown in Fig. 3 with  $10 \times 8$  squares of size  $15 \times 15 \text{ mm}^2$ , respectively, was used. The 99 corners serve as features and are extracted in every image using the method described in [15] to assure sub-pixel accuracy. The true pose of the checker board is obtained according to [18] based on the given size of the squares. Note that in [18] the true pose of the checker board is determined by applying a least mean squares fit on a single image, whereas the proposed algorithm estimates the 3D structure of a scene by means of a least mean squares fit applied to the whole image sequence. Comparing the obtained results with the determined true pose of the object is actually a comparison between two methods conducting different least mean squares fits.

The deviation  $E_{\text{rec}}$  of the reconstructed 3D scene point coordinates  $X_i$  from the ground truth values  $X_i^{\text{true}}$  is given by  $E_{\text{rec}} = \left[ \frac{1}{N} \sum_{i=1}^N (X_i - X_i^{\text{true}})^2 \right]^{1/2}$ . To determine an appropriate weight parameter  $\alpha$  we computed  $E_{\text{rec}}$  for different  $\alpha$  values in the range between 0 and 1. For  $\alpha = 0$  the global minimisation is equivalent to Structure-from-Motion initialised with the calculated Depth-from-Defocus values. One must keep in mind, however, that the absolute scaling factor is then part of the gauge freedom of the Bundle Adjustment method, resulting in a corresponding “flatness” of the error function. Small  $\alpha$  values lead to an instable convergence. The value of  $E_{\text{rec}}$  levels off to 16 mm for  $\alpha \approx 0.3$  and obtains its minimum value of 7 mm for  $\alpha = 0.42$ . The root mean square deviation of the reconstructed size of the squares from the true value of 15 mm



**Fig. 4.** Dependence of  $E_{\text{rec}}$  (left diagram),  $E_{\text{repr}}$  (right diagram, dashed curve, left axis), and  $E_{\text{def}}$  (right diagram, solid curve, right axis) on the weight parameter  $\alpha$

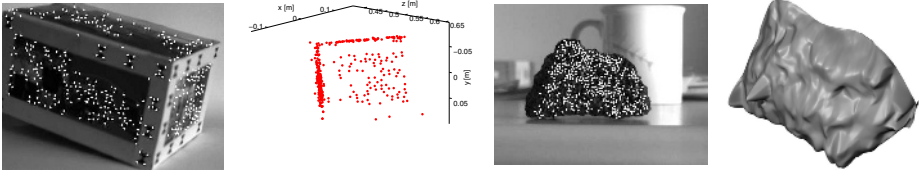
then amounts to 0.2 mm or 1.3%. The most accurate scene reconstruction results are obtained with  $\alpha$  between 0.3 and 0.5. The reconstructed 3D scene points  $X_i$  for  $\alpha = 0.42$  are illustrated in Fig. 3, the dependence of  $E_{\text{rec}}$  on  $\alpha$  in Fig. 4 (left).

In addition to the reconstruction error  $E_{\text{rec}}$ , a further important error measure is the reprojection error  $E_{\text{repr}} = \left[ \frac{1}{MN} \sum_{i=1}^N \sum_{j=1}^M (\mathcal{P}(T_j X_i) - x_{ij})^2 \right]^{1/2}$  denoting the root-mean-square deviation between the measured 2D feature positions  $x_{ij}$  and the reconstructed 3D scene points  $X_i$  reprojected into the images using the reconstructed camera transforms  $T_j$ . The defocus error denotes the root-mean-square deviation between measured and expected radii  $\sigma_{ij}$  of the Gaussian PSFs according to  $E_{\text{def}} = \left[ \frac{1}{NM} \sum_{i=1}^N \sum_{j=1}^M (\mathcal{S}([T_j X_i]_z) - \sigma_{ij})^2 \right]^{1/2}$ . Fig. 4 (right) shows the relation between the weight parameter  $\alpha$ , the reprojection error  $E_{\text{repr}}$ , and the defocus error  $E_{\text{def}}$ . For  $\alpha > 0.3$  the defocus error stabilises to 0.58 pixels per feature. Larger  $\alpha$  values lead to a stronger influence of the Depth-from-Defocus values on the optimisation result, leading to an increasing reprojection error  $E_{\text{repr}}$  due to the inaccuracy of the estimated  $\sigma_{ij}$  values. Although the depth values derived by Depth-from-Defocus are noisy, they are sufficient to establish a reasonably accurate absolute scale. Hence, this first evaluation shows that the combined approach is able to reconstruct scenes at absolute scales without prior knowledge. For constant f-stop number, pixel size, and relative accuracy of the inferred depth  $D$ , it can be shown that the required focal length and aperture of the lens are largely proportional to  $\sqrt{D}$  (proof omitted here). Hence, our approach is restricted to the close-range domain ( $D \sim 1$  m) as long as standard video cameras and lenses are used.

In order to demonstrate the performance of our approach on a non-planar test object of known dimensions we applied our method to the cuboid-shaped object shown in Fig. 5. This object displays a sufficient amount of texture to generate “good features to track” [4]. In addition, black markers on white background with known mutual distances are placed near the edges of the cuboid. As described in Section 3.2, feature points are extracted and tracked using the KLT algorithm, and the 3D coordinates of the scene points are obtained by minimising the error term  $E_{\text{comb}}$  according to Eq. (5).

The reprojection error  $E_{\text{repr}}$  for  $\alpha = 0.5$  amounts to 4.99 pixels. After removing tracking outliers (detected by their associated very large reprojection errors of more than  $3E_{\text{repr}}$ ) the value of  $E_{\text{repr}}$  drops to 1.08 pixels while  $E_{\text{def}}$  amounts





**Fig. 5.** 3D reconstruction of a cuboid and a lava stone ( $\alpha = 0.5$ )

to 0.24 pixels. In order to verify the absolute scale, we compared for  $\alpha = 0.5$  the reconstructed pairwise distances between the black markers on the object (as seen e.g. in the top right corner of the front side) to the corresponding true distances. For this comparison we utilised a set of three pairs of markers with an average true distance of 23.3 mm. The corresponding reconstructed average distance amounts to 23.9 mm, which is 2.6% larger than the ground truth value.

As a real-world object, we examined the lava stone shown in Fig. 5. The resulting reprojection error  $E_{\text{repr}}$  amounts to 2.77 pixels. After outlier rejection,  $E_{\text{repr}}$  decreases to 0.96 pixels while  $E_{\text{def}}$  amounts to 0.19 pixels. The reconstructed shape of the lava stone was again obtained with  $\alpha = 0.5$ . The reconstruction is approximately 2.3% larger than the real object.

In all examples, the fact that the reconstructed absolute scale of the scene appears to be systematically somewhat too large is likely due to a slight deadadjustment of the camera lens after calibration, which may readily occur for standard video lenses as a consequence e.g. of vibrations or variable ambient temperature.

## 5 Summary and Conclusion

We have described a method for combining geometric and real-aperture methods for monocular 3D reconstruction of static scenes at absolute scales. The proposed algorithm is based on a sequence of images of the object acquired by a monocular camera of fixed focal setting from different viewpoints. Feature points are tracked over a range of distances from the camera, resulting in a varying degree of defocus for each tracked feature point. After determining the best focused image of the sequence, we obtain information about absolute depth by a Depth-from-Defocus approach. The inferred PSF radii for the corresponding scene points are utilised to compute a regularisation term for an extended Bundle Adjustment algorithm that simultaneously optimises the reprojection error and the absolute depth error for all feature points tracked across the image sequence. The proposed method yields absolutely scaled 3D coordinates of the object feature points without any prior knowledge about the scene structure. We have demonstrated experimentally that the proposed algorithm yields absolutely scaled 3D coordinates of the feature points with typical relative errors between 2 and 3 percent. Possible application scenarios of our approach are in the domains of self-localisation and mapping for mobile robotic systems as well as pose estimation in the context of industrial machine vision tasks.

## References

1. Scharstein, D., Szeliski, R., Zabih, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. In: IEEE Workshop on Stereo and Multi-Baseline Vision. (2001)
2. Harris, C., Stephens, M.: A combined corner and edge detector. In: Proc. of The Fourth Alvey Vision Conference, Manchester. (1988) 147–151
3. Lowe, D.G.: Object recognition from local scale-invariant features. In: Proc. of the International Conference on Computer Vision ICCV, Corfu. (1999) 1150–1157
4. Shi, J., Tomasi, C.: Good features to track. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94), Seattle (1994)
5. Horn, B.K.: Robot Vision. McGraw-Hill Higher Education (1986)
6. Chaudhuri, S., A.R.: Depth from Defocus: A Real Aperture Imaging Approach. Springer Verlag, Berlin (1999)
7. Ens, J., Lawrence, P.: An investigation of methods for determining depth from focus. IEEE Trans. Pattern Anal. Mach. Intell. **15** (1993) 97–108
8. Pentland, A.P.: Depth of scene from depth of field. In: Proc. Image Understanding Workshop. (1982) 253–259
9. Pentland, A.P.: A new sense for depth of field. IEEE Trans. Pattern Anal. Mach. Intell. **9** (1987) 523–531
10. Watanabe, M., Nayar, S., Noguchi, M.: Real-time computation of depth from defocus. In: Proc. of SPIE: Three-Dimensional and Unconventional Imaging for Industrial Inspection and Metrology. (1995)
11. Grossmann, P.: Depth from focus. Pattern Recogn. Lett. **5** (1987) 63–69
12. Subbarao, M., Choi, T.: Accurate recovery of three-dimensional shape from image focus. IEEE Trans. Pattern Anal. Mach. Intell. **17** (1995) 266–274
13. Subbarao, M.: Efficient depth recovery through inverse optics. In Freeman, H., ed.: Machine Vision for Inspection and Measurement. Academic Press, New York (1989) 101–126
14. Myles, Z., da Vitoria Lobo, N.: Recovering affine motion and defocus blur simultaneously. IEEE Trans. Pattern Anal. Mach. Intell. **20** (1998) 652–658
15. Krüger, L., Wöhler, C., Würz-Wessel, A., Stein, F.: In-factory calibration of multiocular camera systems. In: Photonics Europe, Automatic Target Recognition XIV., Proceedings of the SPIE. Volume 5457. (2004) 126–137
16. Triggs, B., McLauchlan, P., Hartley, R., Fitzgibbon, A.: Bundle adjustment – A modern synthesis. In Triggs, W., Zisserman, A., Szeliski, R., eds.: Vision Algorithms: Theory and Practice. LNCS. Springer Verlag, Berlin (2000) 298–375
17. Pedrotti, F.L.: Introduction to Optics, 2nd Edition. Prentice Hall (1993)
18. Bouguet, J.: Camera calibration toolbox for MATLAB. [www.vision.caltech.edu/bouguetj/calib\\_doc](http://www.vision.caltech.edu/bouguetj/calib_doc) (1997)
19. Madsen, K., Nielsen, H.B., Tingleff, O.: Methods for non-linear least squares problems. <http://www2.imm.dtu.dk/pubdb/p.php?660> (1999)

# An Effective Stereo Matching Algorithm with Optimal Path Cost Aggregation

Mikhail Mozerov

Computer Vision Center and Departament d'Informàtica  
Universitat Autònoma de Barcelona (UAB), 08193 Cerdanyola, Spain  
mozerov@cvc.uab.es

**Abstract.** This paper presents a stereo matching algorithm for obtaining dense disparity maps. Our main contribution is to introduce a new cost aggregation technique of a 3D disparity-space image data, referred to as the Optimal Path Cost Aggregation. The approach is based on the dynamic programming principle, which exactly solves one dimensional optimization problem. Furthermore, the 2D extension of the proposed technique proves an excellent approximation to the global 2D optimization problem. The effectiveness of our approach is demonstrated with several widely used synthetic and real image pairs, including ones with ground-truth value.

## 1 Introduction

Stereo matching is used in many applications, therefore computational stereo has traditionally been, and continues to be one of the most actively researched topics in computer vision. In a survey [9] Scharstein and Szeliski categorize and compare a wide array of algorithms, from window-level correlation to dynamic programming. More recent survey and evaluation by Brown et al. [2] focus on correspondence methods, methods for occlusion, and real-time implementations. It is commonly accepted that the local methods [1, 11, 13] are contrasted with the global methods [4, 5, 8, 12]. Global correspondence methods exploit nonlocal constraints and the use of these constraints makes the computational complexity of global matching significantly greater than that of local matching. Many approaches aim to obtain accurate dense disparity map by using 2D global optimization. Unfortunately, the exact solution of 2D global optimization problem can be found only with exponentially complex algorithms. Dynamic Programming (DP) techniques [3, 4, 7] reformulate the 2D global task to the set of one dimensional optimization problems. Global approaches like Graph Cuts [5] and Belief Propagation [10] resign itself to approximate solutions that can be found by polynomially complex algorithms, but still both approaches is rather slow.

To solve the 2D optimization problem avoiding the exponential complexity of the global solution we introduce a new cost aggregation algorithm, referred to as the Optimal Path Cost Aggregation (OPCA). The proposed algorithm uses the same principle as DP and has the same computational complexity. However, the presented technique accumulates the 2D global cost and, thus, finds an approximate solution of

2D global optimization problem. In the framework of the proposed approach stereo matching is considered as a cost aggregation process of an input 3D disparity-space image (DSI), like in the work [13] for example. Cost aggregation techniques are usually associated with local methods. However, we aim to solve global optimization problem using a cost aggregation algorithm. That is, to find for each disparity  $d$  and pixel  $p$  of a stereo image, the cost (the energy) of the best global solution, which include this particular pair  $(d, p)$ . Eventually, the final optimal solution (or the optimal disparity map) can be easily extracted by analyzing the 3D optimal cost data.

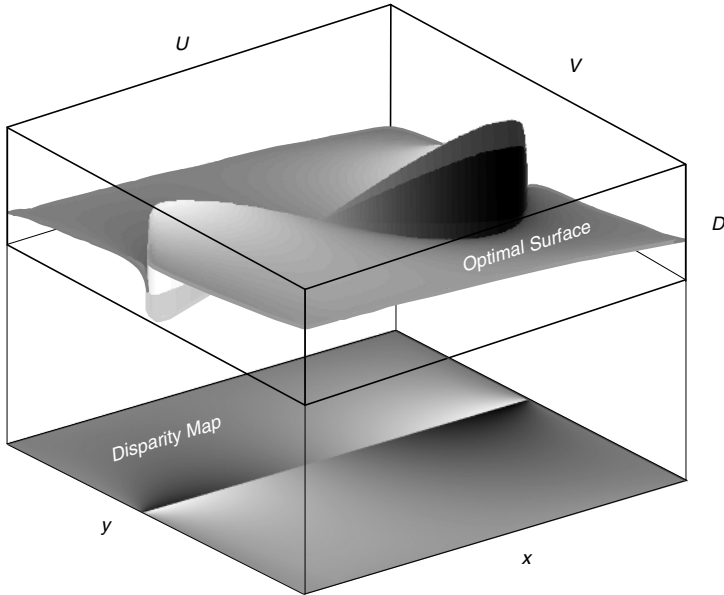


Fig. 1. Geometric interpretation of DSI

## 2 Disparity Space Image Processing Approach

The DSI representation is very popular in stereo matching [2,9] due to the clear geometric interpretation of this model. Indeed, the sought disparity map should coincide with one of all the possible surfaces in the DSI as it is shown in Fig. 1. Furthermore, the global optimization approach assumes that an integral of the initial cost values over such a surface should satisfy a chosen optimality criterion.

The 3D approach assumes that DSI has dimensions  $0 \leq u \leq U_{max}$ ,  $0 \leq v \leq V_{max}$ , and  $0 \leq d \leq D_{max}$ . The stereo images suppose to be rectified, each element  $(u, v, d)$  of the DSI projects to the pixel  $(u, v)$  in the left image and to the pixel  $(u-d, v)$  in the right image. To be strict, we assume that the row size of the right image bigger than the left one:  $-D_{max} \leq u, \leq U_{max}$  instead of  $0 \leq u, \leq U_{max}$ . The row index  $u=0$  is associated with pixels column  $x = D_{max}$  in the both stereo images. Let  $E_n(u, v, d)$

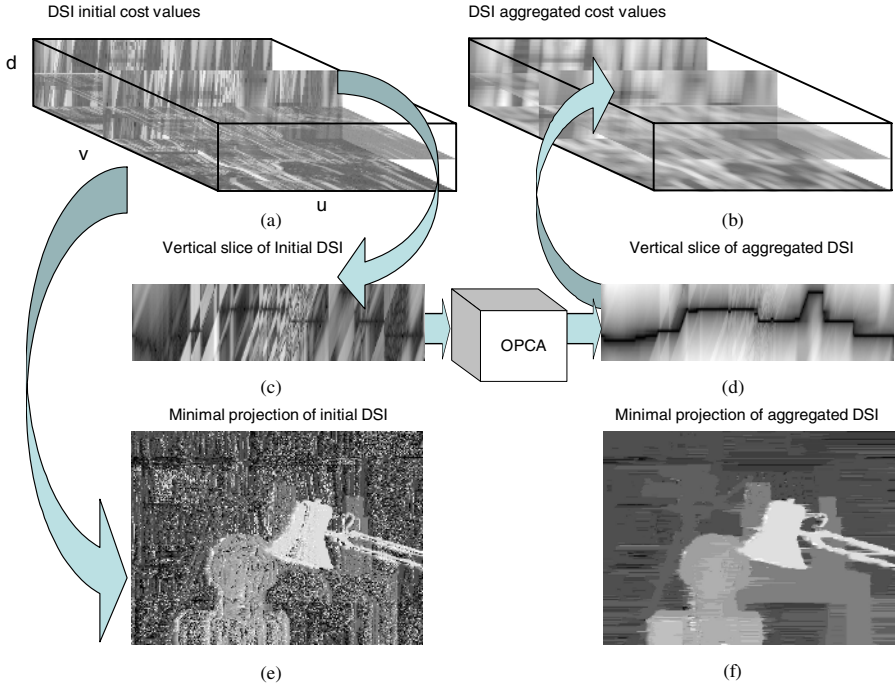


Fig. 2. Scheme of a DSI cost aggregation process

denote the DSI cost value assigned to element  $(u, v, d)$  at the DSI cost aggregation step  $n$ . Initial cost values (or pixelwise distance)  $E_0(u, v, d)$  are calculated using one of the pixel-to-pixel matching metrics that interested readers can find in a survey paper by Brown et al. [3]. In our work absolute difference is considered:

$$E_0(u, v, d) = |I_1(u, v) - I_2(u - d, v)|, \tag{1}$$

where  $I_1(u, v)$  and  $I_2(u, v)$  are the left and the right stereo image respectively. The initial DSI of the Tsukuba stereo pair obtained with equation (1) is illustrated in Fig. 2 (a).

We introduce the minimal projection of the DSI onto the  $(U, V)$  plane that is used as an approximation to the sought disparity map. So, the minimal projection is defined by

$$P_n(u, v) = \arg \min_d E_n(u, v, d). \tag{2}$$

The minimal projections of the initial DSI and the processed one are illustrated in Fig. 2 (e) and (f).

To compare the various solutions  $P(u, v)$  of the processed DSI we introduce the conditional distance  $\mathbb{R}$  between two stereo images as an evaluation criterion. The  $\mathbb{R}$  is defined as follow

$$\mathbb{R}(I_1, I_2 | P(u, v)) = \sum_{u,v} E_0(u, v, P(u, v)) + \lambda \sum_{u,v} |\partial_u P(u, v)| + \mu \sum_{u,v} |\partial_v P(u, v)|; \tag{3}$$

where  $\partial_u P(u, v) = P(u+1, v) - P(u, v)$  denotes the discrete version of the partial derivative operator;  $\lambda$  and  $\mu$  are discontinuity weights or regularization parameters that add smoothness constraints to the sought solution  $P(u, v)$ .

It is natural to assume that the best matching of stereo images is achieved with the disparity map  $P(u, v)$ , which minimizes the conditional distance  $\mathbb{R}$  between the images. Thus, we can formulate our problem as follow: find the disparity map function  $P(u, v)$ , which minimize the conditional distance between the matched stereo images  $I_1$  and  $I_2$  i.e.

$$P(u, v) = \arg \min_P \left\{ \sum_{u,v} E_0(u, v, P) + \lambda \sum_{u,v} |\partial_u P| + \mu \sum_{u,v} |\partial_v P| \right\}. \tag{4}$$

Unfortunately, the exact solution of such a problem can be found only with exponentially complex algorithms. Let us consider two special cases of the conditional distance  $\mathbb{R}$  that allow the exact solution of Eq. (4) using non-exponentially complex algorithms. Those are:

- **$\lambda$  and  $\mu=0$ .** This special case does not imply any smoothness constraints of the disparity map function  $P(u, v)$ , and thus the general problem is naturally reduced to  $U_{max} \times V_{max}$  the mutually independent sub-problems of the pixel-to-pixel matching. For such a singularity of the conditional distance  $\mathbb{R}$ , the minimal projection  $P_0(u, v)$  of the initial DSI calculated by Eq. (2) satisfies the exact solution of Eq. (4). An example of a disparity map without smoothness constraints is illustrated in Fig. 2.(e).
- **$\lambda$  or  $\mu=0$ .** This special case implies only one-dimensional smoothness constraints of the disparity map function  $P(u, v)$ , and thus the general problem can be reduced to  $U_{max}$  or  $V_{max}$  the mutually independent sub-problems of the one-dimensional optimization. Such problems can be solved by means of the DP algorithm. An example of a disparity map with one-dimensional smoothness constraints is illustrated in Fig. 2. (f).

Let us consider the DP algorithm for the case  $\mu=0$ . Then Eq. (4) is rewritten as

$$P(u | v_0) = \arg \min_P \left\{ \sum_u E_0(u, v_0, P(u | v_0)) + \lambda \sum_u |\partial_u P(u | v_0)| \right\}, \tag{5}$$

where  $v_0$  is a fixed index of  $V$  dimension of the processed DSI. The final solution is a bunch of one-dimensional functions  $P(u | v_0)$  with different fixed index  $v_0$ , where each discrete function  $P(u | v_0)$  coincides with the optimal path through the 2D DSI trellis as it is shown in Fig. 3. Here the 2D DSI is a slice ( $v = v_0$ ) of the 3D DSI like in Fig. 2 (c). The term “optimal” means that the sum of the cost values along this path plus the weighted length of the path is minimal among all other possible paths.

The optimal path problem is usually addressed with the method of dynamic programming. The method consists of step-by-step control and optimization that is given by a recurrence relation

$$S_n(u, d) = E_n(u, d) + \min_{d+k \in [0, D_{max}]} \{ S_n(u-1, d+i) + \lambda |d+k| \}, \tag{6}$$

$$S_n(0, d) = E_n(0, d).$$

By using the recurrence relation the minimal value of the objective function in Eq.(5) can be found at the last step of optimization. Next, the algorithm works in

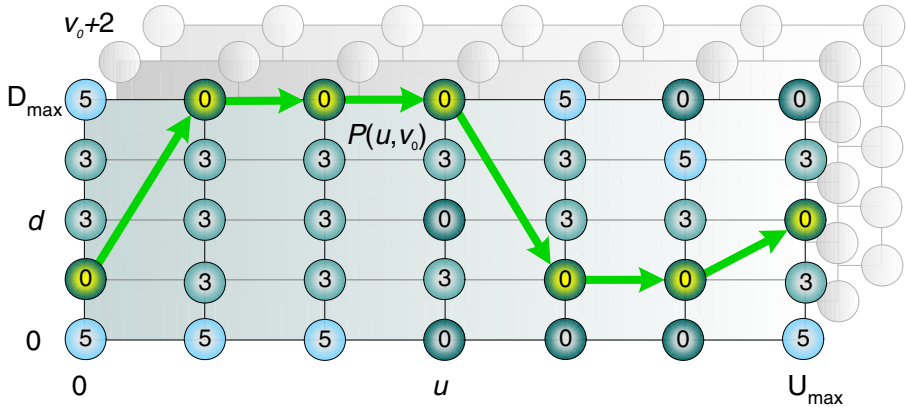


Fig. 3. The optimal path trough the DSI trellis

reverse order and recovers a sequence of optimal steps (using the lookup table  $K(u,d)$  of the stored values of the index  $k$  in the recurrence relation (6)), and eventually the optimal path by

$$\begin{aligned}
 P(u-1) &= P(u) + K(u, P(u)), \\
 P(U_{\max}) &= \arg \min_d S(U_{\max}, d).
 \end{aligned}
 \tag{7}$$

Unfortunately, the DP algorithm does not solve 2D optimization problem. However, the recurrence computational scheme Eq. (6) might be used as the base for an effective cost aggregation, and we show the way in the next section.

### 3 Optimal Path Cost Aggregation Algorithm

In this section we introduce the OPCA as a simple transformation of the 2D DSI matrix, i.e.

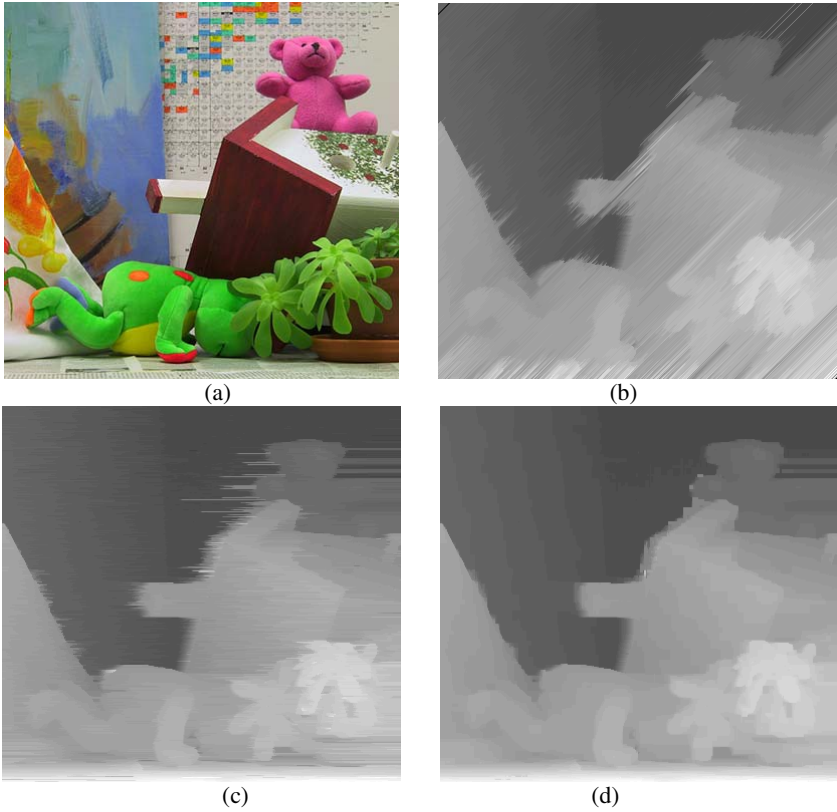
$$E_{n+1}(u, d) = A\{E_n(u, d)\},
 \tag{8}$$

where  $A\{\bullet\}$  denotes the OPCA operator. The operator must replace the cost values  $E_n(u,d)$  of the processed 2D DSI to the value that is equal to the sum of the costs over the sub-optimal path, which include the particular point  $(u, d)$ :

$$E_{n+1}(u, d) = \min_{d=P(u)} \left\{ \sum_{j=0}^{U_{\max}} E_n(j, P(j)) + \lambda \sum_{j=0}^{U_{\max}} |\partial_j P(j)| \right\}.
 \tag{9}$$

Let us divide the sought sub-optimal path in Eq. (9) into two region, i.e.

$$\begin{aligned}
 E_{n+1}(u, d) &= \min_{d=P(u)} \left\{ \sum_{j=0}^u (E_n(j, P(j)) + \lambda |\partial_j P(j)|) \right\} + \\
 &+ \min_{d=P(u)} \left\{ \sum_{j=u}^{U_{\max}} (E_n(j, P(j)) + \lambda |\partial_j P(j)|) \right\} - E_n(u, d).
 \end{aligned}
 \tag{10}$$



**Fig. 4.** (a) Test stereo image Teddy. (b) Disparity map with diagonal OPCA. (c) Disparity map with v-directional OPCA. (d) Disparity map with two-pass OPCA.

We note that the DP algorithm can be used in the reverse index direction. In such a case the recurrence operator in Eq. (6) is rewritten

$$\begin{aligned}
 S_n^R(u, d) &= E_n(u, d) + \min_{d+i \in [0, D_{\max}]} \{S_n(u+1, d+i) + \lambda|d+i|\}, \\
 S_n(U_{\max}, d) &= E_n(U_{\max}, d).
 \end{aligned}
 \tag{11}$$

Finally, using the direct recurrence in Eq. (6) and the reverse recurrence in Eq. (11), the optimal sum in Eq. (10) can be evaluated, yielding:

$$E_{n+1}(u, d) = A\{E_n(u, d)\} = S_n^R(u, d) + S_n(u, d) - E_n(u, d).
 \tag{12}$$

Now, if we apply the OPCA operator Eq. (12) to each  $v$  slice of the initial 3D DSI matrix  $E_0(u, v, d)$ , as it is shown in the scheme Fig.2., then the sought disparity map  $P(u, v)$  is just the minimal projection Eq. (2) of the aggregated DSI. In this case the solution in Fig. 4.(c) exactly coincides with the disparity map obtained by the DP algorithm. However, the key benefit of the proposed scheme is that we rid of the graph recovery in Eq. (7). Instead of this, we handle the aggregated costs of the DSI. Consequently, of the OPCA operator can be applied repeatedly.



For instance, if the first application of the OPCA is  $v$ -directional (the 3D DSI is cut to the  $V_{max}$  2D DSI sub-matrixes as in the scheme Fig. 2.), next we can apply the OPCA in the  $u$ -direction (the 3D DSI is cut to the  $U_{max}$  2D DSI sub-matrixes). In other words, the two-pass cost aggregation is carried out:

$$E_{uv}(u, v, d) = A_u \left\{ A_v \left\{ E_0(u, v, d) \right\} \right\}. \quad (13)$$

The cost value  $E_{uv}(u, v, d)$  is proportional to the sum of the initial costs over optimal surface Fig. 1. It means that the solution obtained with the two-pass OPCA is an approximation to 2D optimization problem. The result of such approximation is illustrated by the disparity map in Fig. 4. (d). The comparison with other global optimization algorithms (Graph Cuts and Belief Propagation) demonstrates that even the two-pass OPCA algorithm achieves the better result. Furthermore, to improve the resultant disparity map the base operator Eq. (12) can be applied in other directions different from the above considered  $u$  and  $v$  directions (e.g., in a diagonal direction as it is shown in Fig. 4. (b)).

For example, the four-pass OPCA can be calculated by

$$E_{\wedge uv}(u, v, d) = A_{\wedge} \left\{ A_u \left\{ A_v \left\{ E_0(u, v, d) \right\} \right\} \right\}. \quad (14)$$

The proposed algorithm with use of the OPCA is now summarized as follows:

- Prepare a 3D DSI array,  $(u, v, d):(x, y)$  and set initial cost values  $E_0$  using Eq. (1).
- Iteratively update cost values  $E_n$  using the OPF that is defined in Eq. (12) with various DSI cut directions, until the quality of minimal projection of the processed DSI does not improve.
- To carry out the quality evaluation use the conditional distance criterion defined in Eq. (3).
- Utilize the minimal projection Eq. (2) of the updated DSI as the sought disparity map.

Usually four steps of the iteration process Eq. (14) are sufficient. The running time of each OPCA iteration is on the order of  $O(UVD)$ , where  $UV$  is approximately the size of the stereo image;  $D$  is the range of disparities.

## 4 Computer Experiments

In this section experimental results are presented to illustrate the performance of the OPCA method. To demonstrate the effectiveness of our algorithm, we have applied it to several test stereo images. Initial cost values are set by using the absolute difference of image intensities for each pixel. For the color images the norm has to be the vector distance (the square root of the sum of the squared differences of the color vector components). In our computer experiment both regularization weights  $\lambda = \mu$  is set proportional to the mean value of the initial costs of the processed DSI, because such a choice refers to the early path definitions [8], where the path roughness was penalized by additional cost values included in this path.

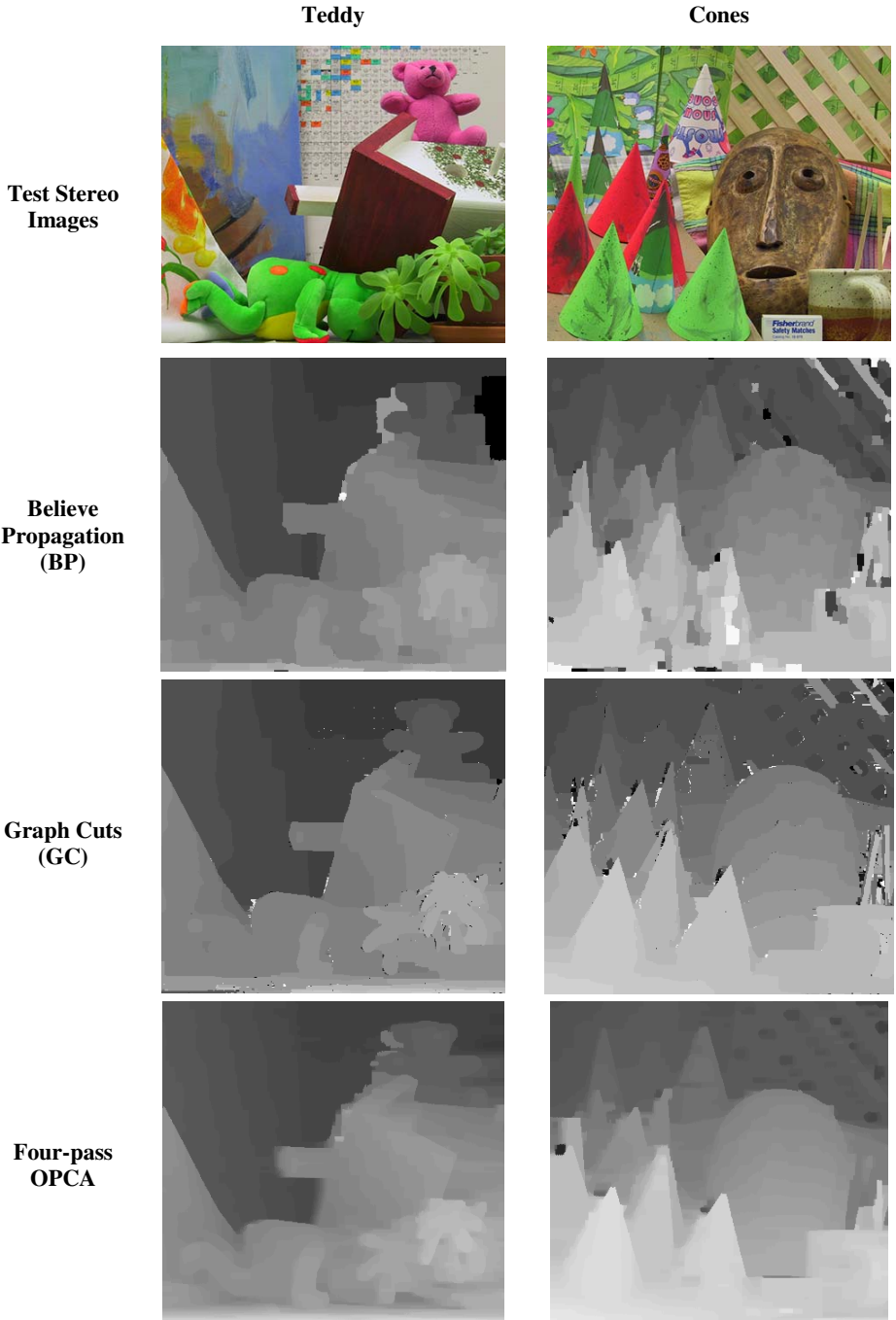


Fig. 5. Comparative results of different stereo methods

Fig. 5 presents the comparative results of different global stereo matching methods on two test stereo images "Teddy" (450×375 Disp. 32) and "Cones" (450×375 Disp. 32). The quantitative evaluation of the processing results is based on the Mean Squared Error criterion relative to the ground truth disparity maps and given in Table 1.

**Table 1.** Quantitative evaluation of the results in Fig. 5

Methods	Teddy	Cones
BP	1.58	2.21
GC	0.91	1.34
OPCA	0.63	0.79

So numerical analysis on the base of the mean squared errors (MSE) criterion shows that the proposed algorithm has advantage over conventional global optimization matching algorithms. Furthermore, the OPCA algorithm is much faster than global methods.

## 5 Conclusion

A new stereo matching method based on the OPCA technique has been proposed. The proposed aggregation technique is based on the dynamic programming principle, which exactly solves one dimensional optimization problem. The OPCA can be utilized repeatedly in contrast to dynamic programming. Furthermore, the application of the 2D extension of the OPCA proves an excellent approximation to the global 2D optimization problem.

## Acknowledgements

This work has been supported by EC grant IST-027110 for the HERMES project and by the Spanish MEC under projects TIC2003-08865. M. Mozerov also acknowledges the support of the Ramon y Cajal research program, Ministerio de Educacion y Ciencia, Spain.

## References

1. Bhatand, D. N., Nayar, S. K.: Ordinal measures for image correspondence. *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 20 (1998) 415-423
2. Brown, M. Z., Burschka, D., and Hager, G. D.: Advances in computational stereo. *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 25 (2003) 993-1008
3. Gong, M., and Yang, Y-H.: Fast unambiguous stereo matching using reliability-based dynamic programming. *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 27 (2005) 998-1003

4. Hirschmuller, H.: Accurate and efficient stereo processing by semi-global matching and mutual information. Proc. Computer Vision and Pattern Recognition (CVPR 2005), Vol. 2 (2005) 807-814
5. Kim, J., Kolmogorov, V., and Zabih, R.: Visual correspondence using energy minimization and mutual information. Proc. International Conference on Computer Vision, (2003) 1033–1040
6. Lin, M. H., and Tomasi, C.: Surfaces with occlusions from layered stereo. IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 26 (2004) 073–1078
7. Ohta, Y., and Kanade, T.: Stereo by intra – and intra-scanline search using dynamic programming. IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 7 (1985) 139-154
8. Roy, S., and Cox, I. J.: A maximum-flow formulation of the N-camera stereo correspondence problem. Proc. Int’l Conf. Computer Vision, (1998) 492-499
9. Scharstein, D., and Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. International Journal of Computer Vision, Vol. 47 (2002) 7–42
10. Sun, J., Shum, H. Y., and Zheng, N. N.: Stereo matching using belief propagation. IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 25 (2003) 787–800
11. Tomasi, C., and Manduchi, R.: Stereo matching as a nearest-neighbor problem. IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 20 (1998) 333–340
12. Zhao, H.: Global optimal surface from stereo. Proc. Int’l Conf. Pattern Recognition, Vol. 1 (2000) 101-104
13. Zitnik, C. L., and Kanade, T.: A cooperative algorithm for stereo matching and occlusion detection. IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 22 (2000) 675-684

# Tracking Camera Parameters of an Active Stereo Rig

Thao Dang and Christian Hoffmann

Institut für Mess und Regelungstechnik, University of Karlsruhe, Germany

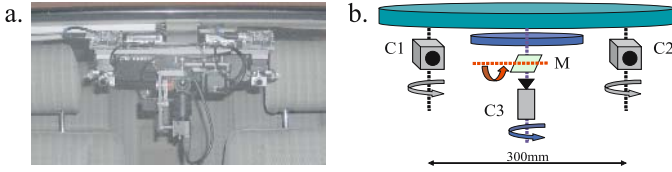
**Abstract.** This contribution presents an approach for the continuous self-calibration of an active stereo rig with verging cameras. The proposed self-calibration recovers extrinsic parameters up to scale as well as the focal lengths of both cameras. Three different categories of constraint equations are evaluated and formulated as a Gauss-Helmert model for self-calibration: bundle adjustment with reduced parameter vector, the epipolar constraint, and the trilinear constraints. The optimization of the constraints is implemented as a robust Iterated Extended Kalman Filter that allows initial stereo calibration as well as continuous tracking of the camera parameters. The performance of the algorithm is demonstrated on synthetic and real imagery.

## 1 Introduction

Stereo self-calibration refers to the automatic determination of extrinsic and intrinsic camera parameters of a stereo rig without prior information about the observed scene. Particularly, no special calibration objects are required and the calibration can be maintained continuously while the sensor is in use. We believe that especially in the automotive field, self-calibration is a vital ability required for the introduction of stereo cameras in the market. The objective of this contribution differs from many self-calibration tasks since we assume that an initial guess of the camera calibration is readily available (e.g. camera orientation is given with errors up to a few degrees), and our self-calibration has to refine these initial guesses and track slow drift in the camera calibration parameters. We thus address only part of the full self-calibration problem, however, we believe that this simplification is valid for a variety of applications e.g. when the parameters of the stereo rig are given within the tolerances of the manufacturing process or in active vision when (perturbed) commanded camera parameters are available.

We have developed an active camera platform consisting of three cameras (Fig. 1): one tele camera for monoscopic vision tasks and two cameras for stereopsis. The camera platform is capable of vergence, i.e. all cameras can be rotated independently. The vision sensor is intended to implement active vision capabilities for autonomous driving. Additionally, the platform constitutes an excellent test bed for stereo self-calibration.

Several approaches to the calibration of an active stereo rig with verging cameras have been described in the literature: Some use careful offline calibration of the camera parameters and the motor axes and then rely solely on the



**Fig. 1.** a. The active camera platform mounted in our experimental vehicle. The platform consists of two (stereo) cameras and a tele camera. The two stereo cameras can be rotated independently about their yaw axes while the tele camera is capable of both horizontal and vertical rotations to compensate vehicle pitch. b. Schematic view of the active camera platform (C1, C2: stereo cameras, C3: tele camera, M: mirror).

commanded motor angles [1]. Others employ online self-calibration, but restrict themselves to update only two extrinsic camera orientation angles and thus inherently assume ideal mechanical setups [2,3].

The goal of this work is to derive a recursive approach that continuously updates five extrinsic parameters and the focal lengths of both cameras by means of a robust, iterated extended Kalman filter (IEKF). We investigate three different constraints that may be used in tracking the calibration parameters: recursive bundle adjustment with reduced dimension of the parameter state vector, the epipolar constraint between a pair of stereo images, and the trifocal constraint. A Gauss-Helmert type model is employed to ensure that physically relevant errors are minimized. We find that recursive bundle adjustment and the epipolar constraint may complement each other in practical applications: bundle adjustment provides highest accuracy (and might suffice on its own in some environments), the epipolar constraint is not affected by independently moving objects in the scene and may thus stabilize the calibration process. We propose an algorithm that combines both the epipolar constraint for instantaneous measurements and recursive bundle adjustment integrating spatio-temporal correspondences. The algorithms are demonstrated on both synthetic and real imagery.

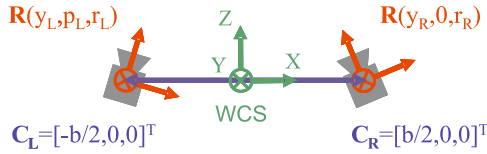
The paper is organized as follows: Sec. 2 briefly outlines the mathematical camera model used to describe our active vision system. The recursive self-calibration based on a robust IEKF and the different geometric constraints are presented in Sec. 3. Our algorithm is evaluated on synthetic and real-life imagery (Sec. 4). Sec. 5 summarizes our results and concludes the paper.

## 2 Camera Model

Throughout this paper, we employ the ideal pinhole model to describe the stereo cameras (see e.g. [4]). Using this model, a 3d point  $\mathbf{X}$  and its 2d image coordinates  $\mathbf{x}$  (both in homogenous coordinates) are related by the following projective equation

$$\mathbf{x} = \lambda \mathbf{K} \mathbf{R} [\mathbf{I}, -\mathbf{C}] \mathbf{X} = \lambda \mathbf{P} \mathbf{X} \quad \text{with } \mathbf{K} = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix}, \quad (1)$$

where  $\lambda$  is an unknown scalar factor. The matrix  $\mathbf{K}$  comprises the intrinsic camera parameters, i.e. the focal length  $f$  and the image center  $[c_x, c_y]^T$ . Aspect



**Fig. 2.** Extrinsic parameters of active stereo rig. The world coordinate system (WCS) is attached to the baseline  $b$  of the stereo cameras.

ratio and image skew are omitted here since they are negligible in most modern CCD and CMOS cameras. The rotation matrix  $\mathbf{R}$  and the vector  $\mathbf{C}$  specify the extrinsic camera parameters. For simplicity, we abbreviate perspective projection according to (1) by  $\mathbf{x} = \pi(\mathbf{X})$ . The inversion of the pinhole projection will be denoted as  $\mathbf{X} = \Pi^{-1}(\mathbf{x}, X_Z)$ . Please note that depth component  $X_Z$  of  $\mathbf{X}$  is required for a unique reprojection of  $\mathbf{x}$ .

Fig. 2 depicts the extrinsic parameters of our active camera platform. We define a world coordinate system (WCS) whose origin lies in the center of the baseline and whose  $X$ -axis is aligned with the baseline. To eliminate the remaining degree-of-freedom (DOF), we impose that the  $Z$ -axis of the WCS is parallel to the plane defined by the baseline and the optical axes of the right camera. Given this WCS, it is convenient to represent camera orientations as a concatenation of yaw–pitch–roll rotations with angles  $\omega = [y, p, r]^T$ :

$$\mathbf{R}(\omega) = \mathbf{R}(y, p, r) = \mathbf{R}_Z(r) \mathbf{R}_X(p) \mathbf{R}_Y(y) , \tag{2}$$

where  $\mathbf{R}_Z, \mathbf{R}_X, \mathbf{R}_Y$  are rotations about the  $Z$ -,  $X$ -, and  $Y$ -axis, respectively.

The projection matrices  $\mathbf{P}_L$  and  $\mathbf{P}_R$  can be obtained from Fig. 2:

$$\mathbf{P}_R = \mathbf{K}_R \mathbf{R}_R [\mathbf{I}, -\mathbf{C}_R] \tag{3}$$

$$\mathbf{P}_L = \mathbf{K}_L \mathbf{R}_L [\mathbf{I}, -\mathbf{C}_L] \tag{4}$$

with  $\mathbf{R}_R = \mathbf{R}(y_R, 0, r_R), \mathbf{R}_L = \mathbf{R}(y_L, p_L, r_L), \mathbf{C}_R = [b/2, 0, 0]^T$  and  $\mathbf{C}_L = [-b/2, 0, 0]^T$ .

The stereo rig has thus six extrinsic parameters:  $y_L, p_L, r_L$  denote the orientation of the left camera with respect to the WCS,  $y_R, r_R$  specify the yaw and roll angles of the right camera (the pitch angle is omitted since the WCS is aligned with the optical axis of the right camera), and  $b$  is the base length of the stereo cameras. Since cameras are scale blind, we will not update  $b$  in our self-calibration and assume that the base length is precisely known.

### 3 Self-calibration

**Bundle adjustment.** Consider a set of object points  $\mathbf{X}_i, i \in S_b$  that is moving rigidly through space, i.e. with a simple motion model  $\mathbf{X}_i(k+1) = \mathbf{R}(\omega(k))\mathbf{X}_i(k) + \mathbf{V}(k)$ . The projections of these object points in the left and right images are denoted  $\mathbf{x}_{R,i}, \mathbf{x}_{L,i}$ , respectively, and we are given noisy measurements  $\hat{\mathbf{x}}_{R,i}(k) = \mathbf{x}_{R,i}(k) + \mathbf{e}_{R,i}(k), \hat{\mathbf{x}}_{L,i}(k) = \mathbf{x}_{L,i}(k) + \mathbf{e}_{L,i}(k)$ .

The objective of bundle adjustment is to find the 3d structure  $\mathbf{X}_i$ , the object motion  $(\omega, \mathbf{V})$ , and the camera parameters  $(\omega_R, \omega_L, f_R, f_L)$  such that the distance between the projections  $\mathbf{x}_{R,i}, \mathbf{x}_{L,i}$  and the measured coordinates  $\hat{\mathbf{x}}_{R,i}, \hat{\mathbf{x}}_{L,i}$  is minimal over all frames  $k$  of a sequence. Bundle adjustment has been widely used in photogrammetry and as a refinement step for off-line camera calibration since it can provide highly accurate results. However, it has several shortcomings: First, it requires an initial guess of the parameters with sufficient quality to guarantee convergence. Second, it is usually implemented as a batch approach that requires that all input data is given at once. Third, the parameter space is high dimensional since each tracked point  $\mathbf{X}_i$  introduces three additional DOF, resulting in difficult and time consuming optimization procedures. As stated earlier, we assume that a sufficient initial guess is available and cover only the latter problems. A robust Iterated Extended Kalman Filter will be used, so that all data will be processed as soon as it arrives. To reduce the state dimension, we decompose each  $\mathbf{X}_i$  into its projection onto the right image  $\mathbf{x}_{R,i}$  and its depth  $\rho_i$ :  $\rho_i$  cannot be recovered directly and is thus included in the parameter vector, whereas  $\mathbf{x}_{R,i}$  is treated as a (directly accessible) observation in the measurement constraint as will be shown later. Thus, in our formulation each tracked parameter introduces only one DOF and the dimension of the state vector is reduced significantly.

Given a true image position  $\mathbf{x}_{R,i}(k)$  and the true depth  $\rho_i(k)$  of a tracked point, we can reconstruct  $\mathbf{X}_i(k)$  via inverse pinhole projection

$$\mathbf{X}_i(k) = \mathbf{\Pi}_R^{-1}(\mathbf{x}_{R,i}(k), \rho_i(k)). \tag{5}$$

Using  $\mathbf{X}_i(k)$ , we are able to predict image positions  $\mathbf{x}_R(k+1)$  and  $\mathbf{x}_L(k)$ :

$$\begin{aligned} \mathbf{x}_R(k+1) &= \pi_R(\mathbf{R}(\omega(k))\mathbf{X}(k) + \mathbf{V}(k)) \\ \mathbf{x}_L(k) &= \pi_L(\mathbf{X}(k)) \end{aligned} \tag{6}$$

Thus, for each time instant  $k$ , Eqs. (5) and (6) constitute an implicit measurement constraint between the observed quantities and the parameter vector  $\mathbf{z}_b = [\rho_i, \omega, \mathbf{V}, \omega_R, \omega_L, f_R, f_L]^T$

$$\mathbf{h}_b(\mathbf{z}_b, \mathbf{x}_{R,i}(k), \mathbf{x}_{L,i}(k), \mathbf{x}_{R,i}(k+1)) = \begin{bmatrix} \pi_R\{\mathbf{R}(\omega)\mathbf{\Pi}_R^{-1}(\mathbf{x}_{R,i}(k), \rho_i) + \mathbf{V}\} - \mathbf{x}_{R,i}(k+1) \\ \pi_L\{\mathbf{\Pi}_R^{-1}(\mathbf{x}_{R,i}(k), \rho_i)\} - \mathbf{x}_{L,i}(k) \end{bmatrix} = \mathbf{0} \tag{7}$$

Using this constraint, the objective of our self-calibration is to minimize at each  $k$  the pixel error <sup>1</sup>

$$\sum_{i \in S_b} \left\| (\mathbf{e}_{R,i}(k), \mathbf{e}_{L,i}(k), \mathbf{e}_{R,i}(k+1))^T \right\|^2 \tag{8}$$

subject to the constraint

$$\mathbf{h}_b(\mathbf{z}_b, \hat{\mathbf{x}}_{R,i}(k) - \mathbf{e}_{R,i}(k), \hat{\mathbf{x}}_{L,i}(k) - \mathbf{e}_{L,i}(k), \hat{\mathbf{x}}_{R,i}(k+1) - \mathbf{e}_{R,i}(k+1)) = \mathbf{0} \tag{9}$$

---

<sup>1</sup> In fact, each pixel error is actually weighted by its inverse covariance matrix. This is omitted for brevity.



evaluated for all features  $i \in S_b$ . Implicit measurement constraints as given by Eqs. (8–9) are related to Gauss-Helmert models (e.g. [5]). We wish to emphasize that Eq. (8) minimizes a physically relevant geometric error corresponding to pixel distances in the image. In addition, the dimension of the parameter vector in our optimization problem is  $N+13$  (where  $N$  is the number of tracked points), while standard bundle adjustment would require  $3N+13$  elements.

**Epipolar constraint.** The epipolar constraint was introduced by [6] and constitutes an elementary relation between two stereo images. Geometrically, it states that the optical centers of both the left and right camera and corresponding image points  $\mathbf{x}_L$  and  $\mathbf{x}_R$  all lie within the same plane. This constraint is expressed mathematically using the *fundamental matrix*  $\mathbf{F}$ :

$$h_e(\mathbf{F}, \mathbf{x}_L, \mathbf{x}_R) = \mathbf{x}_L^T \mathbf{F} \mathbf{x}_R = 0. \quad (10)$$

It is straightforward to show that for our camera model described in Sec. 2, the fundamental matrix is given by

$$\mathbf{F} = \mathbf{K}_L^{-T} \mathbf{R}_L [\mathbf{C}_R - \mathbf{C}_L]_{\times} \mathbf{R}_R^T \mathbf{K}_R^{-1}, \quad (11)$$

where  $[\cdot]_{\times}$  denotes the skew-symmetric matrix operator. Please note that the epipolar constraint (10) does not involve the 3d position of the observed object point, i.e. the epipolar constraint decouples the extrinsic camera parameters from the 3d structure of the observed scene. Furthermore, the epipolar constraint constitutes only a necessary condition for two image points to correspond to the same object point since it neglects matching errors along the epipolar line.

Given noisy image positions  $\hat{\mathbf{x}}_{R,i} = \mathbf{x}_{R,i} + \mathbf{e}_{R,i}$  and  $\hat{\mathbf{x}}_{L,i} = \mathbf{x}_{L,i} + \mathbf{e}_{L,i}$ , our objective is to find the camera parameters  $(\omega_R, \omega_L, f_R, f_L)$  that minimize the sum of squared pixel errors:

$$\sum_{i \in S_e} \left\| (\mathbf{e}_{R,i}, \mathbf{e}_{L,i})^T \right\|^2 \quad (12)$$

subject to the epipolar constraints

$$h_e(\mathbf{F}, \hat{\mathbf{x}}_{L,i} - \mathbf{e}_{L,i}, \hat{\mathbf{x}}_{R,i} - \mathbf{e}_{R,i}) = 0 \quad \forall i \in S_e. \quad (13)$$

Even though the epipolar constraint has some theoretical disadvantages compared to bundle adjustment since it does not provide as much information, it still has some practical benefits: First, the parameter space for self-calibration is small since an explicit representation of the scene structure is not required. Second, the epipolar constraint between stereo images does not require the rigidity of the observed scene and is thus unaffected by independently moving objects.

**Trilinear constraints.** The trilinear conditions formulated in [7] relate the coordinates of corresponding points in three images. Similar to the epipolar constraint, the trilinearities decouple scene structure from camera calibration since they do not require the 3d position of the observed point explicitly. Contrary to the epipolar constraint, however, they provide a *sufficient* condition for three

image coordinates to correspond to the same object point. For more details on the trilinear constraint, the reader is referred to [4].

Consider a triplet of corresponding image points  $\mathbf{x}_{R,i}(k), \mathbf{x}_{L,i}(k), \mathbf{x}_{R,i}(k+1)$  in the current right, left and subsequent right camera frame, respectively. For brevity, we denote these image positions by  $\mathbf{x}^A, \mathbf{x}^B, \mathbf{x}^C$  and their associated projection matrices are given by

$$\mathbf{A} = \mathbf{K}_R \mathbf{R}_R [\mathbf{I}, -\mathbf{C}_R] \tag{14}$$

$$\mathbf{B} = \mathbf{K}_L \mathbf{R}_L [\mathbf{I}, -\mathbf{C}_L] \tag{15}$$

$$\mathbf{C} = \mathbf{K}_R \mathbf{R}_R [\mathbf{R}(\omega(k)), \mathbf{V}(k) - \mathbf{C}_R] , \tag{16}$$

where  $(\omega(k), \mathbf{V}(k))$  describes the 3d motion of the stereo rig.

The geometry of three cameras can be captured elegantly by the *trifocal tensor* [8]. In this contribution, we employ a Euclidean parametrization of the trifocal tensor and compute  $\mathbf{T}$  from the projection matrices  $\mathbf{A}, \mathbf{B}, \mathbf{C}$  as

$$\mathbf{T}_l^{qr} = (-1)^{4+l} \det \begin{bmatrix} \sim \mathbf{a}^l \\ \mathbf{b}^q \\ \mathbf{c}^r \end{bmatrix} . \tag{17}$$

$\mathbf{b}^q$  and  $\mathbf{c}^r$  refer to the  $q$ -th and  $r$ -th row of the matrices  $\mathbf{B}$  and  $\mathbf{C}$ , respectively.  $\sim \mathbf{a}^l$  is the matrix  $\mathbf{A}$  without the  $l$ -th row.

Using the trifocal tensor, the trilinear constraints between point triplets are given by:

$$g_{qr}(\mathbf{T}, \mathbf{x}^A, \mathbf{x}^B, \mathbf{x}^C) = \sum_{l=1}^3 \mathbf{x}_l^A \left( \mathbf{x}_q^B \mathbf{x}_r^C T_l^{33} - x_r^C T_l^{q3} - x_q^B T_l^{3r} + T_l^{qr} \right) = 0 . \tag{18}$$

Eq. (18) actually yields nine constraints for the possible choices of  $q, r \in \{1, 2, 3\}$ . Four of these constraints are linearly independent [7], but as shown in [5], the trilinearities impose only three constraints onto the geometry of the image triplet if a minimal parametrization is used. An optimal choice of the constraints is non trivial, in fact the selection of the constraints should be adapted to the current motion of the stereo and the position of the observed 3d point. This has not yet been implemented in our work. Instead, we found that for our stereo rig with fixed base length, a combination of two trilinear constraints  $(q, r) = (1, 1), (1, 2)$  and the epipolar constraint between the left and right stereo frame gives adequate results.

Using the selected constraints and given  $i$  triplets of corresponding points in three images, our self-calibration algorithm has to minimize the cost function

$$\sum_{i \in S_t} \left\| (\mathbf{e}_{R,i}(k), \mathbf{e}_{L,i}(k), \mathbf{e}_{R,i}(k+1))^T \right\|^2 \tag{19}$$

subject to the constraint

$$\begin{bmatrix} g_{11}(\mathbf{T}, \hat{\mathbf{x}}_{R,i}(k) - \mathbf{e}_{R,i}(k), \hat{\mathbf{x}}_{L,i}(k) - \mathbf{e}_{L,i}(k), \hat{\mathbf{x}}_{R,i}(k+1) - \mathbf{e}_{R,i}(k+1)) \\ g_{12}(\mathbf{T}, \hat{\mathbf{x}}_{R,i}(k) - \mathbf{e}_{R,i}(k), \hat{\mathbf{x}}_{L,i}(k) - \mathbf{e}_{L,i}(k), \hat{\mathbf{x}}_{R,i}(k+1) - \mathbf{e}_{R,i}(k+1)) \\ h_e(\mathbf{F}, \hat{\mathbf{x}}_{L,i} - \mathbf{e}_{L,i}, \hat{\mathbf{x}}_{R,i} - \mathbf{e}_{R,i}) \end{bmatrix} = \mathbf{0} \tag{20}$$

for all  $i \in S_t$ .

**Recursive optimization.** In the preceding sections we have derived three geometric optimization criteria. If we assume that the measurement errors  $\mathbf{e}_R, \mathbf{e}_L$  are realizations of a Gaussian white noise process, we can solve our self-calibration problem using an Iterated Extended Kalman filter (IEKF) with implicit measurement formulation as described in e.g. [9]. The state vector  $\mathbf{z}$  of the IEKF comprises the depths of all  $N$  tracked bundle adjustment features in the set  $S_b$ , object motion and camera parameters:

$$\mathbf{z} = [\rho_1, \dots, \rho_N, \omega, \mathbf{V}, y_R, r_R, y_L, p_L, r_L, f_R, f_L]^T . \tag{21}$$

While the measurement equations of the filter are given by Eqs. (8,9), (12,13) and (19,20), we still need to define the system model that governs the dynamics of our state vector. For simplicity, we assume that the camera is moving with a constant velocity model pertubated by Gaussian white noise, i.e.

$$\begin{bmatrix} \omega(k+1) \\ \mathbf{V}(k+1) \end{bmatrix} = \begin{bmatrix} \omega(k) \\ \mathbf{V}(k) \end{bmatrix} + \begin{bmatrix} \mathbf{n}_\omega(k) \\ \mathbf{n}_V(k) \end{bmatrix} . \tag{22}$$

If additional information is available (such as e.g. commanded steering angles and accelerations or more precise vehicle motion models), it should be incorporated at this point. However, we found that the simple motion model already provides good results in our first experiments. The depths  $\rho_i$  then evolve as

$$\rho_i(k+1) = [0, 0, 1] (\mathbf{R}(\omega(k)) \mathbf{X}_i(k) + \mathbf{V}(k)) , \quad i \in S_b \tag{23}$$

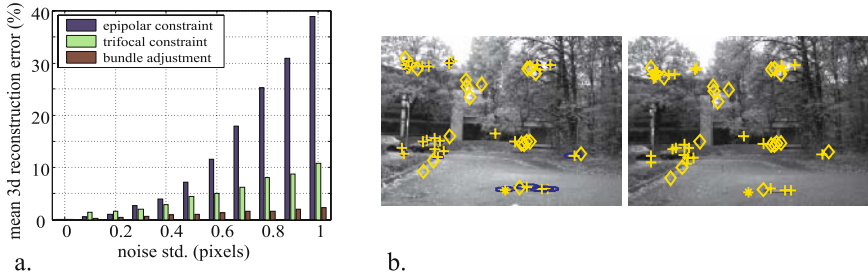
with  $\mathbf{X}_i(k) = \mathbf{\Pi}_R^{-1}(\mathbf{x}_{R,i}(k), \rho_i(k))$ . The dynamics of the extrinsic camera parameters are assumed to be governed by

$$\begin{bmatrix} y_R(k+1) \\ r_R(k+1) \\ y_L(k+1) \\ p_L(k+1) \\ r_L(k+1) \end{bmatrix} = \begin{bmatrix} y_R(k) \\ r_R(k) \\ y_L(k) \\ p_L(k) \\ r_L(k) \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_R(k) \\ u_L(k) \end{bmatrix} + \mathbf{n}_{\omega_R, \omega_L}(k) , \tag{24}$$

where  $u_R, u_L$  denote the commanded yaw angles of the right and left stereo camera, respectively. The covariance matrix of the system noise  $\mathbf{n}_{\omega_R, \omega_L}$  is small when no command signals have been sent to the motors and large when new gaze directions are set. Similarly, constant focal lengths are assumed with additive Gaussian white noise  $\mathbf{n}_f$

$$\begin{bmatrix} f_R(k+1) \\ f_L(k+1) \end{bmatrix} = \begin{bmatrix} f_R(k) \\ f_L(k) \end{bmatrix} + \mathbf{n}_f(k) . \tag{25}$$

Feature points for recursive bundle adjustment, epipolar constraint, and trilinear constraints are acquired using Lowe’s SIFT feature detector [10]. In addition, the search region used for feature matching is predicted using the current filter state and its uncertainty. As indicated above, robustness is an essential property of our self-calibration algorithm since correlation based matching is prone to occasional gross errors due to periodic patterns or occlusions and there may be independently moving objects in the scene. We have thus employed random sampling in the innovation stage of the IEKF as proposed in [11].



**Fig. 3. a.** Comparison of self-calibration results on various noise levels. The plot shows the mean 3d reconstruction error obtained with the different self-calibration methods. **b.** Second and third stereo frame of sample sequence. The automatically selected features are also shown: +: successfully tracked features,  $\diamond$ : stereo features for epipolar constraint, \*: invalid tracking features.

## 4 Examples

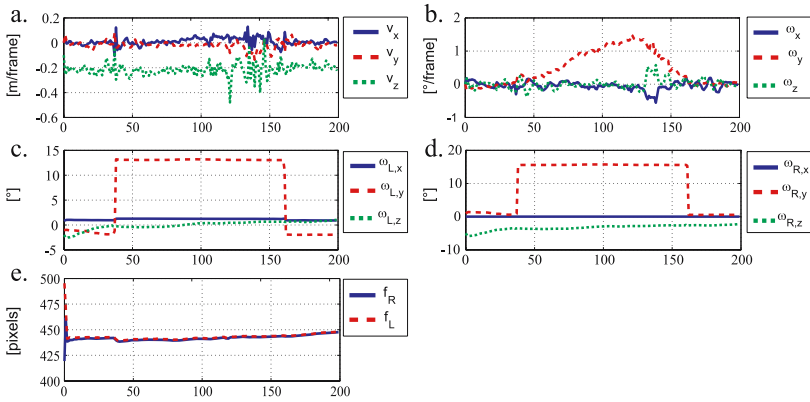
The proposed self-calibration was first evaluated on simulated data. We randomly generated synthetic stereo sequences of a moving point cloud with 40 points. Each sequence was 50 stereo frames long and Gaussian white noise was added to the image coordinates of all points in both images. The initial guess for the stereo calibration deviated  $2^\circ$  in each component from the true extrinsic parameters and differed by 10% from the true focal lengths.

To assess the self-calibration results, we compute the mean relative 3d reconstruction error of all points in the last frame of the sequence. Given the true image coordinates  $\mathbf{x}_L$  and  $\mathbf{x}_R$  in both images and the estimated camera parameters  $\hat{\omega}_L$  and  $\hat{\omega}_R$ , we can determine the 3d position  $\hat{\mathbf{X}}$  of the corresponding object point using Hartley’s triangulation method [12]. The relative 3d reconstruction error is then computed as  $\epsilon_{rel} = \|\hat{\mathbf{X}} - \mathbf{X}\| / \|\mathbf{X}\|$ , where  $\mathbf{X}$  denotes the true 3d position.

Fig. 3a depicts the results of the proposed algorithm. The standard deviations of the pixel error varied from 0 to 1 pixels and 50 independent simulations were run on each noise level. We compared three different versions of the algorithms: a) using only the epipolar constraint, b) using only the trifocal constraint, and c) using recursive bundle adjustment only. We found that bundle adjustment gives best results, but as indicated above, is the most time consuming method.

The stereo calibration tracking was also tested on real imagery. Fig. 3b shows sample frames of the sequence with the extracted features. We have chosen a version of our algorithm combining at most 30 tracking features  $S_b$  in bundle adjustment and 30 stereo features  $S_e$  in the epipolar constraint. The vehicle was first driving straight for about 40 frames and then made a left turn. This is also reflected in the estimated motion parameters  $(\omega, \mathbf{V})$  (Fig. 4). The cameras were rotated twice in the sequence: first about  $15^\circ$  to the left before starting the turn at frame 38, second about  $-15^\circ$  after completing the turn (frame 168). Both changes in the gaze direction are captured by the self-calibration.

Since ground truth for this real imagery example was not available, we used stereo reconstruction results to assess the performance of our self-calibration.



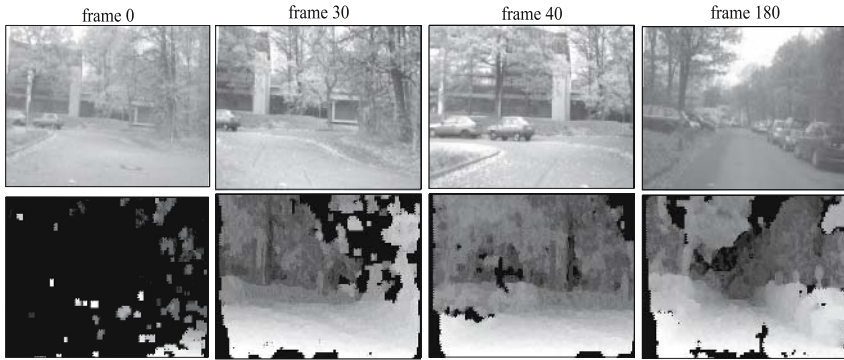
**Fig. 4.** Extracted 6d ego motion (a,b) and camera parameters (c,d,e) for the sequence depicted in Fig. 3b. The cameras were rotated twice:  $15^\circ$  to the left before starting the turn (frame 38) and then  $15^\circ$  to the right after leaving the curve in frame 162.

Stereo reconstruction was performed by first rectifying the images with the estimated camera parameters, so that all corresponding pixels in both rectified frames should have the same  $y$ -coordinate. Then, correlation based matching as described in [13] was performed. Please note that we fully relied on the stereo rectification and used only a 1d search region for stereo matching, so that erroneous camera parameters have great influence on the matching performance.

The left column of Fig. 5 displays the stereo reconstruction results using the initial stereo parameters. As the initial parameter setting was just a manual guess of the camera parameters, stereo reconstruction was not possible here. Valid disparity images are already obtained after two frames and at frame 30 — just before rotating the cameras to the left —, the self-calibration has converged to reliable camera parameters. The estimated stereo calibration even remains valid after the two camera rotations in frames 38 and 168 and gives satisfying results over the whole sequence.

## 5 Conclusion

This contribution presented the self-calibration of an active stereo rig based on three different criteria: bundle adjustment with reduced dimension of the parameter vector, epipolar constraint, and trilinear constraints. The optimization of the camera parameters is implemented as a robust Iterated Extended Kalman filter that minimizes physically relevant errors in the image plane. The synthetic examples in Sec. 4 reveal that bundle adjustment with reduced parameter vector outperforms the other constraints by nearly a factor of three in terms of accuracy. However, since the epipolar constraint is not deteriorated by independently moving objects, it may be beneficial in practical applications to combine two sets of bundle adjustment and epipolar constraint features to stabilize the self-calibration process. Such a combination was tested on a real imagery sequence.



**Fig. 5.** Stereo reconstruction results obtained with the stereo calibration from Fig. 4 (top: right camera frames, bottom: disparity images). Note that the initial calibration parameters at frame 0 are set manually and do not allow meaningful 3d reconstruction.

The proposed algorithm allowed both initial calibration refinement as well as a continuous update of the parameters of the active stereo system. We envision that our results may also contribute to the self-calibration of standard stereo rigs in a variety of applications. Future work will focus on the incorporation of lens distortion parameters into the camera parameter tracking.

## References

1. Gehrig, S.K.: Large-field-of-view stereo for automotive applications. In: *OmniVis 2005*, Beijing (2005)
2. Bjorkman, M., Eklundh, J.: Real-time epipolar geometry estimation of binocular stereo heads. *PAMI* **24**(3) (2002) 425–432
3. Petterson, N., Petersson, L.: Online stereo calibration using FPGAs. In: *IEEE Intelligent Vehicles Symposium*. (2005)
4. Hartley, R., Zisserman, A.: *Multiple View Geometry in Computer Vision*. Cambridge University Press (2002)
5. Förstner, W.: On weighting and choosing constraints for optimally reconstructing the geometry of image triplets. In: *ECCV*. Volume 2. (2000) 669–684
6. Longuet-Higgins, H.: A computer algorithm for reconstructing a scene from two projections. *Nature* **293** (1981) 133–135
7. Shashua, A.: Algebraic functions for recognition. *PAMI* **17**(8) (1995) 779–789
8. Hartley, R.: A linear method for reconstruction from lines and points. In: *International Conference on Computer Vision*. (1995) 885–887
9. Zhang, Z., Faugeras, O.: *3D Dynamic Scene Analysis*. Springer (1992)
10. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* **60**(2) (2004) 91–110
11. Dang, T., Hoffmann, C.: Stereo calibration in vehicles. In: *IEEE Intelligent Vehicles Symposium*, Parma, Italy (2004) 268–273
12. Hartley, R.I., Sturm, P.: Triangulation. *Computer Vision and Image Understanding* **68**(2) (1997) 146–157
13. Hirschmüller, H., Innocent, P.R., Garibaldi, J.M.: Real-time correlation-based stereo vision with reduced border errors. *IJCV* **47**(1-3) (2002) 229–246

# Handling Camera Movement Constraints in Reinforcement Learning Based Active Object Recognition

Christian Derichs\* and Heinrich Niemann

Chair for Pattern Recognition, Department of Computer Science, University  
Erlangen-Nürnberg, Martensstr. 3, 91058 Erlangen  
{derichs, niemann}@informatik.uni-erlangen.de

**Abstract.** In real world scenes, objects to be classified are usually not visible from every direction, since they are almost always positioned on some kind of opaque plane. When moving a camera selectively around those objects for classifying them in an active manner, a hemisphere is fully sufficient for positioning meaningful camera viewpoints. Based on this constraint, this paper addresses the problem of handling planned camera actions which nevertheless lead to viewpoints beyond the plane of that hemisphere. Those actions arise from the uncertainty in the current vertical camera position combined with the view planning method's request of a relative action. The latter is based on an optimized and interpolating query of a knowledge base which is built up in a Reinforcement Learning training phase beforehand.

This work discusses the influence of three different, intuitive and optimized, methods for handling invalid action suggestions generated by Reinforcement Learning. Influence is measured by the difference in classification results after each step of merging the image data information with active view planning.

**Keywords:** Active Vision, Viewpoint Selection, Reinforcement Learning.

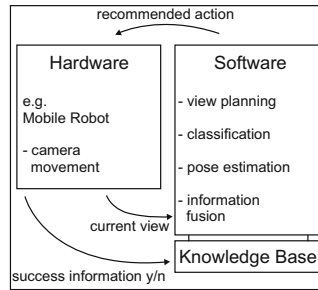
## 1 Introduction

The basic idea of active object recognition is the optimized selection of the viewpoints relative to an item in order to classify it reliably with a minimal amount of recorded sensor data, such as camera images. Naturally, this comes along with the necessity of moving a camera around the considered object and fusing the gathered information. Aspiring to optimality, some kind of camera movement planning is required, which in our approach is based on Reinforcement Learning [11]. We have already shown in various publications [4] [5] that this approach outperforms a random viewpoint selection when the camera movement is restricted to a circular path around the object.

The enhancement from the 1-dimensional path to the complete, 2-dimensional sphere around the item is quite simple at a first glance. But the restriction appearing in practice is that the majority of objects is positioned on an opaque plane, like a table or the floor. Consequently, cameras cannot be positioned in any way to take a meaningful

---

\* This work was funded by the German Science Foundation (DFG) under grant SFB 603/TP B2. Only the authors are responsible for the content.



**Fig. 1.** The elementary structure of the active view planning system

image from below those objects. Keeping this in mind, it is a reasonable approach to initially model the object only with image information gained from camera viewpoints that are arranged on a hemisphere around this object. In contrast to this modeling, later real world camera movements, as well as their horizontal and vertical positions relative to the considered object, are almost always afflicted with uncertainty. In particular, a camera controlling mobile robot may encounter an object worth classifying at some time during its so far undirected movement in an arbitrary environment. So regarding the relation between camera and object position, no initial alignment of coordinate systems can be done. Consequently, incoming information is limited to the recorded image of the object. Based on this information, a probabilistic suggestion about the relative camera position and the object class can then be established and an optimal next viewpoint for recognition can be calculated. Approaching the latter mostly means moving the whole robot as well as changing the camera angle, both being afflicted with inaccuracy.

Now the problem regarding the hemisphere constraint results from the fact that the software part (see figure 1) might randomly choose or even plan a relative action which actually would not result in a position on the mentioned hemisphere. Unfortunately, neither the hardware front end is not able to detect such an invalid action without actually performing it. Since the movement is at least partially performed when eventually recognizing the impracticability of the proposed action, a replanning would waste one step in the recognition process, which is considered worst-case in optimal, active view planning. Therefore, the hardware in particular must not reject or request an action from the software part, but has to promptly handle the given action instruction somehow. Considering the other direction, in order to keep the approach universal the software is assumed to not explicitly get to know anything about the success of the real action, it merely obtains the next image. In our framework, success information is just used for rating executed actions (see chapter 2).

Thus, the problem under consideration is how to immediately deal with requested actions that are invalid in a sense of camera movement constraints and how to adapt this to the knowledge representation, in fact without raising the planning effort disproportionately. The latter is exactly the point most of the related work is missing. For example, [9] explicitly puts a lot of effort into the exact recovery of conditions obtained before performing a failed action in order to try another action then. [10] proposes an approach which emerges from robot navigation and modifies a planned, demanded action or action sequence if this did not lead to a recognizable state enhancement in the preceding time steps. Of course, in our active view planning system we cannot wait for the system



to recognize such a dead end. Other work from the area of *Replanning* displaces the error handling work towards the training phase, like [2], which alternately inhibits action components during training in order to directly learn backup plans. Similarly, [13] introduces a *plan transformation* that varies a proposed action to similar one which is most probably valid. But this transformation advice is again based on an expanded training phase including the buildup of a *plan library*. Thus, those approaches are worth considering if training complexity is not a matter. A completely different, Reinforcement Learning adjusted idea is suggested by [6], which incorporates the risk of a valid action to become invalid due to some inaccuracy into an action's rating. A similar approach and its common disadvantages are discussed in chapter 3.

Chapter 2 gives an outline of the basic methods of Reinforcement Learning and explains the role of the problem specific variables within this framework. Afterwards, chapter 3 first introduces the probabilistic description of the class and pose assumptions of an object via a set of particles and then shows three methods for propagating these particles according to the action handling direction under consideration. Chapter 4 compares the proposed methods regarding their impact on experimental classification results.

## 2 Reinforcement Learning

### 2.1 Basic Principles

In a basic Reinforcement Learning approach, the three decisive items are states  $s_t$ , actions  $\mathbf{a}_t$  that lead to new positions and rewards  $r_t$  that rate the performed action  $\mathbf{a}_t$  given state  $s_t$  as its starting point. A timestamp  $t$  clarifies that in the assumed environment we have multiple episodes of  $T$  temporally successive actions  $\langle \mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_T \rangle$  and resulting states  $\langle s_1, s_2, \dots, s_{T+1} \rangle$  each.

For our purpose:

- **States** within the development environment are multi-modal probability distributions (see [4] [5]). They are discrete concerning the number of object classes  $\Omega_\kappa$  and continuous within the values for horizontal ( $\Phi_h$ ) and vertical ( $\Phi_v$ ) camera positions relative to the object. Consequently, they contain probabilistic assumptions about the class and pose of an object under consideration. Those assumptions arise from the information extracted from the image data  $\mathbf{f}_t$  the camera acquires in each time step. Thus, the state densities can be presented by

$$s_t = p(\mathbf{q}(t) | \langle \mathbf{f} \rangle_t, \langle \mathbf{a} \rangle_{t-1}) \quad \text{with} \quad \mathbf{q} = (\Omega_\kappa, \Phi_h, \Phi_v)^T. \quad (1)$$

Here,  $\langle \mathbf{f} \rangle_t = \mathbf{f}_t, \mathbf{f}_{t-1}, \dots, \mathbf{f}_1$  and  $\langle \mathbf{a} \rangle_{t-1} = \mathbf{a}_{t-1}, \dots, \mathbf{a}_1$  indicate the fusion of all image information and camera movement knowledge gathered during a whole episode. Therefore, the probability distribution  $s_t$  itself is a fused product of information gathered in multiple time steps. For a detailed explanation of the fusion procedure refer to [3].

- **Actions** are the movements of the camera in-between the taking of two consecutive images. For the purpose of this work, actions  $\mathbf{a}_t = (a_{h,t}, a_{v,t})^T$  are limited to two components, the horizontal ( $a_h$ ) and the vertical ( $a_v$ ) movement of the camera fixed on a predefined hemisphere around the object.

- **Rewards** follow the usual definition in Reinforcement Learning

$$R_t = \sum_{n=0}^{\infty} \gamma^n r_{t+n} \quad \text{with} \quad \gamma \in [0; 1], \quad (2)$$

where  $r_t$  is the immediate reward when analyzing a next state  $s_t$  and  $\gamma$  is a weighting factor whose influence is increasingly reduced with a growing step indicator  $n$ . Summing up the sequentially appearing rewards  $r_t$  results in a forward-looking reward, called return  $R$ , when at time step  $t$  within an episode. In practice, episode lengths are finite and summation in (2) is aborted accordingly. Since rewards are calculated from the resulting state representation  $s_t$ , the choice of a significant property of those densities is the crucial task in Reinforcement Learning. So we decided for high rewards when the most probable class  $\Omega_{\kappa}$  out of  $k$  classes has high confidence, according to

$$r_t = \max_i \int_{\Phi_h} \int_{\Phi_v} p(\mathbf{q}^i(t+1) | \langle f \rangle_{t+1}, \langle \mathbf{a} \rangle_t) d\mathbf{q}^i \quad \text{with} \quad \mathbf{q}^i = (\Omega_{\kappa=i}, \Phi_h, \Phi_v)^T. \tag{3}$$

Consequently, rewards in this approach obey the relation  $k^{-1} \leq r_t \leq 1$ .

### 2.2 Calculation of the Behavior Policy

Now given a rewarding rule, we can build up a knowledge base during training containing action-value functions  $Q$  which represent the quality of an action  $\mathbf{a}$  in state  $s$  depending on the return's expectation value:

$$Q(s, \mathbf{a}) = E \{ R_t | s_t = s, \mathbf{a}_t = \mathbf{a} \} \tag{4}$$

Of course, during training we acquire a fully calculated return  $R_t$  since we perform the whole episode before expanding the knowledge base. But at runtime in every time step  $t$  we can only make assumptions about the future behavior, so an expectation value is the appropriate formulation. In general, we cannot expect to get only such states  $s$  during evaluation that we have already seen in the training phase, since we work in a continuous environment. To nevertheless be able to declare a best action in every situation we acquire, an approximation term  $\widehat{Q}(s, \mathbf{a})$  is necessary:

$$\widehat{Q}(s, \mathbf{a}) = \frac{\sum_{(s', \mathbf{a}')} K(d(\theta(s, \mathbf{a}), \theta(s', \mathbf{a}'))) Q(s', \mathbf{a}')}{\sum_{(s', \mathbf{a}')} K(d(\theta(s, \mathbf{a}), \theta(s', \mathbf{a}')))} \tag{5}$$

- $(s', \mathbf{a}')$  are the state-action pairs already stored in the knowledge base.
- $\theta(s, \mathbf{a})$  is the resulting multi-modal density function when transforming  $s$  according to an action  $\mathbf{a}$ . The various rules for this transformation are the gist of this paper and will be extensively discussed in chapter 3.
- $d$  calculates the distance between two density functions using the extended Kullback-Leibler distance
- $K(x) = \exp(-x^2/D^2)$  is a Gaussian kernel for weighting those distances  $d$ . The free kernel parameter  $D$  determines the smoothness or rather the local fineness of the approximation in (5).

Readdressing the topic of this work, the approximative character of  $\widehat{Q}(s, \mathbf{a})$  is—next to the movement uncertainty—the main reason for obtaining invalid actions during runtime at all. The final step for finding the best action in the current state is an optimized global Adaptive Random Search [12] over all actions in question, followed by a local Simplex. Consequently, we obtain the optimal considered action  $\check{\mathbf{a}}$ :

$$\check{\mathbf{a}} = \underset{\mathbf{a}}{\operatorname{argmax}} \widehat{Q}(s, \mathbf{a}). \tag{6}$$

### 3 Handling Critical Actions

To understand the complete process when invalid actions occur, we first give a more precise insight into the structure of the multi-modal densities representing the states  $s_t$ . Note that in object recognition tasks, pose and class probability functions are generally not normally distributed. So unfortunately we cannot make use of the well known Kalman Filter [8] for the necessary density propagation. Instead, the proposed method applies a Particle Filter [1] to that problem, resulting in a probability density that is represented by a set  $\Gamma_t = \{\rho_t^1, \dots, \rho_t^N\}$  of  $N$  single particles  $\rho_t$ . Each of those particles  $\rho = \{\tilde{\Omega}_\kappa, \tilde{\Phi}_h, \tilde{\Phi}_v, \omega\}$  contains information about the class  $\tilde{\Omega}_\kappa$ , the horizontal pose ( $\tilde{\Phi}_h$ ) and the vertical pose ( $\tilde{\Phi}_v$ ) it is representing. Additionally it holds an entry for its own weighting  $\omega$ . Density propagation is then easily accomplished via the Condensation algorithm which is directly aligned to those particle representations. For details about this method refer to [7].

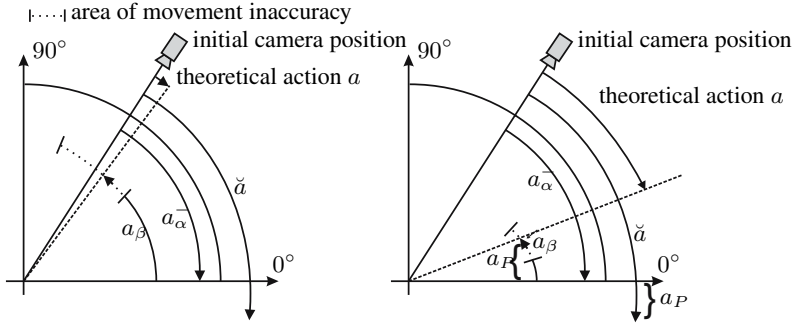
The purpose of this work is to provide and compare meaningful instructions for camera movements based on invalid movement demands, i.e. when they would exceed either the north pole or the hemisphere's plane in the vertical direction. Note that the north pole was additionally addressed as a critical edge in order to provide some symmetry to the planning task. Besides, this assumption should prevent problems when integrating actions' costs into the learning process, which is outside the scope of this paper.

Finally, the movement adaption strategies  $\mathcal{H}$  are :

- 1.) **Pseudo-Persistence  $\mathcal{P}_F$  with Penalization  $F$**
- 2.) **Movement up to the Critical Edge and Stopping  $S$**
- 3.) **Edge-Reflected Movement  $\mathcal{R}$**

To keep things concise, we reduce the following examinations to the critical, vertical action  $a$  and state component  $\Phi$ , respectively. Camera movements are called valid if  $\Phi \in \mathcal{T} = [0^\circ; 90^\circ]$  for the resulting vertical camera position  $\Phi$ . Actions ending up at  $\Phi = 90^\circ$  are named  $a^+$ , those leading to  $\Phi = 0^\circ$  are called  $a^-$ . Please note that we can exactly determine having reached  $0^\circ$  or  $90^\circ$  via the camera mechanics whereas all other position specifications are just probabilistic. This is due to the inaccuracy of relative actions which have to be performed by the hardware, e.g. a mobile robot. In particular, we will show the demand of repeating a performed action  $a$  which accordingly cannot be done exactly. Thus the repeated action is symbolized by  $\tilde{a}$ .

1.) The first approach is quite intuitive and obeys mainly the idea of Reinforcement Learning. Here, the actual proceeding concerning the camera movement is to return to the starting point if a required vertical action  $\tilde{a}$  turns out to be inexecutable *during motion*. Otherwise the action is simply performed. Considering the former, that means we can divide the whole movement process at this time step into two sub-movements indicated by subscripts  $\alpha$  and  $\beta$ . So, first an action  $a_\alpha = a_\alpha^{+/-}$  physically takes the camera to a critical edge. Then we need to reverse this action in order to go back to the starting point. Regarding the action inaccuracies the rule for this is  $a_\beta = -\tilde{a}_\alpha^{+/-}$  (see the left drawing of figure 2). Please remember that we cannot just fix the camera in the starting point for this time step since we don't know about the possible success of a movement before actually trying it (see section 1). Since then the camera position would only change relative to the action inaccuracy, this behavior is called **pseudo-persistence**. To understand the particle handling correctly, remember that the software part in figure 1



**Fig. 2.** Exemplary camera movement handling for invalid actions using the pseudo-persistence (left) or edge-reflection method (right). The inaccuracy within the second sub-action  $a_\beta$  is chosen exemplary.

is not aware of any information about the hardware’s action operability. So each particle of the current state density is moved in a translative manner just according to its represented hypothesis and the proposed action  $\check{a}$ . To point out the resulting similarities and differences between propagating the particles in the state representation and the physical action handling, invalid actions (combinations of sub-movements) are mapped to the one *valid*, but still theoretical action  $a$  which would result in the corresponding final camera position:

$$a = \begin{cases} a_\alpha^+ - \check{a}_\alpha^+ & \text{if } \check{a} > a_\alpha^+ \\ a_\alpha^- - \check{a}_\alpha^- & \text{if } \check{a} < a_\alpha^- \\ \check{a} & \text{else} \end{cases} ; \quad \tilde{\Phi}(\rho_{t+1}) = \begin{cases} \tilde{\Phi}(\rho_t) & \text{if } \tilde{\Phi}(\rho_t) + \check{a} \notin \mathcal{Y} \\ \tilde{\Phi}(\rho_t) + \check{a} & \text{else} \end{cases} \quad (7)$$

Above all, that implies that huge amounts of particles might change position even if only pseudo-persistence movement is physically performed.

Knowing that an illegal move leads to a next view from almost the same position, which in general adds very little new information to the classification task, it appears to be a rational and Reinforcement Learning consistent idea to punish those actions during training. So, if using the pseudo-persistence method  $\mathcal{P}_F$  and applying the punishment  $F$  to the reward function ( $r_t = F$  in (3)), it should be possible to learn to avoid those illegal actions during the following evaluation phase. However, that is where the problem occurs with this approach. Since we have no a-priori knowledge about the domain of appearing rewards for valid moves in general it is impossible to establish an optimal  $F$  beforehand. So either the determination of adequate values for such a punishment has to be integrated into a far more complex and time consuming learning process [6] or it has to be set generously high by hand. The latter is obviously suboptimal since also legal movements into the border area would be partially disadvantaged because of the approximation mentioned in (5). This means that optimal next best viewpoints might be ignored merely because of their proximity to the illegal area, which is a quite well known problem in Reinforcement Learning. Even originally valid actions within an episode during training could be negatively affected if an invalid action follows later on in the same episode, regarding the return (2) with  $\gamma > 0$ . Nevertheless, this is the approach a problem-unspecific Reinforcement Learning method would apply. Thus, chapter 4 will also show the classification results when applying various punishment terms  $F$ .

2.) The second approach tackles the afore mentioned problem of unadjusted punishment terms by avoiding their occurrence in general. This is simply done by **stopping a critical camera movement** if either the north pole or the hemisphere's plane is reached, thus  $a_\beta = 0$ . This way, non-executable actions cannot be avoided either, but each of them can again be uniquely mapped to a valid action and can be rated by the usual Reinforcement Learning reward (3). Again, density particles have to be handled with the same procedure, yielding the following directions:

$$a = \begin{cases} a_\alpha^+ & \text{if } \check{\alpha} > a_\alpha^+ \\ a_\alpha^- & \text{if } \check{\alpha} < a_\alpha^- \\ \check{\alpha} & \text{else} \end{cases} ; \quad \tilde{\Phi}(\rho_{t+1}) = \begin{cases} 90^\circ & \text{if } \tilde{\Phi}(\rho_t) + \check{\alpha} > 90^\circ \\ 0^\circ & \text{if } \tilde{\Phi}(\rho_t) + \check{\alpha} < 0^\circ \\ \tilde{\Phi}(\rho_t) + \check{\alpha} & \text{else} \end{cases} \quad (8)$$

Please note that stopping at the critical edges differs from most other, possibly randomly chosen successive actions in the way that it is deterministic. In particular, each particle can be propagated in exactly the same manner a camera would move when being located at this particle's parameter hypotheses - in fact without knowing the real outcome of the camera movement beforehand.

3.) The obvious drawback of the previous approach is the preference of edges as arrival points for the camera movement. Since we always permit relative vertical actions within the range of  $[-90^\circ; 90^\circ[$ , theoretically every second move ends up at one of those two edges during the Reinforcement Learning training phase, since we only perform random actions here. Intuitively, this might result in a heavily unbalanced knowledge base built during this training.

In order to overcome this potential barrier as well, our proposed idea for an optimized action handling is the **edge-reflected** movement and particle propagation, respectively. As the name says, a *remaining* action potential  $a_P$  is continued in the reverse direction whenever reaching a critical edge at runtime, thus  $|a_\beta| = |\check{\alpha}| - |a_\alpha|$ . The corresponding illustration can be found in the right drawing of figure 2. This way, each valid movement has exactly two related movements it can emerge from, the originally valid and the reflected one. In contrast to the edge-stopping method this relation is one-to-one now. This results in a uniform probability distribution for reaching any viewpoint in the next time step. Additionally, the deterministic behavior is assured once more and the calculation instructions can be expressed as:

$$a = \begin{cases} a_\alpha^+ - (\check{\alpha} - \tilde{a}_\alpha^+) & \text{if } \check{\alpha} > a_\alpha^+ \\ a_\alpha^- + (\tilde{a}_\alpha^- - \check{\alpha}) & \text{if } \check{\alpha} < a_\alpha^- \\ \check{\alpha} & \text{else} \end{cases} ;$$

$$\tilde{\Phi}(\rho_{t+1}) = \begin{cases} 180^\circ - \tilde{\Phi}(\rho_t) - \check{\alpha} & \text{if } \tilde{\Phi}(\rho_t) + \check{\alpha} > 90^\circ \\ -\tilde{\Phi}(\rho_t) - \check{\alpha} & \text{if } \tilde{\Phi}(\rho_t) + \check{\alpha} < 0^\circ \\ \tilde{\Phi}(\rho_t) + \check{\alpha} & \text{else} \end{cases} . \quad (9)$$

## 4 Experimental Results

For experimental evaluation of our active object recognition task, we have chosen four classes of real toy figures, discriminable by a quiver and a lamp. Corresponding to the 2-dimensional approach of this paper, figure 3 shows some examples of images that can

be acquired from viewpoints situated on the hemisphere around the object. Please note that every recorded image was superposed by a uniform Gaussian noise before processing it any further. This way overall classification results are equally downgraded, but differences in the various applied methods for action handling will be more distinguishable. For classification itself image features are extracted by a Principal Component Analysis using only the ten best eigenvectors in the transformation matrix.

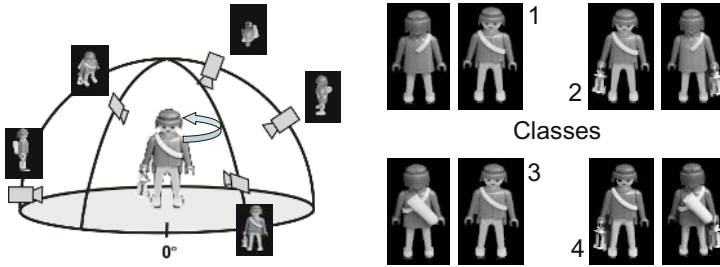
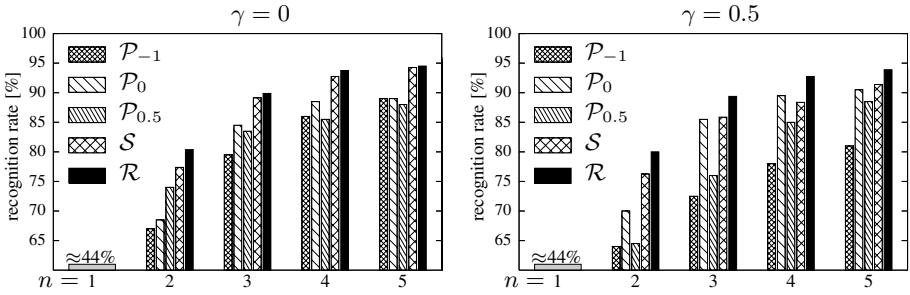


Fig. 3. Views of the toy object classes

Reinforcement Learning training was done with 50 episodes per class, each containing eight camera actions in between nine recorded images, whereas all actions were performed completely randomly. These training stages were separately executed for every combination of action handling  $\mathcal{H} \in \{P_{-1}, P_0, P_{0.5}, S, R\}$  and weighting  $\gamma \in \{0, 0.5\}$  introduced in (2). During the evaluation phase, 50 exploiting episodes were performed for each class based on the particular knowledge base built up during training. The kernel parameter  $D$ , which varies  $K(x)$  in (5), was set to a well-proven value of  $D = 10$  (see [5]). Figure 4 shows the classification results we achieved with the various combinations after the fusion of the information data of  $n$  images. Results are displayed up to a step width of  $n = 5$  since later results just converge to a saturation and thus would just reduce clarity. In any case, the most important values are those of early steps ( $n = 2, 3$ ) as they show the immediate gain or loss in classification certainty most real decisions would rely on. Concerning this, it is obvious that the proposed new methods of  $\mathcal{H} = S, \mathcal{R}$  clearly outperform those based on a punishment for invalid camera actions. Nevertheless, especially the right chart of figure 4 points out that there are indeed penalization terms (like  $F = 0$ ) that can achieve a comparably high learning quality. But as mentioned, this comes at a price of having a priori knowledge about the range of the regular Reinforcement Learning rewards, and then it still needs some experience and object specific previous knowledge to optimize the punishment value. Those preconditions cease to apply when using the improved edge-reflecting or edge-stopping method.

Since results for  $\mathcal{H} = S$  are quite close to those of  $\mathcal{H} = \mathcal{R}$  regarding the *recognition rates*, we should additionally pay attention to the *pose estimation accuracy* which might turn the balance then. To evaluate this, in figure 5 we additionally depicted the localization error of the vertical object pose after  $n$  steps of image fusion. Therefore, we concentrated on  $\mathcal{H} = S, \mathcal{R}$  and again distinguished between  $\gamma = 0$  and  $\gamma = 0.5$ . These results explicitly show that, for pose estimation accuracy as well, the edge-reflecting method is the one to prefer. As with the classification results, we achieve obvious enhancements mainly within the early steps. Obviously, when using  $\mathcal{H} = S$ , the accumulation of particles at one of the critical edges for invalid camera movement demands

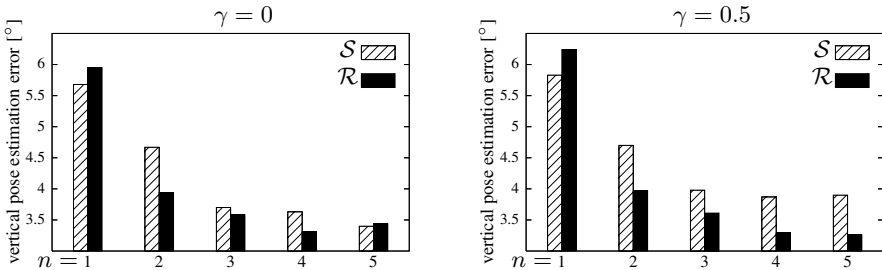


**Fig. 4.** Recognition rate [%] after  $n$  planned actions and information fusion. The various columns display the influence of the compared action handling methods  $\mathcal{H}$  for invalid action demands. The left figure ( $\gamma = 0$ ) considers one-step returns in the Reinforcement Learning, the right one ( $\gamma = 0.5$ ) represents ahead looking returns.

takes effect, resulting in an unbalanced distribution and thus a more imprecise pose representation. Regarding the weighting  $\gamma$  we can postulate that its value does not affect the general ranking of the three proposed action handling methods. Thus, it can be considered a noncritical parameter for the method selection decision.

### 5 Summary and Future Work

The focus of this work was on object recognition tasks with predefined constraints in valid camera positions and thus camera movements. Therefore, we emphasized the problem of providing an immediate alternative camera movement direction when the requested one turns out to be non-executable. For that purpose, we compared three different action handling approaches and their influence on the underlying state density representation. Summing up the visualized results, the edge-reflecting version for handling non-executable actions turned out to outperform the others in practice, according to our presumption. We pointed out the advantage of the edge-stopping and the edge-reflecting methods compared to other approaches introduced in the literature. In particular, the former explicitly support our demand of classification with a minimal number of



**Fig. 5.** Estimation error in the vertical object pose [ $^{\circ}$ ] after  $n$  planned actions and information fusion. Values are compared for the edge-stopping ( $\mathcal{H} = \mathcal{S}$ ) and the edge-reflecting ( $\mathcal{H} = \mathcal{R}$ ) action handling methods. Again, the left figure shows results for  $\gamma = 0$  and the right one for  $\gamma = 0.5$  in the Reinforcement Learning return function (2).

camera movements since they avoid replanning and thus discarding already performed movements.

Further work on this topic will concentrate on the assignment of costs to the various movement actions, bringing the proposed problem to a completely new dimension. One of the main questions is whether the applied action handling methods can address cost integration at all. In any case, adjustment should nevertheless be a complex procedure.

## References

1. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking. *IEEE Transactions of Signal Processing*, 50:174–188, 2002.
2. K.H. Chang and M. Edhala. Execution Error Recovery for Planning Systems. In *In Proceedings of the 7th Annual International Phoenix Conference on Computers and Communications*, pages 492–496, Scottsdale, USA, 1988.
3. F. Deinzer, J. Denzler, and H. Niemann. On Fusion of Multiple Views for Active Object Recognition. In *Pattern Recognition – 23rd DAGM Symposium*, pages 239–245, Munich, Germany, 2001.
4. F. Deinzer, J. Denzler, and H. Niemann. Viewpoint Selection - Planning Optimal Sequences of Views for Object Recognition. In *Computer Analysis of Images and Patterns - CAIP '03*, number 2756 in *Lecture Notes in Computer Science*, pages 65–73, Groningen, Netherlands, 2003.
5. F. Deinzer, Ch. Derichs, and H. Niemann. Aspects of optimal viewpoint selection and viewpoint fusion. In *Computer Vision - ACCV 2006*, volume 2, pages 902–912, Hyderabad, India, Januar 2006.
6. P. Geibel and F. Wysotzki. Risk-Sensitive Reinforcement Learning Applied to Control under Constraints. *Journal of Artificial Intelligence Research*, 24:81–108, 2005.
7. M. Isard and A. Blake. CONDENSATION — Conditional Density Propagation for Visual Tracking. *IJCV* 98, 29(1):5–28, 1998.
8. R.E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, pages 35–44, 1960.
9. C. A. Knoblock. Planning, Executing, Sensing, and Replanning for Information Gathering. In *In Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 1686–1693, Montreal, Canada, 1995.
10. A. Ranganathan and S. Koenig. A Reactive Robot Architecture with Planning on Demand. In *In Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, volume 2, pages 1462–1468, Las Vegas, USA, 2003.
11. R.S. Sutton and A.G. Barto. *Reinforcement Learning*. A Bradford Book, Cambridge, London, 1998.
12. A. Törn and A. Žilinskas. *Global Optimization*, volume 350 of *Lecture Notes in Computer Science*. Springer, Heidelberg, 1987.
13. R. van der Krogt, M. de Weerd, and C. Witteveen. A Resource Based Framework for Planning and Replanning. In *In Proceedings of the IEEE/WIC International Conference on Intelligent Agent Technology*, pages 173–186, Halifax, Canada, 2003.



# The Inversion Camera Model\*

Christian Perwass and Gerald Sommer

Institut für Informatik, CAU Kiel  
Christian-Albrechts-Platz 4, 24118 Kiel, Germany  
{chp, gs}@ks.informatik.uni-kiel.de

**Abstract.** In this paper a novel camera model, the *inversion camera model*, is introduced, which encompasses the standard pinhole camera model, an extension of the division model for lens distortion, and the model for catadioptric cameras with parabolic mirror. All these different camera types can be modeled by essentially varying two parameters. The feasibility of this camera model is presented in experiments where object pose, camera focal length and lens distortion are estimated simultaneously.

## 1 Introduction

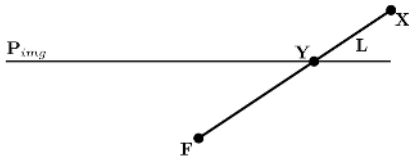
In a typical application utilizing wide angle lens cameras, the cameras' images have to be rectified before they can be used. Various lens distortion models have been suggested for this purpose, like the widely used polynomial model [5], the bicubic model [7], the rational model [1] or the division model [3]. Another type of imaging systems that are particularly useful for navigation applications are catadioptric cameras, since they allow a 360 degree view in a single image. Geyer and Daniilidis showed in [4] how such systems can be modeled quite easily mathematically.

In this paper a novel camera model, the *inversion camera model*, is introduced, which combines the pinhole camera model, a lens distortion model and a model for catadioptric cameras with parabolic mirrors. As is shown later on, the lens distortion model is just the division model introduced by Fitzgibbon in [3] and the catadioptric camera model has been first presented by Geyer and Daniilidis in [4]. However, the authors found that both models can be represented in much the same way using inversion in a sphere. This also extends the division model to lenses with an angular field of view (FOV) of 180 degrees or more.

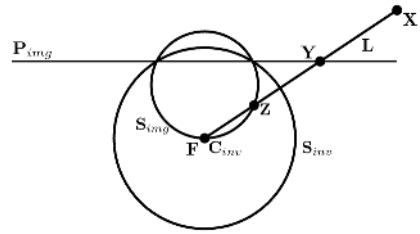
Inversion in a sphere can be represented as a (tri-)linear function in the Geometric Algebra of conformal space, which makes this algebra an ideal mathematical framework to work with the inversion camera model. The inversion camera model can be expressed as an algebraic entity of Geometric Algebra, i.e. a *multivector*, and a covariance matrix can be associated with it, which makes it directly applicable to statistical *linear* estimation methods as presented in [10,9]. This is demonstrated in section 3, where results of the simultaneous estimation of object pose, camera focal length and lens distortion are presented.

---

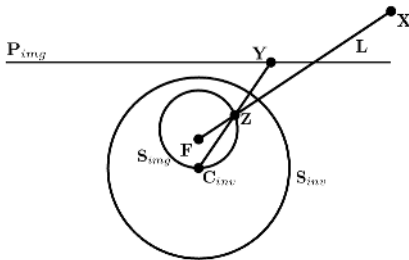
\* This work has been supported by DFG grant SO-320/2-3.



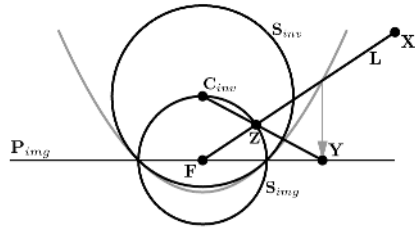
(a) Basic pinhole camera setup



(b) Pinhole camera setup with inversion



(c) Projective camera with lens distortion



(d) Parabolic catadioptric camera

**Fig. 1.** Different cameras representable by inversion camera model

A detailed understanding of Geometric Algebra is not necessary to follow the ideas presented in this paper. General introductions to Geometric Algebra can be found in [11,6,2]. Discussions of the application of Geometric Algebra to the estimation of geometric entities and operators, which are most closely related to this text, are [12,10,9].

The structure of this paper is as follows. First a general introduction to the inversion camera model is given, which is followed by a detailed discussion of the representation of lens distortion and parabolic mirror imaging systems. Finally, experiments on the simultaneous estimation of pose, focal length and lens distortion are presented to test the feasibility of the inversion camera model.

## 2 The Inversion Camera Model

The basic setup of the inversion camera model is shown schematically in figure 1 for the different imaging setups. Figure 1(a) shows the setup of the pinhole camera model. Point  $F$  is the focal point or optical center, point  $X$  is a world point and  $Y$  is the image of  $X$  on the image plane  $P_{img}$ . In a typical problem setup, the image point  $Y$  is given and the projection ray  $L$  has to be evaluated.

If the pinhole camera’s internal calibration is given, the projection ray  $L$  can immediately be evaluated in the camera’s coordinate frame.

In the inversion camera model this pinhole camera setup is represented as shown in figure 1(b). The sphere  $S_{inv}$  with center  $C_{inv}$  is used to perform an inversion of the image plane  $P_{img}$  which results in the sphere  $S_{img}$ . In particular, image point  $Y$  is mapped to point  $Z$ . In figure 1(b) the center  $C_{inv}$  of inversion sphere  $S_{inv}$  coincides with the focal point  $F$ . In this case the inversion of  $Y$  in  $S_{inv}$  results again in a point on the projection ray  $L$ , independent of the inversion sphere radius. Therefore, this setup is equivalent to the standard pinhole camera setup.

Figure 1(c) demonstrates what happens when the inversion sphere is moved below the focal point. Now the image point  $Y$  is mapped to  $Z$  under an inversion in  $S_{inv}$ . The corresponding projection ray  $L$  is constructed by  $F$  and  $Z$  and thus does not pass through  $Y$  anymore. It will be shown later on that this results in a lens distortion model similar to the division model proposed by Fitzgibbon [3].

Simply by moving the inversion sphere  $S_{inv}$  and the image plane  $P_{img}$ , catadioptric cameras with a parabolic mirror can be modeled. This construction is shown in figure 1(d), and is based on work by Geyer and Daniilidis [4]. An inversion of image point  $Y$  in sphere  $S_{inv}$  generates point  $Z$ . In this case, it is equivalent to an inverse stereographic projection of  $Y$  on the image sphere  $S_{img}$ , which is how this mapping is described in [4]. The corresponding projection ray  $L$  is again the line through  $F$  and  $Z$ .

The image  $Y$  of a world point  $X$  generated in this way is equivalent to the image generated by a parabolic mirror whose focal point lies in  $F$ , as is shown in [4]. That is, a light ray emitted from point  $X$  that would pass through the focal point  $F$  of the parabolic mirror, is reflected down parallel to the central axis of the parabolic mirror. This is also indicated in figure 1(d). The reflected light ray intersects the image plane  $P_{img}$  exactly in the point  $Y$ .

While the construction for the parabolic mirror in terms of a stereographic projection has been known for some while, the authors recognized that the stereographic projection can be replaced by an inversion, which makes this model readily representable in the Geometric Algebra of conformal space (CGA). In the following the mathematical details of the inversion camera model will be discussed.

**Mathematical Formulation.** In all calculations that follow, a right handed coordinate system is assumed, whereby  $e_1$  points towards the right along the horizontal image plane direction,  $e_2$  points upwards along the vertical image plane direction and  $e_3$  points from the image plane center towards the focal point or optical center. This implies that objects that are in front of the camera will have a negative  $e_3$  coordinate.

The geometric setup of the inversion camera model as presented in the previous section, can be modeled algebraically in CGA as follows. Like all transformations in Geometric Algebra, the image point transformation in the inversion camera model will be represented by a versor  $K$ . That is, if  $Y$  represents an image point, then  $Z := KY\widetilde{K}$  is the transformed image point. As can be seen in figure 1 the point  $Z$  will in general not lie on the image plane. However, the

goal is to find a  $\mathbf{K}$  such that  $\mathbf{Z}$  lies on the 'correct' projection ray. The transformed image point in the image plane can then be estimated by intersecting the projection ray with the image plane.

One of the simplest forms  $\mathbf{K}$  can take on is

$$\mathbf{K} = \mathbf{T}_s \mathbf{S} \widetilde{\mathbf{T}}_s \mathbf{D}, \tag{1}$$

where  $\mathbf{S}$  is a sphere centered on the origin,  $\mathbf{T}_s$  is a translator (translation operator) and  $\mathbf{D}$  a dilator (isotropic scaling operator). This form was also found to behave well numerically. The dilator scales the image, which has the same effect as varying the focal length, if the inversion sphere  $\mathbf{S}_{inv} := \mathbf{T}_s \mathbf{S} \widetilde{\mathbf{T}}_s$  is centered on the focal point (cf. figure 1(b)). If the inversion sphere is not centered on the focal point, the dilator also influences the distortion. In the following, the transformation  $\mathbf{K} \mathbf{Y} \widetilde{\mathbf{K}}$  is analyzed in some detail.

To simplify matters, it is assumed that  $\mathbf{T}_s$  translates the inversion sphere only along the  $\mathbf{e}_3$  axis. Furthermore,  $\mathbf{S}$  is a sphere of radius  $r$  centered on the origin. This is expressed in CGA as  $\mathbf{S} := \mathbf{e}_o - \frac{1}{2} r^2 \mathbf{e}_\infty$  and  $\mathbf{T}_s := 1 - \frac{1}{2} \tau_s \mathbf{e}_3 \mathbf{e}_\infty$ . It may then be shown that  $\mathbf{S}_{inv} = \mathbf{T}_s \mathbf{S} \widetilde{\mathbf{T}}_s = s_1 \mathbf{e}_3 + \frac{1}{2} s_2 \mathbf{e}_\infty + \mathbf{e}_o$ , with  $s_1 := \tau_s$  and  $s_2 := \tau_s^2 - r^2$ . The inversion sphere  $\mathbf{S}_{inv}$  can thus be regarded as a vector with two free parameters, that influence the sphere's position along  $\mathbf{e}_3$  and its radius.

The dilation operator  $\mathbf{D}$  for a scaling by a factor  $d \in \mathbb{R}$  is given by  $\mathbf{D} = 1 + \frac{1-d}{1+d} \mathbf{E}$ , where  $\mathbf{E} := \mathbf{e}_\infty \wedge \mathbf{e}_o$ . For brevity we define  $\tau_d := -\frac{1-d}{1+d}$ , such that  $\mathbf{D} = 1 - \tau_d \mathbf{E}$ . The image point transformation operator  $\mathbf{K}$  is then given by

$$\mathbf{K} = \mathbf{S}_{inv} \mathbf{D} = k_1 \mathbf{e}_3 + k_2 \mathbf{e}_\infty + k_3 \mathbf{e}_o + k_4 \mathbf{e}_3 \mathbf{E}, \tag{2}$$

with  $k_1 := s_1$ ,  $k_2 := \frac{1}{2} s_2 (1 - \tau_d)$ ,  $k_3 := 1 + \tau_d$  and  $k_4 := -\tau_d s_1$ .

In the model setup, the image plane  $\mathbf{P}_{img}$  passes through the origin and is perpendicular to  $\mathbf{e}_3$ . That is, image points lie in the  $\mathbf{e}_1 - \mathbf{e}_2$ -plane. An image point will be denoted in Euclidean space by  $\mathbf{y} \in \mathbb{R}^3$  and its embedding in conformal space by  $\mathbf{Y} := \mathcal{C}(\mathbf{y}) \in \mathbb{G}_{4,1}$ , that is  $\mathbf{Y} = \mathbf{y} + \frac{1}{2} \mathbf{y}^2 \mathbf{e}_\infty + \mathbf{e}_o$ . The embedded image point  $\mathbf{Y}$  is then mapped to the point  $\mathbf{Z}$  on the image sphere  $\mathbf{S}_{img}$ , via  $\mathbf{Z} = \mathbf{K} \mathbf{Y} \widetilde{\mathbf{K}}$ . Intersecting the line through the focal point  $\mathbf{F}$  and the transformed point  $\mathbf{Z}$  with the image plane gives the respective undistorted Euclidean image point  $\mathbf{y}_d \in \mathbb{R}^3$ . From a straight forward, if tedious calculation, it follows that

$$\mathbf{y}_d = \frac{-(s_1^2 - s_2) d}{s_1 (s_2 - s_1) + (s_1 - 1) d^2} \mathbf{y} = \frac{\beta}{1 + \alpha \mathbf{y}^2} \mathbf{y}, \tag{3}$$

where  $\alpha := \frac{(s_1 - 1) d^2}{s_1 (s_2 - s_1)}$  and  $\beta := \frac{-(s_1^2 - s_2) d}{s_1 (s_2 - s_1)}$ . Note that  $\mathbf{y}_d / \beta$  is the division model as proposed by Fitzgibbon in [3].

Typically, lens distortion models are used to remove the distortion in an image independent of the focal length or angular field of view (FOV) of the imaging system that generated the image. This is usually done by either requiring that lines which appear curved in the image have to be straight, or by enforcing multi-view constraints given a number of images of the same scene. The rectified

image can then be used for any other type of application. For this purpose and for lenses with a FOV of at most  $180^\circ$ , the inversion model is equivalent to the division model.

However, here the applicability of the inversion model as a *camera model* is investigated. That is to say, the lens distortion of a camera system is modeled directly in the context of a constraint equation. This is shown in section 3 in the context of monocular pose estimation.

**Focal Length and Lens Distortion Relationship.** The distortion generated by the inversion model as given in equation (3), has the effect that focal length and distortion are not independent, since  $\alpha$  and  $\beta$  are not independent. The factor  $\beta$  mainly represents an overall scaling of the image, while  $\alpha$  mainly influences the distortion. The exact relationship will be discussed in the following.

First of all, note that the interrelation of  $\alpha$  and  $\beta$  does not represent a drawback as compared to the division model, if the image rectification is done independently and previous to any other calculations, as pose estimation. However, if the inversion camera model is used directly in a constraint equation as in equation (5) in section 3, then not every level of distortion can be rectified for every focal length or field of view (FOV).

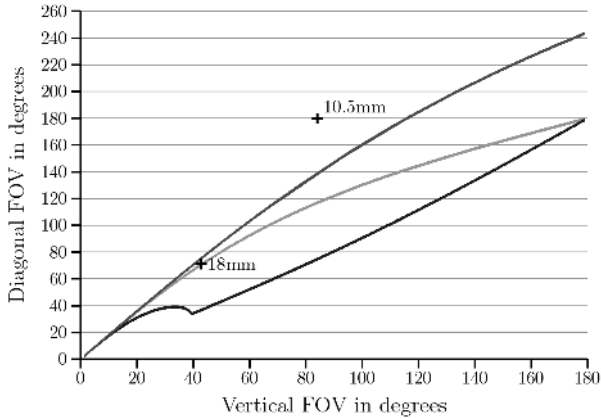
The relationship between the transformed image point  $\mathbf{y}_d$  and the initial image point  $\mathbf{y}$  is given by the factor  $\omega := \frac{\beta}{1+\alpha \mathbf{y}^2}$ , such that  $\mathbf{y}_d = \omega \mathbf{y}$ . The factor  $\omega$  is therefore a function of the squared radial distance  $\mathbf{y}^2$  of an image point from the image center. The locations of constant  $\omega$  in an image thus form concentric circles about the image center. These circles will be called *iso-circles* in the following. An iso-circle of particular interest in the analysis is the one that touches the upper and lower borders of the image, i.e. its radius is equal to half the vertical extent of the image. This particular iso-circle will be called *vertical iso-circle* and its radius will be denoted by  $\rho_v$ .

The value of  $\omega$  for image points on the vertical iso-circle is directly related to the vertical angular field of view (vFOV). Note that the relation to the focal length is more complex if lens distortion is present, since the focal length is now a function of  $\mathbf{y}^2$ . That is, the focal length depends on the position of an image point in the image. It is therefore more useful to define an *effective* focal length (EFL) as the focal length of the image points on the vertical iso-circle.

The value of  $\omega$  for image points on the vertical iso-circle will be denoted by  $\omega_v$  and is given by  $\omega_v = \frac{\beta}{1+\alpha \rho_v^2}$ . The Euclidean position vector  $\mathbf{f}$  of the focal point is in the following parameterized as  $\mathbf{f} = \tau_f \mathbf{e}_3$ . That is, if the image is neither scaled nor distorted,  $\tau_f$  is the focal length. It may be shown that the EFL  $\mathbf{f}_e$  is related to  $\omega_v$  by  $\mathbf{f}_e = \tau_f / \omega_v$ .

It is possible to vary the inversion sphere center  $\tau_s$  and the image scaling  $d$  such that the diagonal angular field of view (dFOV), i.e. the image distortion, is varied, while  $\mathbf{f}_e$  and thus the vFOV are kept constant. This relationship is shown in figure 2.

Here  $\tau_f = 1$ ,  $r = 0.5$  and the image plane size was assumed to be  $23.7 \times 15.6\text{mm}$ , which is the CCD-chip size of a D70 digital SLR camera. The middle, green line shows the relation between the vFOV and the *diagonal* angular field



**Fig. 2.** Vertical vs. diagonal field of view (FOV) for pinhole model (middle, green graph), maximum trapezoidal (top, red graph) and maximum cushion (bottom, blue graph) distortion. Inversion sphere radius is 0.5.

of view (dFOV) for a standard pinhole setup. The top, red line gives the relation for maximum trapezoidal distortion and the bottom, blue line for the maximum cushion distortion. It was found that the maximum dFOV does not depend on  $\tau_f$  or  $r$ . The location of the kink in the minimum dFOV plot does depend on the combination of  $\tau_f$  and  $r$ , though.

To check, whether the inversion camera model can model actual lenses, the vFOV and dFOV of two lenses were measured and plotted. The first was the zoom lens SIGMA DC 18-125mm, 1:3.5-5.6 D, set to 18mm. This lens lies in the achievable distortion range of the inversion camera model.

The second lens was the Nikkor AF Fisheye 10.5mm, 1:2.8 G ED. This lens is a corrected fisheye, whereby the image does not appear circular but fills the whole image. This is achieved by obtaining a 180° FOV only along the diagonal and compressing the image more along the vertical than the horizontal. As can be seen in figure 2 the 10.5mm lens cannot correctly be represented by the inversion camera model. This is due to the different type of projection of fisheye lenses, which cannot be modeled by the inversion camera model. In the pose estimation experiments presented in section 3, it turns out that the inversion camera model approximates the 10.5mm lens well enough, though, to achieve good pose estimation results.

It is important to note that the above analysis is only an indicator whether a lens may be representable in the inversion camera model, since the actual lens distortion will in general be a more complex function. However, it was already shown in [3] that the division model, which is equivalent to the inversion model in the case of lens distortion, is a sufficiently good approximation of lens distortion for many applications.

**Catadioptric Camera with Parabolic Mirror.** With respect to figure 1(d), the generation of image point  $\mathbf{Y}$  from world point  $\mathbf{X}$  via reflection at a parabolic

mirror, can be represented mathematically by projecting  $\mathbf{X}$  onto the sphere  $\mathbf{S}_{img}$  followed by an inversion in the sphere  $\mathbf{S}_{inv}$ . In contrast to the inversion model setup for lens distortion, the focal point  $\mathbf{F}$  lies on the image plane in this case.

The relation of the physical parabolic mirror with respect to the mathematical setup is indicated in figure 1(d). In a standard setup the sphere  $\mathbf{S}_{img}$  has unit radius and is centered on the focal point  $\mathbf{F}$  of the parabolic mirror. The corresponding parabolic mirror then has to pass through the intersection points of  $\mathbf{S}_{img}$  with the image plane  $\mathbf{P}_{img}$ . The inversion sphere  $\mathbf{S}_{inv}$  has to be centered on  $\mathbf{C}_{inv}$  and has to pass through the intersection points of  $\mathbf{S}_{img}$  with  $\mathbf{P}_{img}$ . This fixes the radius of  $\mathbf{S}_{inv}$  to be  $\sqrt{2}$ .

If the location and radius of the inversion sphere  $\mathbf{S}_{inv}$  is fixed, the only free parameter left in the inversion camera model from equation (1) is the dilation, i.e. scaling of the image.

It may be shown that the relation between the image scaling  $d$  and the focal length of the parabolic mirror  $\mu$  is given by  $d = 1/(2\mu)$ . The image point transformation operator for such a parabolic mirror setup is therefore  $\mathbf{K} = \mathbf{S}_{inv} \mathbf{D}$  whereby

$$\mathbf{S}_{inv} = -e_3 - \frac{1}{2} e_\infty + e_o, \quad \mathbf{D} = 1 + \frac{2\mu - 1}{2\mu + 1} \mathbf{E}. \tag{4}$$

### 3 Experiments

The accuracy of the inversion camera model as compared to other lens distortion models, is the same as that of the division model introduced in [3]. A comparison of a number lens distortion models, including the division model can be found in [1], where it is shown that the division model with one free parameter has a rectification quality that is comparable to a fourth order radial polynomial approach with two free parameters.

To demonstrate the feasibility of the inversion camera model in the context of an application, monocular pose estimation experiments were carried out. In these experiments not only the pose of a known object from a single camera view was estimated but also the camera’s focal length and lens distortion. In the case of a catadioptric imaging system with a parabolic mirror, the object’s pose and the mirror’s focal length were computed.

The monocular pose estimation treated here, assumes that a model of the object is known, whose pose in space (location and orientation) is to be estimated. This model is given as a set of feature points, and it is also assumed that the correspondences between object feature points and image points are known.

**Monocular Pose Estimation.** The problem is therefore to evaluate the transformation operator (motor)  $\mathbf{M}$ , such that a model point  $\mathbf{X}$  comes to lie on the projection ray of a corresponding image point  $\mathbf{Y}$ . If lens distortion is present or a catadioptric imaging system is used, the image point has to be transformed to a rectified point  $\mathbf{Z}$ , via  $\mathbf{Z} = \mathbf{K} \mathbf{Y} \widetilde{\mathbf{K}}$ , where  $\mathbf{K}$  implements the inversion camera model. Hence, the transformed model point  $\mathbf{X}$  has to lie on the projection ray through the focal point  $\mathbf{F}$  and  $\mathbf{Z}$ . This can be formulated in CGA as

$$((\mathbf{K} \mathbf{Y} \widetilde{\mathbf{K}}) \wedge \mathbf{F} \wedge \mathbf{e}_\infty) \wedge (\mathbf{M} \mathbf{X} \widetilde{\mathbf{M}}) = 0. \quad (5)$$

If  $\mathbf{K}$  is known, then this is basically the same as the pose estimation constraint in [13]. In contrast to [13] both operators  $\mathbf{M}$  and  $\mathbf{K}$  are estimated here using the same concepts as in [10,9]. That means, equation (5) is written as a multilinear equation which is quadratic in the components of  $\mathbf{M}$  and  $\mathbf{K}$ . This equation is then linearized so that a Gauss-Markov model can be employed to estimate  $\mathbf{M}$  and  $\mathbf{K}$  iteratively. The Gauss-Markov estimation is started from a very rough, automatically computed heuristic estimate and can be refined using Gauss-Helmert estimation. The whole estimation process is thus *automatic* and no a priori knowledge about a starting pose is necessary.

**Experimental Setup.** Note that the simultaneous estimation of object pose, focal length and lens distortion is only numerically stable if the object has a sufficiently large appearance in the image and its extension along the optical axis is at least the same as its extension parallel to the image plane. For the following experiments the model of a house was used which was approximately  $20 \times 15 \times 15$ cm in size (L×H×W).

This house model was moved by a robot arm in front of a stationary camera. Since the robot movements have a positioning uncertainty of below 1 mm, these positions can be used as ground truth. Note that the model was not rotated since an exact calibration of the rotation center with respect to the model was not possible. The model was translated in an area of approximately 50cm parallel to the image plane and 35cm perpendicular to the image plane. The closest approach of the object to the camera was approximately 10cm.

Neither an internal nor an external calibration of the cameras was carried out before the pose estimation experiments. However, the CCD-chip size in millimeters and its resolution in pixels were known and it was assumed that the optical axis passes at a right angle through the center of the CCD-chip.

Two different cameras were used. A Nikon D70 digital SLR camera with a pixel size of  $7.8 \times 7.8 \mu\text{m}$  and a resolution of  $3008 \times 2000$  pixel was used to take pictures with three different lenses: the zoom lens SIGMA DC 18-125mm, 1:3.5-5.6 D, set to 18mm, the Nikkor AF Fisheye 10.5mm, 1:2.8 G ED, and the Sigma 8mm 1:4.0 EX DG Circular-Fisheye. The other camera was a LogLux camera-link camera with a resolution of  $1280 \times 1024$  pixel and a pixel size of  $6.7 \times 6.7 \mu\text{m}$ , which was used with the parabolic mirror catadioptric imaging system.

Eight markers were attached to the visible corners of the house model. The correspondences between these model points and their apparent positions in the images were found manually. The constraint equation given in equation (5) was then used to estimate the object pose and camera model parameters for each of the images taken. For each camera-lens setup the house was moved to the same six positions. Because no external calibration of the cameras with the robot arm was available, the pose estimation accuracy is measured as the difference between the true and the estimated object translations.

In fact, the 15 difference vectors between all pairs of the 6 house positions were evaluated separately for the true and the estimated house positions. Then the



rotation was found that best aligned the true and estimated difference vectors. This is basically an external calibration of the camera. Two quality measures were then calculated. First, the root mean squared (RMS) Euclidean distance between the true and estimated aligned difference vectors and second, the RMS of the ratio of the Euclidean distance between the difference vectors and the length of the true vector. That is, the latter is the RMS percentage error.

The algorithm was implemented in CLUScript, an interpreted programming language, and was executed with CLUCalc [8]. The software was run on a 1.6GHz Pentium M processor. An optimized implementation in C++ may be expected to increase the execution speed by a factor of 10.

**Results.** The results of the experiments are shown in table 1. It may be surprising that the pose estimation is most accurate for the fisheye lenses, which were found not to be exactly representable by the inversion camera model. This is because the house model only appeared in part of the image, whose distortion can be modeled quite well locally. Furthermore, the camera could be placed closer to the objects with the fisheye lenses (8mm, 10.5mm), than with the 18mm lens. The larger error in the LogLux camera results are mainly due to the low effective resolution when using a parabolic mirror. A 360 degree view is in this case mapped to a circular band in the image.

Note again that these pose estimation results were achieved without a full camera calibration. Instead focal length and lens distortion were estimated simultaneously with the object pose.

**Table 1.** Experimental results of pose estimation

Camera/Lens	RMS Err.	Rel. RMS Err.	Mean Iter.	Mean Time
D70 / 8mm	2.63mm	1.48%	5.17	0.41s
D70 / 10.5mm	2.37mm	1.50%	5.50	0.41s
D70 / 18mm	5.51mm	3.12%	5.17	0.42s
LogLux / Cata.	7.87mm	3.66%	8.83	0.70s

## 4 Conclusions

In this paper a novel camera model is introduced, the *inversion camera model*. It combines in a single model the standard pinhole camera model, the division model of lens distortion and the model of parabolic mirror imaging systems. The inversion camera model is based on the inversion of image points in a sphere, which can be expressed in a straight forward manner in Geometric Algebra as a multilinear operator. This also implies that the camera model operator can be treated just like any other transformation operator in Geometric Algebra, as for example, a Euclidean transformation. Thus linear statistical estimation methods as presented in [10,9] can be applied.

The experimental results show that this camera model can be employed successfully in the simultaneous estimation of object pose, and camera model pa-

rameters in a 'half' calibrated camera setup. Next to the model's good behaviour in an actual application, it is also another example of the unifying nature of Geometric Algebra.

## References

1. David Claus and Andrew W. Fitzgibbon. A rational function lens distortion model for general cameras. In *CVPR (1)*, pages 213–219, 2005.
2. Leo Dorst. Honing geometric algebra for its use in the computer sciences. In G. Sommer, editor, *Geometric Computing with Clifford Algebra*, pages 127–151. Springer-Verlag, 2001.
3. A. W. Fitzgibbon. Simultaneous linear estimation of multiple view geometry and lens distortion. In *CVPR (1)*, pages 125–132, 2001.
4. C. Geyer and K. Daniilidis. Catadioptric projective geometry. *International Journal of Computer Vision*, (45):223–243, 2001.
5. R. I. Hartley and A. Zissermann. *Multiple View Geometry in Computer Vision*. CUP, Cambridge, UK, 2 edition, 2003.
6. D. Hestenes and G. Sobczyk. *Clifford Algebra to Geometric Calculus: A Unified Language for Mathematics and Physics*. Reidel, Dordrecht, 1984.
7. E. Kilpelä. Compensation of systematic errors of image and model coordinates. *International Archives of Photogrammetry*, XXIII(B9):407–427, 1980.
8. C. Perwass. CLUCalc. WWW, <http://www.clucalc.info/>, 2005.
9. C. Perwass and W. Förstner. *Uncertain Geometry with Circles, Spheres and Conics*, volume 31 of *Computational Imaging and Vision*, pages 23–41. Springer-Verlag, 2006.
10. C. Perwass, C. Gebken, and G. Sommer. Estimation of geometric entities and operators from uncertain data. In *27. Symposium für Mustererkennung, DAGM 2005, Wien, 29.8.-2.9.005*, volume 3663 of *LNCS*, pages 459–467. Springer-Verlag, Berlin, Heidelberg, 2005.
11. C. Perwass and D. Hildenbrand. Aspects of geometric algebra in Euclidean, projective and conformal space. Technical Report Number 0310, CAU Kiel, Institut für Informatik, September 2003.
12. C. Perwass and G. Sommer. Numerical evaluation of versors with Clifford algebra. In Leo Dorst, Chris Doran, and Joan Lasenby, editors, *Applications of Geometric Algebra in Computer Science and Engineering*, pages 341–349. Birkhäuser, 2002.
13. B. Rosenhahn and G. Sommer. Pose estimation in conformal geometric algebra, part II: Real-time pose estimation using extended feature concepts. *Journal of Mathematical Imaging and Vision*, 22:49–70, 2005.

# Determining an Initial Image Pair for Fixing the Scale of a 3D Reconstruction from an Image Sequence

Christian Beder and Richard Steffen

Institute for Photogrammetry  
Bonn University, Germany  
beder@ipb.uni-bonn.de  
rsteffen@uni-bonn.de

**Abstract.** Algorithms for metric 3d reconstruction of scenes from calibrated image sequences always require an initialization phase for fixing the scale of the reconstruction. Usually this is done by selecting two frames from the sequence and fixing the length of their base-line. In this paper a quality measure, that is based on the uncertainty of the reconstructed scene points, for the selection of such a stable image pair is proposed. Based on this quality measure a fully automatic initialization phase for simultaneous localization and mapping algorithms is derived. The proposed algorithm runs in real-time and some results for synthetic as well as real image sequences are shown.

## 1 Introduction

In recent years the fully automatic 3d reconstruction of scenes and camera trajectories from monocular image sequences has received a lot of attention. In the early work of [7] and [6], the extraction of feature points together with their uncertainty represented by covariance matrices was developed. More recently, feature extraction and tracking of features across image sequences was improved by [24], [14] and [13]. This reliable feature extraction methods enabled the 3d reconstruction from image sequences (e.g. [1], [29],[18]) using robust estimation of the epipolar geometry. The use of self-calibration techniques (cf. [20],[19]) or prior knowledge of the internal camera calibration leads to a metric 3d reconstruction, that is defined up to a similarity transformation. In the calibrated case efficient real-time algorithms, that are also able to cope with planar scenes were developed by [17] and [25]. Starting from this prerequisites the field of real-time simultaneous localization and mapping has recently emerged and was given much attention by many researchers (e.g. [4],[3],[5],[27],[15],[23]).

As the calibrated 3d reconstruction is only defined up to a similarity transformation, somehow fixing the scale is an important task. Although scale is a gauge parameter and therefore does not affect the overall accuracy of the reconstruction it does affect the stability of the reconstruction algorithms and must therefore be chosen carefully. Usually this is done by initially selecting two reference images

and fixing the length of their base-line. Those key-frames are selected based on image sharpness and disparity (cf. [16]), based on the distribution of matched points in the images (cf. [12]), based on selecting the most appropriate motion model (cf. [28],[21],[22]) or based on evaluating the bundle-adjustment of the whole sequence (cf. [26]).

The contribution of this work is to present a statistically motivated measure for the quality of the pair of reference images. Based on this quality measure an efficient algorithm is proposed, that automates the manual setting of initial number of frames or the initialization phase required for example by the approach of [4]. It turns out, that, in case of known internal camera calibration, this is a very efficient alternative to the model selection approach of [28],[21] and [22], who proposed to decide when the base-line is large enough by checking, if the image pair is related only by a homography or the full epipolar constraint. A drawback of this approach is, that it cannot handle planar objects, which our approach can.

To achieve this goal, another subject of recent computer vision research is employed. It has been studied by [2] [11], [10],[8] and [9], how uncertainties can be efficiently represented and propagated for geometric reasoning tasks involving projective geometric entities. Especially the work of [9], who showed how covariance matrices easily transform for various projective geometric constructions, plays a key role in this work.

The paper is organized as follows: In section 2 first the shape of confidence ellipsoids of scene points resulting from a given point correspondence and camera pose is exploited. Based on this shape, more explicitly its roundness, a measure for the quality of the image pair for the task of fixing the scale is derived. In section 3 an algorithm is outlined, that is used to determine the optimal image pair for fixing the scale of a 3d reconstruction. Finally some results on simulated and real image sequences are shown in section 4.

## 2 Confidence Ellipsoids of Scene Points

Now the propagation of uncertainty from two measured corresponding image points on the reconstructed scene point is analyzed. If a scene point  $\mathbf{X}$  is observed by two projective cameras  $\mathbf{P}'$  and  $\mathbf{P}''$ , the image coordinates are

$$\mathbf{x}' \cong \mathbf{P}'\mathbf{X} \quad (1)$$

and

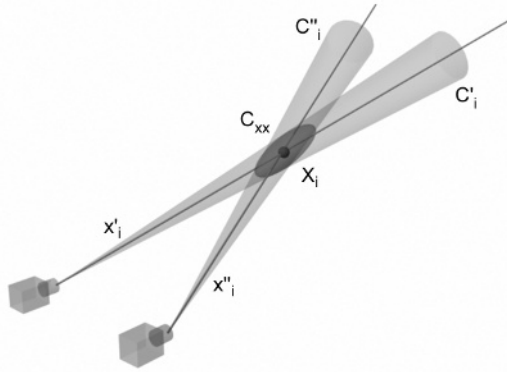
$$\mathbf{x}'' \cong \mathbf{P}''\mathbf{X} \quad (2)$$

Denoting with  $\mathbf{S}(\cdot)$  the first two rows of the skew-symmetric matrix inducing the cross-product

$$\mathbf{S}(\mathbf{x}) = \begin{pmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \end{pmatrix} \quad (3)$$

the two conditions can be written as

$$\mathbf{S}(\mathbf{x}')\mathbf{P}'\mathbf{X} = -\mathbf{S}(\mathbf{P}'\mathbf{X})\mathbf{x}' = \mathbf{0} \quad (4)$$



**Fig. 1.** Scene geometry: Projecting rays of two corresponding image points  $\mathbf{x}'_i$  and  $\mathbf{x}''_i$  together with their uncertainties  $C'_i$  and  $C''_i$  are observed by two cameras. The corresponding scene point  $\mathbf{X}_i$  has the uncertainty ellipsoid  $C_{XX}$ . The roundness of this object, i.e. the ratio of its smallest and longest axis, is a measure of the quality of the scene geometry.

and

$$S(\mathbf{x}'')P''\mathbf{X} = -S(P''\mathbf{X})\mathbf{x}'' = \mathbf{0} \tag{5}$$

if the image points are not at infinity. Both expressions are linear in the scene point as well as in the image points, i.e.

$$\underbrace{\begin{pmatrix} S(\mathbf{x}')P' \\ S(\mathbf{x}'')P'' \end{pmatrix}}_{\mathbf{A}_{4 \times 4}} \mathbf{X} = \mathbf{0} \tag{6}$$

and

$$\underbrace{\begin{pmatrix} -S(P'\mathbf{X}) & 0 \\ 0 & -S(P''\mathbf{X}) \end{pmatrix}}_{\mathbf{B}_{4 \times 6}} \begin{pmatrix} \mathbf{x}' \\ \mathbf{x}'' \end{pmatrix} = \mathbf{0} \tag{7}$$

Now the scene point coordinates and the two image point coordinates are assumed to be random variables. Note that, as all three quantities are homogeneous, the covariance matrices of their distributions are singular. Let the covariance matrices of the image points  $\mathbf{x}'$  and  $\mathbf{x}''$  be given by  $C'$  and  $C''$  respectively, then it has been shown by [9], that the covariance matrix  $C_{XX}$  of the distribution of the scene point coordinates  $\mathbf{X}$  is proportional to the upper left  $4 \times 4$ -submatrix

$$C_{XX} = (N^{-1})_{1:4,1:4} \tag{8}$$

of the inverse of

$$\mathbf{N}_{5 \times 5} = \begin{pmatrix} \mathbf{A}^T \left( \mathbf{B} \begin{pmatrix} C' & 0 \\ 0 & C'' \end{pmatrix} \mathbf{B}^T \right)^{-1} \mathbf{A} \mathbf{X} \\ 0 \end{pmatrix} \tag{9}$$

Note, that no specific distribution must be assumed, as all arguments regard only the second moments. We have neglected the effect of the uncertainty of the projection matrices  $\mathbf{P}'$  and  $\mathbf{P}''$  here, as the relative orientation of the two cameras is determined by many points, so that it is of superior precision compared to a single point.

Now the effect of normalizing the homogeneous vector  $\mathbf{X} = [\mathbf{X}_0^T, X_h]^T$  to Euclidean coordinates on its covariance matrix  $\mathbf{C}_{XX}$  is analyzed. For this to be meaningful it is assumed, that the cameras are calibrated, so that the reconstruction is Euclidean, i.e. defined up to a similarity transformation. The Jacobian of a division of  $\mathbf{X}_0$  by  $X_h$  is

$$J_e = \frac{\partial \mathbf{X}_0 / X_h}{\partial \mathbf{X}_0} = \frac{1}{X_h} \left( I_{3 \times 3} - \frac{\mathbf{X}_0}{X_h} \right) \quad (10)$$

and hence by linear error propagation the covariance matrix of the distribution of the corresponding Euclidean coordinates is

$$\mathbf{C}^{(e)} = J_e \mathbf{C}_{XX} J_e^T \quad (11)$$

The roundness of the confidence ellipsoid is directly related to the condition number of the 3d reconstruction of the point. Therefore it is a measure, how well the two camera poses are suited for the task of 3d reconstruction. Hence, using the singular value decomposition of this covariance matrix

$$\mathbf{C}^{(e)} = U \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix} V^T \quad (12)$$

its roundness is defined as the square root of the quotient of the smallest and the largest singular value

$$R = \sqrt{\frac{\lambda_3}{\lambda_1}} \quad (13)$$

This measure lies between zero and one, is invariant to scale changes and only depends on the relative geometry of the two camera poses, the image points and the object. If the two camera centers are the same, it is equal to zero. If the object is equally far away from the two cameras and the projecting rays of the image points are orthogonal and their covariance matrices are identical and isotropic, it is equal to  $\sqrt{\frac{1}{2}}$ . This is the maximum under the assumption of isotropic covariance matrices. The maximum of one is reached for the same configuration as before except for the covariance matrices of the projecting rays. The principal axis of this covariance matrices must therefore be aligned with the epipolar plane with the extension perpendicular to it being  $\sqrt{2}$  times the extension perpendicular to the viewing direction.

### 3 Determining the Optimal Image Pair

Now the roundness measure of the previous section will be put into the context of finding the optimal image pair for fixing the scale of a 3d reconstruction. Of

course the global optimal solution can only be found by checking all image pairs. As our intended application is the real-time initialization of a simultaneous localization and mapping system, the proposed algorithm fixes the first frame of the sequence and terminates, when the first acceptable second frame is reached. The acceptability will be determined via the roundness of the confidence ellipsoids of the reconstructed scene points. The details are as follows:

1. Fix the first image of the sequence and let its projection matrix be

$$\mathbf{P}' = [\mathbf{I} | \mathbf{0}]$$

2. Extract the interest points  $\mathbf{q}'_i$  together with their covariance matrices  $\mathbf{C}_{q'_i q'_i}$  from this image and apply the known camera calibration matrix  $\mathbf{K}$  to the image coordinates and their covariance matrices, yielding the directions

$$\mathbf{x}'_i = \mathbf{K}^{-1} \mathbf{q}'_i$$

and their covariance matrices

$$\mathbf{C}'_i = \mathbf{K}^{-1} \mathbf{C}_{q'_i q'_i} \mathbf{K}^{-T}$$

3. For each new image of the sequence do the following

- (a) Extract the interest points  $\mathbf{q}''_i$  together with their covariance matrices  $\mathbf{C}_{q''_i q''_i}$  from this new image and apply the known camera calibration matrix yielding again the directions

$$\mathbf{x}''_i = \mathbf{K}^{-1} \mathbf{q}''_i$$

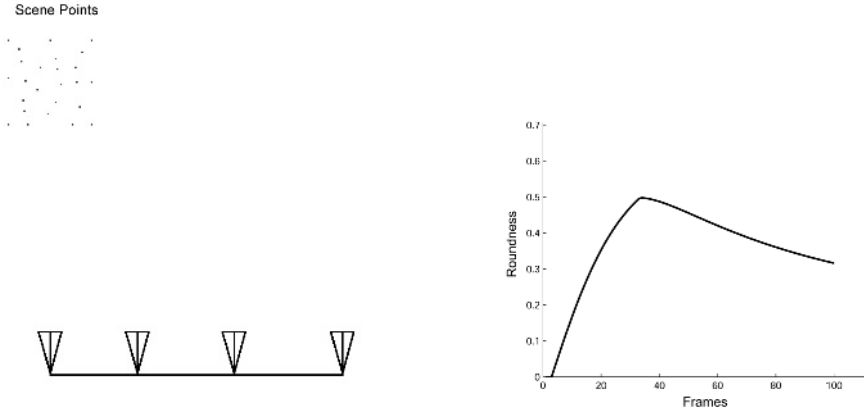
and their covariance matrices

$$\mathbf{C}''_i = \mathbf{K}^{-1} \mathbf{C}_{q''_i q''_i} \mathbf{K}^{-T}$$

- (b) Determine the point correspondences  $\mathbf{x}'_i \leftrightarrow \mathbf{x}''_i$  and relative orientation  $\mathbf{R}$ ,  $\mathbf{t}$  to the first image of the sequence according to the algorithm proposed in [17]. The camera matrix for the current image is then

$$\mathbf{P}'' = [\mathbf{R} | -\mathbf{t}]$$

- (c) Determine the scene point positions  $\mathbf{X}_i$  for each found correspondence by forward intersection. This can for example be done by solving the homogeneous equation system (6) using the singular value decomposition of the matrix  $\mathbf{A}$ .
- (d) Determine the roundness  $R_i$  (cf. equation (13)) of each scene point  $\mathbf{X}_i$ 's confidence ellipsoid as outlined in the previous section.
- (e) If the mean roundness is above a given threshold  $T$ , use this image pair to fix the scale of the reconstruction and continue with the main application.



**Fig. 2.** Left: Synthetic image sequence trajectory of the translation experiment, where the camera faces the object and is moved to the right. Right: Roundness measure against video frame for the synthetic translation experiment. A maximum is reached at the frame, where the angle of the rays is approximately  $35^\circ$ . As the distance to the object increases, the roundness decreases again.

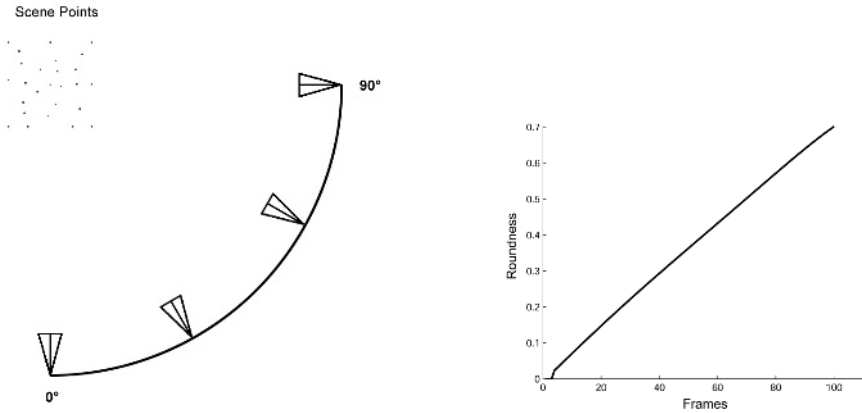
## 4 Results

To evaluate the usefulness of the proposed roundness measure, experiments on synthetic as well as real image sequences were carried out. The setup for the synthetic experiments is shown on the left hand sides of figure 2 and figure 3. The cameras were assumed to be normalized and the image points were assumed to have isotropic and equal covariance matrices. Note, that by definition the overall scale is irrelevant, since the proposed roundness measure only depends on the relative scales and is therefore scale-invariant.

In the first experiment, depicted in figure 2, the camera was facing the object and then moved to the right. The resulting roundness measure is shown on the right hand side in figure 2. It can be observed, that a maximum roundness is reached, where the angle of the projecting rays is approximately  $35^\circ$ . As the distance of the second camera to the object increases, the roundness decreases again. The optimal image pair, i.e. the pair yielding highest stability, is therefore not only dependent on the intersection angle of the projecting rays, but also on the relative distances of the cameras to the object.

The second synthetic experiment was to rotate the camera around the object at equal distance as depicted on the left hand side of figure 3. The resulting roundness measure is shown on the right hand side in figure 3. It can be seen, that it directly corresponds to the rotation angle and the maximum of  $\sqrt{\frac{1}{2}}$  is reached at an angle of  $90^\circ$ , where the intersection of the rays is optimal for the accuracy of the 3d reconstruction.





**Fig. 3.** Left: Synthetic image sequence trajectory of the rotation experiment, where the camera is moved at equal distance around the object. Right: Roundness measure against video frame for the synthetic rotation experiment. The maximum of  $\sqrt{\frac{1}{2}}$  is reached at the angle of  $90^\circ$ .

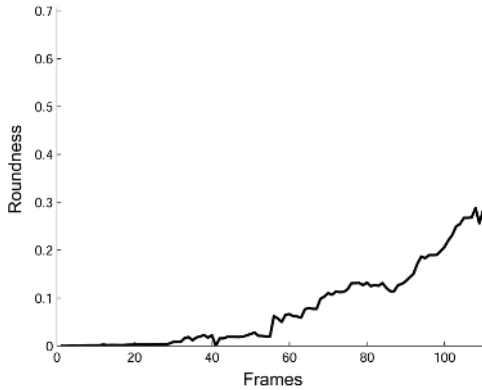
Finally a real image sequence was taken using a cheap hand-held consumer web-cam. Two exemplary frames are depicted in figure 4. Features were extracted and tracked and the roundness measure was computed for each frame with respect to the first image, which is depicted on the left hand side in figure 4. The resulting roundness measure is shown in figure 5. In the first 25 frames the camera was not moved, so that the roundness measure stays near to zero. When the camera starts moving, the expected accuracy of the depth of the 3d reconstruction, and hence the proposed roundness measure, is increasing. After about 110 frames the roundness measure reached the threshold  $T = \sqrt{\frac{1}{10}}$ . This threshold is not a critical parameter but a minimal requirement stemming from the goal of achieving a condition number of approximately 10 for the 3d reconstruction. The corresponding last frame is depicted on the right hand side in figure 4. It can be seen, that still enough corresponding points can be identified, so that the determination of the relative orientation between the frames is not an issue. Note also, that all processing was performed in real-time.

## 5 Conclusion

A fully automatic real-time algorithm for initially fixing the scale of the 3d reconstruction in simultaneous localization and mapping applications with calibrated cameras was proposed. As the metric reconstruction is fixed up to a similarity transformation in the calibrated case, the shape of the confidence ellipsoids of reconstructed scene points is a meaningful quantity. The roundness of this confidence ellipsoids can be used to decide, when the accuracy of the



**Fig. 4.** Left: First image of the real image sequence. Right: Last image of the real image sequence, where the roundness of the scene point covariance matrices was sufficiently high.



**Fig. 5.** Roundness measure against video frame for the real image sequence. The camera was not moved for the first 25 frames. The threshold value of  $T = \sqrt{\frac{1}{10}}$  was reached after the movement was sufficiently large on the frame depicted on the right in figure 4.

reconstruction is most stable, as it is directly related to the condition number of the 3d reconstruction of the point. Hence, choosing the image pair for which this roundness is maximal is the most stable choice for initially fixing the scale of a 3d reconstruction.

The proposed algorithm was demonstrated to work on synthetic as well as real monocular image sequences. Since the most complex operations are only the inversion of a  $5 \times 5$ -matrix and two singular value decompositions of a  $4 \times 4$ - and a  $3 \times 3$ -matrix, the dominant part of the computation time is taken by the feature extraction and tracking, enabling a real-time initialization phase.

## References

1. Paul A. Beardsley, Philip H. S. Torr, and Andrew Zisserman. 3d model acquisition from extended image sequences. In *Proc. of ECCV (2)*, pages 683–695, 1996.
2. A. Criminisi, I. Reid, and A. Zisserman. A plane measuring device. *Image and Vision Computing*, 17(8):625–634, 1999.
3. A. J. Davison and D. W. Murray. Simultaneous localisation and map-building using active vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):865–880, July 2002.
4. A.J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *Proc. International Conference on Computer Vision, Nice*, pages 1403–1410, October 2003.
5. J. Diebel, K. Reuterswrd, S. Thrun, J. Davis, and R. Gupta. Simultaneous localization and mapping with active stereo vision. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3436–3443, 2004.
6. W. Förstner. A framework for low level feature extraction. In *Proc. of European Conference on Computer Vision*, pages 383–394, 1994.
7. C.G. Harris and M.J. Stephens. A combined corner and edge detector. In *Fourth Alvey Vision Conference*, pages 147–151, 1988.
8. S. Heuel and W. Förstner. Matching, reconstructing and grouping 3d lines from multiple views using uncertain projective geometry. In *CVPR '01. IEEE*, 2001.
9. Stephan Heuel. *Uncertain Projective Geometry - Statistical Reasoning for Polyhedral Object Reconstruction*, volume 3008 of *LNCS*. Springer, 2004.
10. Kenichi Kanatani. Uncertainty modeling and model selection for geometric inference. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 26(10):1307–1319, 2004.
11. Kenichi Kanatani and Daniel D. Morris. Gauges and gauge transformations for uncertainty description of geometric structure with indeterminacy. *IEEE Transactions on Information Theory*, 47(5):2017–2028, July 2001.
12. Reinhard Koch, Marc Pollefeys, and Luc Van Gool. Robust calibration and 3d geometric modeling from large collections of uncalibrated images. In W. Förstner, J. Buhmann, A. Faber, and P. Faber, editors, *Proceedings of the DAGM*, Informatik Aktuell, pages 412–420. Springer, 1999.
13. David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, 2004.
14. J. Matas, O.Chum, M.Urban, and T.Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *BMVC*, pages 384–393, 2002.
15. Jason Meltzer, Rakesh Gupta, Ming-Hsuan Yang, and Stefano Soatto. Simultaneous localization and mapping using multiple view feature descriptors. In *Proc. of IROS*, 2004.
16. David Nistér. Frame decimation for structure and motion. In *SMILE*, pages 17–34, 2000.
17. David Nistér. An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):756–777, 2004.
18. M. Pollefeys, R. Koch, M. Vergauwen, and L. Van Gool. Automated reconstruction of 3d scenes from sequences of images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 55(4):251–267, 2000.
19. Marc Pollefeys and Luc Van Gool. A stratified approach to metric self-calibration. In *Proc. CVPR*, pages 407–412, 1997.

20. Marc Pollefeys and Luc Van Gool. Stratified self-calibration with the modulus constraint. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21(8):707–724, 1999.
21. Marc Pollefeys, Luc Van Gool, Maarten Vergauwen, Kurt Cornelis, Frank Verbiest, and Jan Tops. Video-to-3d. In *Proceedings of Photogrammetric Computer Vision*, 2002.
22. Jason Repko and Marc Pollefeys. 3d models from extended uncalibrated video sequences: Addressing key-frame selection and projective drift. In *Proc. of 3DIM*, 2005.
23. S. Se, D. G. Lowe, and J. J. Little. Vision-Based Global Localization and Mapping for Mobile Robots. *IEEE Transactions on Robotics*, 21(3):364–375, 2005.
24. Jianbo Shi and Carlo Tomasi. Good features to track. In *1994 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94)*, pages 593 – 600, 1994.
25. Henrik Stewenius, Christopher Engels, and David Nister. Recent developments on direct relative orientation. *ISPRS Journal*. to appear.
26. Thorsten Thormählen, Hellward Broszio, and Axel Weissenfeld. Keyframe selection for camera motion and structure estimation from multiple views. In *Proceedings of European Conference on Computer Vision*, pages 523–535, 2004.
27. S. Thrun, M. Montemerlo, D. Koller, B. Wegbreit, J. Nieto, and E. Nebot. Fast-slam: An efficient solution to the simultaneous localization and mapping problem with unknown data association. *Journal of Machine Learning Research*. To appear.
28. Philip H. S. Torr. Bayesian model estimation and selection for epipolar geometry and generic manifold fitting. *International Journal of Computer Vision*, 50(1):35–61, 2002.
29. Philip H. S. Torr and Andrew Zisserman. Feature based methods for structure and motion estimation. In *Workshop on Vision Algorithms*, pages 278–294, 1999.

# Multi-camera Radiometric Surface Modelling for Image-Based Re-lighting

Oliver Grau

BBC Research, Kingswood Warren, Tadworth, UK  
oliver.grau@rd.bbc.co.uk  
<http://www.bbc.co.uk/rd>

**Abstract.** This contribution describes an automatic method to retrieve the diffuse radiometric surface model of moving persons or other objects along with the object geometry using a multi-camera system. The multi-camera equipped studio allows synchronised capture of the foreground action and a visual hull computation is then used to compute a 3D model of that scene. The diffuse surface reflection parameters are computed using the 3D model from that process together with an illumination map of the studio. The illumination map is a high dynamic range image generated from a series of images of the studio using a camera equipped with a spherical (fish-eye) lens. With this setup our method is able to capture any action in the studio under normal lighting.

## 1 Introduction

The integration of virtual, synthetic objects into real scenes has many applications in film and TV productions, for product and architectural visualisation. For highly realistic results it is important to match the lighting of the real with the synthetic components. Methods to capture the scene lighting of environments using high dynamic range images (HDRI) were pioneered by Paul Debevec [1] and are now widely used in production. These methods involve building up a panoramic representation, by either mapping the environment onto a sphere or a cube. The inserted virtual object is then lit by this HDR illumination map and pasted into the background plate, which is an image of the real scene.

Any object to be inserted into the HDR environment needs to have a surface description that gives the surface properties along with the object shape (or geometry) for the rendering system. This might be the BRDF (bidirectional reflectance distribution function) in the general case or more simplified colour, diffuse and specular reflection parameters. For synthetic objects these parameters are usually assigned manually in the rendering system.

The accurate measurement of the BRDF requires a very defined environment. Approaches usually assume a calibrated environment where camera parameters and object shape are precisely known, e.g. by using a laser scanner [2,3,4]. Further a fixed point light source with known position is used for the computation. Debevec et al. [5] described a method to capture an image based representation of a face by taking a series of images under controlled variation of a light source.

The setups for the mentioned approaches are designed to acquire the surface properties offline and are not able to capture a live action.

The aim for the approach described in this paper is to integrate a sequence of images taken from an actor in a studio into a different lighting situation. This task is also called re-lighting. Since the actor is moving it is not possible to use offline modelling tools like a laser scanner to create highly accurate 3D geometry. Furthermore the dynamic range of the images is limited (8-bit). The approach described here uses a multi-camera system that can capture images of a moving actor synchronously in a chroma-keying environment. A 3D model of the actor is generated using a visual hull computation. A brief outline of this step is given together with an overview of our approach in the next section. A more detailed description of the studio setup can be found in [6]. In addition to the 3D geometry an illumination map is created using spherical (fisheye) HDR images of the studio. The use of the illumination map makes our approach applicable to any lighting situation found in realistic production scenarios. For the computation of the surface properties we currently focus on the diffuse component.

The remainder of this paper describes a method for calibrating the radiometry of the camera system in section 3. Section 4 introduces a new method for the robust computation of radiometric surface properties from multi-camera images. The paper finishes with some results and concluding remarks.

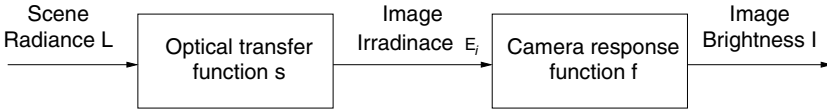
## 2 Overview

The computation of radiometric surface properties from multi-camera images consists of the following processing steps:

1. Geometrical calibration of the multi-camera system
2. Foreground/background segmentation
3. 3D reconstruction using a visual hull computation
4. Radiometric camera calibration
5. Capture of an illumination map of the studio lighting environment
6. Computation of radiometric surface properties

The steps 1-3 are common practice to generate a 3D polygonal surface model  $O$  of the actor or object: The *geometrical calibration* uses a 1m x 1m planar chart. The cameras used for our experiments were Sony DXC-9100P cameras operating in 25 fps progressive mode. From a set of 10-20 multi-camera still frames the calibration method computes a set of camera parameters simultaneously considering centre-point shift and radial distortions. The *segmentation* is based on a chroma-keying facility that is available in the experimental studio, but other methods are applicable (e.g. difference keying). The *visual hull computation* is based on a hierarchical octree approach similar to [7,8] with refinement using super-sampling and (moderate) Gaussian smoothing [9] followed by a marching cubes iso-surface generation.

The *radiometric camera calibration* consists of two steps: The set-up phase is basically a colour balance that defines the operating mode of the camera.



**Fig. 1.** Flow diagram of the transfer functions  $s$  and  $f$  that transform the scene radiance into image brightness (adapted from [10])

The radiometric calibration estimates the transfer function of the cameras that relates the scene radiance  $L$  to the image brightness  $I$  measured by the cameras. Fig. 1 explains this concept (see [10] for more details); The image brightness  $I$  caused by a scene radiance  $L$  is here defined by the optical transfer function  $s$  and the camera response function  $f$ :

$$I = f(E_i) \tag{1}$$

$$E_i = s(L) \tag{2}$$

With the image irradiance  $E_i$ .

The optical transfer function  $s$  considers optical effects, like vignetting, lens aberrations, depth of focus and effects like fixed pattern noise. In this work these effects are not considered, with exception of vignetting for the capture of illumination maps. The camera response function  $f$  however is very important, because the cameras usually have non-linear response. In the case of the broadcast-style Sony video cameras used, this is a deliberate feature (gamma characteristic) to compress the irradiance range. The inverted camera response  $f^{-1}$  gives the relation of image brightness to the scene radiance (ignoring optical effects here).

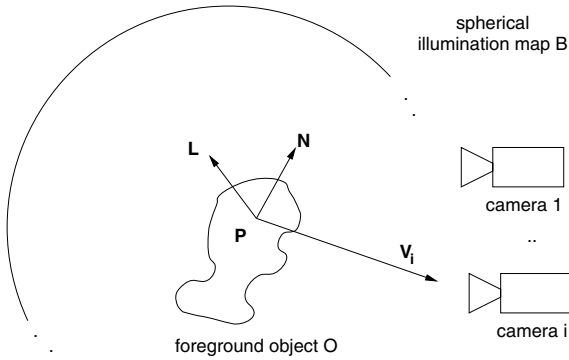
A *map of the illumination*  $B$  is generated using images from the studio taken with a spherical lens (fisheye lens) and standard HDRI techniques as outlined in the next section.

With the 3D model  $O$  of the foreground object and the illumination map  $B$  we can now define a model that describes the expected observation of the illumination of the foreground object from the observation cameras. Fig. 2 illustrates this set-up.

The last step is the *computation of radiometric surface properties* that computes the dominant diffuse surface reflectance parameters. This approach is detailed in section 4.

### 3 Multi-camera Radiometric Calibration and Capture of Illumination Map

The goal of the radiometric camera calibration is to colour balance the multi-camera system and to determine the response functions of the individual cameras. The colour balance sets for each camera the red, green and blue channel to a reference white and black object in the scene that are visible to all cameras simultaneously. As reference object a Macbeth *Colorchecker*<sup>TM</sup> chart for



**Fig. 2.** Relation of a point  $P$  on the foreground object  $O$  and the spherical background model  $B$ .  $\mathbf{N}$  represents the surface normal of the surface point,  $\mathbf{L}$  represents the direction of an incoming light ray and  $\mathbf{V}_i$  is a vector pointing to camera  $i$ .

the white and black level was used. A program has been developed to set the radiometric parameters available in the cameras used <sup>1</sup>. The colour balance determines the range  $[E_{i,min}, E_{i,max}]$  of the image irradiance that is mapped to the dynamic range of the camera, which is 8-bit per channel after digitisation. All scene objects with a radiance lower than the black or higher than the white point are clipped.

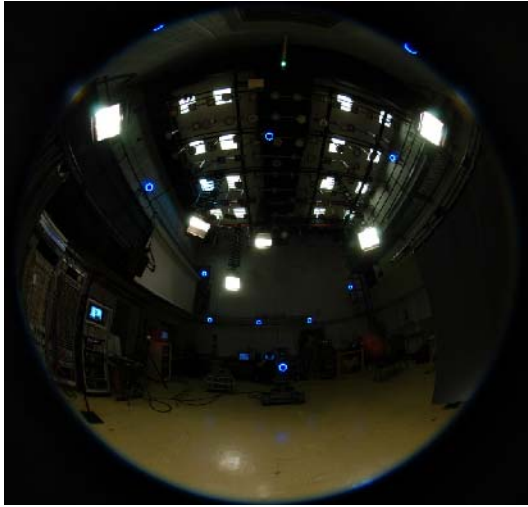
The second step of the radiometric calibration is to estimate the camera response function  $f$  for each camera. We are using the method described in [11]. This method computes a discrete lookup table for the camera response function from a number of images taken under varied camera exposure<sup>2</sup> of a static scene. This process is also known as radiometric auto-calibration. The more recent method described in [10] has the advantage that it is assuming a monotone response function (which is usually the case) and a restricted number of parameters. This makes the estimation of the parameters better conditioned.

The illumination map of the lighting situation is created using a digital stills camera (Nikon D100) equipped with a spherical lens (Costal Optics 185° field of view). Fig. 3 shows a picture taken in the experimental studio. A series of pictures is taken with different exposure times (as describe above) so that even the brightest lights are mapped into the dynamic range of the digital camera (i.e. not over exposed) and detail is retained in the dark areas. This method is called bracketing. After radiometric auto-calibration of the camera the series of images is combined into one HDRI, as described in [11]. The spherical images are transformed into a latitude-longitude mapping for further use.

<sup>1</sup> The Sony DXC-9100P cameras allow only for setting of gain for red and blue and a 'master pedestal' (black level).

<sup>2</sup> Preferably the exposure time (integration time) is varied over changing the lens aperture. In addition a neutral density filter is used to allow the capture of highlights in the studio lamps.





**Fig. 3.** A picture of the studio taken with a Costal Optics spherical lens showing all major light sources

For the use as a model of the illumination the position and orientation of the spherical probe camera and its internal parameters have to be known. This is done by registering the camera to the coordinate reference system of the multi-camera system. For this purpose a number of known positions in the studio are manually extracted in the spherical camera images and a modified calibration method is used to compute the camera pose and internal parameters.

The illumination map will be taken roughly from the position of the actors. The approximation of the studio lighting is assuming that the extent of the acting area is small compared to the distance of the lights to the probing camera. For many practical configurations it is sufficient to use only one (half) sphere since all the major light sources can be captured. This is neglecting any light bouncing from the other half sphere which can not always be tolerated. In this case both hemispheres have to be provided.

## 4 Computation of Radiometric Surface Properties

Using the configuration shown in Fig. 2 the expected appearance of a foreground object can be computed using its computed 3D shape  $O$  and the illumination map  $B$  by rendering a synthetic view for the viewpoint of camera  $i$ . Instead of using a general BRDF we are using a simplified reflection model, as common in many rendering systems to compute the intensity of a pixel  $I$ :

$$I = k_d I_d + k_s I_s + I_e \quad (3)$$

with the diffuse reflection coefficient  $k_d$ ,  $I_d$  the diffuse component,  $I_s$  the specular component and  $I_e$  the error term that covers the (residual) model error. The

determination of the specular reflection coefficient  $k_s$  requires quite accurate surface normals and is neglected here, ie.  $k_s = 0$ .

The diffuse component can be computed as the irradiance  $E$  by integrating all incoming light rays  $L_{in}$ :

$$I_d = E = \int_{\Omega(\mathbf{N})} L_{in}(\omega)(\mathbf{N}\omega)d\omega \quad (4)$$

in the direction  $\omega$  and  $\Omega$  upper hemisphere over surface point with the surface normal  $\mathbf{N}$ .

The intensity of the light rays  $L_{in}$  is taken from the illumination map. The computation of  $I_d$  can be done with standard rendering systems that provide global illumination rendering.

An often used strategy for computing global illumination is to generate a number of rays in random directions from the 3D point on the object surface. For accurate results many rays are necessary and this requires long computation times (up to several hours per frame). Therefore a specialised renderer has been developed that gives a fast computation from the illumination map: For each 3D point on the object surface the irradiance is determined by integrating directly over the related area in the illumination map taking occlusions into account. The area corresponds to the surface hemisphere, i.e. the surface normal marks the centre point of this area. This method is reducing the computational effort significantly (typically in the order of minutes on a recent PC).

The parameter  $k_d$  has three components (one for each colour channel) and can be approximated:

$$k_d = \frac{I - I_e}{I_d} \quad (5)$$

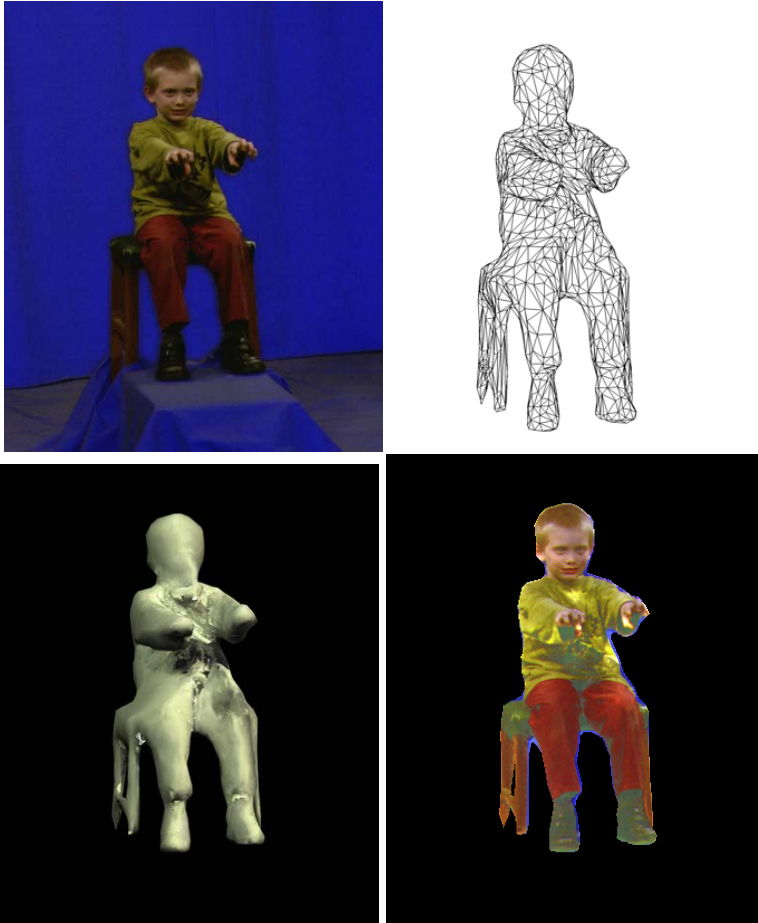
The error term  $I_e$  is used to compensate for errors in the illumination model and is a global parameter here.

In the case that the foreground object has a surface with significant specular reflection these will appear as highlights in the camera images, as depicted on the head in the left image of Fig. 6. A significant highlight is usually overexposed, ie. the image brightness values are clipped. Under these conditions it is not possible to recover the surface colour. However, it is possible to recover the colour from a different camera. This is done by determining the 3D coordinate of a pixel by reprojecting it onto the 3D model. In the next step this point is projected into all available images. Omitting all invisible surface elements (using a depth test) a set of colour values for the surface element can be retrieved. The highlight can be replaced by the median value from the list.

## 5 Results

Fig. 4 shows results of a test production. A scene with a boy was captured using 12 cameras simultaneously (just one camera image shown). Using a visual hull computation as outlined in section 2 this gives a 3D model of the scene

as depicted in the top right of Fig. 4. The model is made of 3000 triangles and shows some typical artefacts due to principle limitations of the visual hull reconstruction.

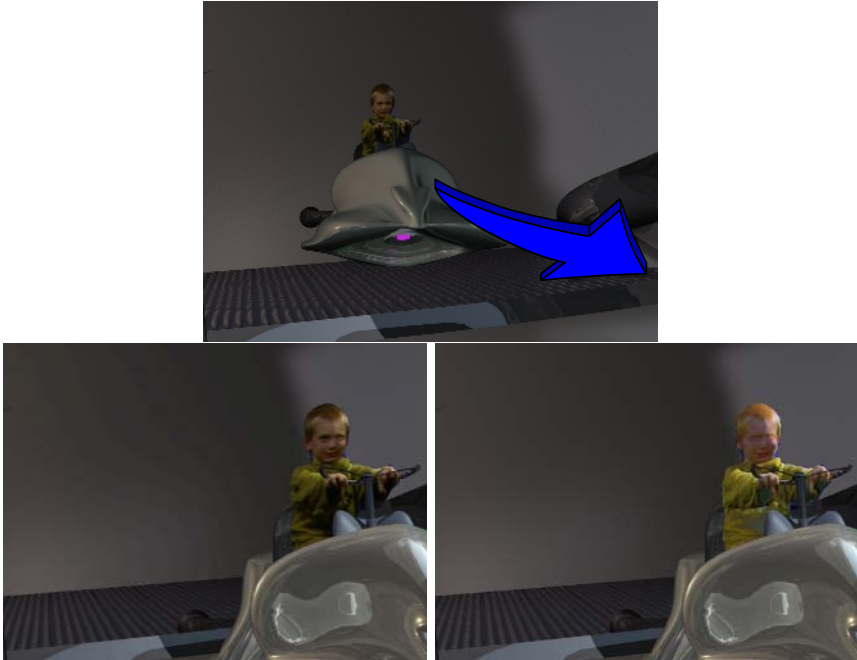


**Fig. 4.** Input image (top left) and wireframe rendered image of the 3D model with 3000 triangles (top right). The irradiance image (bottom left) and the estimated diffuse reflectance (bottom right).

The picture on the left bottom of Fig. 4 shows the irradiance image using the 3D scene model and an illumination map of the studio captured as described in section 3. The bottom right picture shows the computed diffuse reflectance for the input image. It can be clearly seen that most of the shading effects have been compensated. The remaining problems (like the area on the boy's chest) are mainly due to errors in the visual hull.

Fig. 5 gives an example of the usage of the computed reflectance model in a different illumination environment. The boy is sitting on a 'space scooter' and is

moving forward from the inside of a room (top image) into bright sunlight. The pictures are rendered with Cinema 4D (a commercial animation and rendering package). The bottom left of Fig. 5 shows the use of the original camera image in this situation and the right image shows the use of the reflectance map computed with our method. It can be seen that the use of our reflectance model is producing more realistic results under these changed lighting conditions. A video with the results can be found in [12].

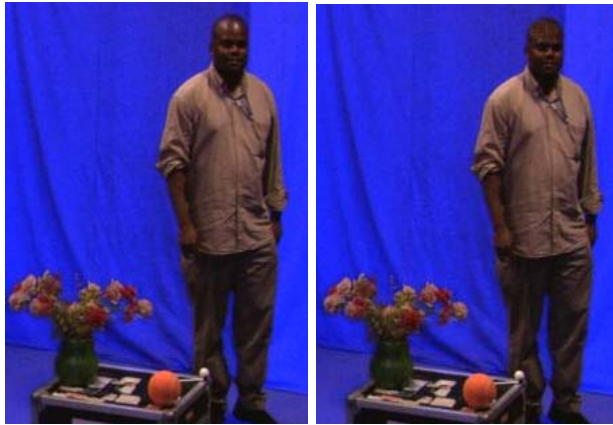


**Fig. 5.** An example of usage of the reflectance model. The top image shows the boy in a room. In the bottom he is rendered in bright sunlight using the original image (left) and our reflectance model (right).

Fig. 6 depicts an example of a person with a more shiny skin. In this case the specular components appear as highlights in the image (left). The image on the right shows the elimination of these highlights by using image information from the other cameras of the multi-camera system as described in the previous section.

## 6 Conclusions

This contribution described a method to estimate the diffuse reflectance parameters of actors captured with a multi-camera system by using an illumination map and a 3D model of the actors.



**Fig. 6.** Compensation of specular highlights by using image information from the multi-camera system

The results in the previous sections have shown that the proposed approach is increasing the range in which the illumination can be changed from the original studio lighting. The additional operational overhead for achieving that is relatively low since only the illumination map has to be captured in addition to the set-up of the multi-camera system.

A limiting factor of the method is the quality of the 3D models in this approach. In particular the surface normals that can be derived from the visual hull computation are not very precise. The diffuse component of the reflection can still be computed quite robustly since it is integrating over the hemisphere of each object surface point. The specular components are very sensitive to wrong surface normals. Therefore this paper was focussing on the diffuse components.

However more work will be carried out in the future to increase quality of the surface reflectance parameters. This will mainly target the accuracy of the 3D reconstruction that would allow better estimation of the surface normals and finally the consideration of the specular components.

## Acknowledgements

I would like to thank my colleagues Susannah Fleming and Lloyd Lukama for their work on camera calibration.

## References

1. Debevec, P.E.: Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In: Proceedings of SIGGRAPH 98, Computer Graphics Proceedings, Annual Conference Series, Orlando, USA (1998) 189–198

2. Sato, Y., Wheeler, M.D., Ikeuchi, K.: Object shape and reflectance modeling from observation. *Computer Graphics* **31**(Annual Conference Series) (1997) 379–388
3. Marschner, S.R., Westin, S.H., Lafortune, E.P.F., Torrance, K.E., Greenberg, D.P.: Image-based brdf measurement including human skin. In: *Proceedings of 10th Eurographics Workshop on Rendering*, Granada, Spain (1999) 139–152
4. Lensch, H., Kautz, J., Goesele, M., Heidrich, W., Seidel, H.P.: Image-based reconstruction of spatial appearance and geometric detail. *ACM Transactions on Graphics* **22**(2) (2003) 234–257
5. Debevec, P., Hawkins, T., Tchou, C., Duiker, H., Sarokin, W., Sagar, M.: Acquiring the reflectance field of a human face. In: *Proceedings of SIGGRAPH 2000, Computer Graphics Proceedings, Annual Conference Series*. (2000)
6. Grau, O., Pullen, T., Thomas, G.A.: A combined studio production system for 3-d capturing of live action and immersive actor feedback. *IEEE Transactions on Circuits and Systems for Video Technology* **14**(3) (2004) 370–380
7. Potmesil, M.: Generating octree models of 3D objects from their silhouettes in a sequence of images. *Computer Vision, Graphics and Image Processing* **40** (1987) 1–29
8. Szeliski, R.: Rapid octree construction from image sequences. *CVGIP: Image Understanding* **58**(1) (1993) 23–32
9. Grau, O.: 3d sequence generation from multiple cameras. In: *Proc. of IEEE, International workshop on multimedia signal processing 2004*, Siena, Italy (2004)
10. Grossberg, M., Nayar, S.: Modeling the Space of Camera Response Functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **26**(10) (2004) 1272–1282
11. Debevec, P., Malik, J.: Recovering high dynamic range radiance maps from photographs. In: *Proceedings of SIGGRAPH 97*. (1997) 369–378
12. Grau, O.: Capturing of light and surface reflection properties for re-lighting. <http://www.bbc.co.uk/rd/projects/virtual/lightcapture> (2006)

# A Multiple Graph Cut Based Approach for Stereo Analysis

Ulas Vural and Yusuf Sinan Akgul

GIT Vision Lab, Department Of Computer Engineering  
Gebze Institute Of Technology, Cayirova, Gebze, Kocaeli 41400, Turkey  
{uvural, akgul}@bilimuh.gyte.edu.tr

**Abstract.** This paper presents an optimization framework for the 3D reconstruction of the surfaces from stereo image pairs. The method is based on employing popular graph cut methods under the dual mesh optimization technique. The constructed system produces noticeably better results by running two separate optimization processes that communicate with each other. The communication mechanism makes our system more robust against local minima and it produces extra side information about the scene such as the unreliable image sections. We validated our system by running experiments on real data with ground truth and we compared our results with the other optimization methods, which showed the accuracy and effectiveness of our method.

## 1 Introduction

The classical breakdown of 3D surface recovery from stereo suggests that first the correspondences between the image pairs should be established and then the 3D surface is reconstructed using these correspondences[9]. The newer techniques take the approach of a global solution by incorporating the correspondence and the 3D reconstruction steps into the same process. This process is larger and more complex but the results are far better than the classical methods if the problem complexity is addressed properly. One common method to manage this larger problem is to pose it as an energy optimization task. An energy functional that penalizes locally unsmooth and discontinuous 3D structure is formulated. Optimization of this functional on the stereo image pairs would produce the desired 3D surface. Despite the elegance and unified nature of such systems, optimizing these functionals are not trivial. The problem is fundamentally NP-Hard and the approximation methods are sensitive to initializations, local minima, and image noise.

Recently, graph cut methods gained popularity in optimizing energy functionals of Computer Vision problems. Graph cuts can guarantee optimal functional values for some restricted cases[10][8]. For the other cases, they guarantee an upper bound in error from the optimal result[6]. Furthermore, since they are based on the deep theory of graph and flow algorithms, there are numerically stable and efficient algorithms for performing cuts[4]. Although the types of energy functionals that can be optimized by graph cuts are limited[13][7], the limitations are

not very restrictive. As a result, graph cuts were applied to many stereo problems including multi-camera scene reconstruction[12], occlusion detection[11], and stereo with plane fitting and layering[3].

Energy optimization with the dual mesh approach was proposed for depth estimation from stereo pairs[2] and tracking of ultrasound tongue sequences[1]. The dual mesh method is a framework that employs two instances of a known energy optimization method. It works on the principle of two simultaneous and interacting optimization processes. The energy optimizations start from the two ends of the search space and the optimizations continue until they find the same position in the search space. The interaction between the optimization processes is used to force the mesh with the high energy towards the other. The dual mesh method was shown to be relatively insensitive to local minima due to its two-way sweep of the search space. It does not have any initialization problems. However, the system can only be used for continuous depth recovery and it might be sensitive to local minima depending on the optimization methods used.

In this paper, we describe a system that uses graph cut energy optimization methods under the framework of dual mesh optimization. The system introduces a number of novel enhancements to both graph cuts and dual mesh framework to achieve a noticeably better energy optimization which results in more accurate 3D surface recovery. The system eliminates the dependency on the initial configuration because the initializations are done exactly the same way for all system input. The proposed system produces other important side information about the 3D scene such as the unreliable image parts for 3D reconstruction without any additional computational load.

Section 2 formally introduces the dual mesh energy. The details of graph cut optimization under the dual mesh framework is explained in Section 3. Experiments and validation work is discussed in Section 4. Section 5 concludes the paper.

## 2 The Dual Mesh Energy

A deformable mesh is a set of horizontally and vertically connected points in 3D space. Each point  $m_{ij}$  is a mesh element and the mesh elements form a 3D surface or set of 3D surfaces. The mesh elements have fixed  $x$  and  $y$  positions. The  $z$  positions of the mesh elements can change. The  $z$  position of a mesh element  $m_{ij}$  is also called the depth value of the element and it is given by the function  $depth(m_{ij})$ . A deformable mesh is positioned in a 3D volume and it interacts with the contents of the volume to localize any desirable 3D surface while maintaining the surface properties such as local continuity and smoothness. The movements of the deformable mesh is governed by an energy functional, which forces the deformable mesh to move towards the 3D surface positions that overlap with the existing real world surfaces. For our system, the energy functional of the deformable mesh  $M$  is dependent on the mesh  $N$  and it is written as

$$E_{Mesh}(M, N) = \sum_{i=1} \sum_{j=1} E_{Data}(m_{ij}) + E_{Smoothness}(m_{ij}) + E_{Tension}(m_{ij}, n_{ij}) \quad (1)$$



The term  $E_{Smoothness}(m_{ij})$  is for satisfying the smoothness constraint of the mesh. Regularization based approaches or using convex functions as smoothness term extend smoothness everywhere. Although these kinds smoothness terms makes the resulting systems more efficient and robust against noise, they do not work well at the object boundaries. For example, [2] can only recover continuous surfaces because it uses such a smoothness term. Non-convex functions can preserve discontinuities but they are more sensitive to local minima and optimizing such functions require more computational power.

Potts style smoothness terms are very simple but effective. They preserve discontinuities and their computational complexity is fair. We use a Potts style smoothness term to keep the discontinuities with a four-neighborhood system.

$$E_{Smoothness}(m_{ij}) = V(m_{ij}, m_{i+1j}) + V(m_{ij}, m_{ij+1}) \tag{2}$$

where

$$V(m_{ij}, m_{kl}) = \begin{cases} 0 & \text{depth}(m_{ij}) = \text{depth}(m_{kl}), \\ \lambda_1 & |\text{depth}(m_{ij}) - \text{depth}(m_{kl})| \leq \text{thresh}_1, \\ \lambda_2 & \text{otherwise.} \end{cases}$$

The tension energy is not always active. It is a mechanism that the dual mesh optimization framework employs to communicate information between the two separately deforming meshes. Under this framework, the functionals of two meshes are optimized separately at different initial 3D positions and it is expected that local minima, occlusions, and discontinuities will prevent them finding the same position at the end of the optimizations. When this happens, the tension term is activated to push the mesh with the worse position towards the other mesh. The details of this term is explained in section 3. We again use a Potts based model for the tension term.

$$E_{Tension}(m_{ij}, n_{ij}) = \begin{cases} 0 & \text{depth}(m_{ij}) = \text{depth}(n_{ij}) \\ \infty & \text{depth}(m_{ij}) > \text{depth}(n_{ij}) \\ \lambda_3 * (\text{depth}(n_{ij}) - \text{depth}(m_{ij})) & (\text{depth}(n_{ij}) - \text{depth}(m_{ij})) \leq \text{thresh}_2, \\ \lambda_3 * \text{thresh}_2 & \text{otherwise.} \end{cases}$$

Note that due to the working mechanism of the dual mesh framework, the elements  $n_{ij}$  of mesh  $N$  normally cannot have depth values less than the depth values of  $m_{ij}$  of mesh  $M$ .

Data energy is the term responsible for the deformation of the deformable mesh with the scene data. If the stereo pair is viewing a volume  $V$  (Fig. 1), and if the point  $P$  in this volume is visible from both cameras, then classical stereo analysis states that the image points  $p_l$  and  $p_r$  on the left and right images should belong to similar image regions. Therefore, the Sum of Squared Differences (SSD) between the local neighborhoods around the corresponding

image points is a good data energy term. For a given mesh element  $m$  at 3D point  $P$  in volume  $V$  (Fig. 1), the data energy term is written as

$$E_{Data}(m) = \sum_i \sum_j (IL_{ij} - IR_{ij})^2, \quad (3)$$

where  $IL_{ij}$  and  $IR_{ij}$  are the left and right image neighborhoods around the image points  $p_l$  and  $p_r$  of point  $P$ . Note that  $m$  does not have to be on a real surface in volume  $V$ . For any 3D position inside  $V$  the above data term can be calculated. If  $m$  is on a real world 3D surface, it is expected that the data energy term stays smaller.

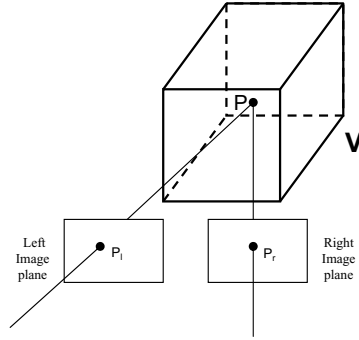


Fig. 1. A volume  $V$  is viewed by a stereo camera system

### 3 Graph Cut Based Dual Mesh Optimization for 3D Surface Recovery

Classically, stereo can be viewed as the problem of assigning a depth value label for each pixel  $p$  in the images. The depth value labels are chosen to be integers representing the distance between the image planes and the 3D point whose projection on the image plane is the pixel  $p$  (Fig. 1). There are only a finite number of labels. Therefore, we can formally define this labeling in terms of depth value sets. Let  $L$  and  $R$  be two sets of pixels in the left and right images respectively on the same epipolar line pair, and let  $D$  be the set of possible depth values. For any pixel  $p_l$  in set  $L$  there is a corresponding depth value label in set  $D$  that ties  $p_l$  to the pixel  $p_r$  in set  $R$ , which contains only the epipolar conjugate pixels of  $L$ . Note that choosing a label from set  $D$  for a pixel  $p_l$  is equivalent to choosing a corresponding pixel  $p_r$  for  $p_l$  and vice versa. A labeling  $f$  represents the complete matching of all pixels in the set  $L$  to their labels in the set  $D$ .

It is possible to minimize only the data energy term (Equation 3) to find the  $D$  labels by choosing the  $p_l$  and  $p_r$  pixels that maximize the data energy term. This mapping would be in polynomial time complexity. However, this mapping would also produce a rough depth map even at good image regions because of the very local decision base. We therefore need to figure out a more global formulation for the labeling problem.

### 3.1 Graph Cut Optimization

A more robust labeling can be achieved with the help of the deformable mesh structure defined in Section 2. The elements of the mesh would represent the pixels and their depths in a stereo image. Therefore, the  $x$  and  $y$  positions of the mesh elements cannot change. The  $z$  position of the mesh would represent the estimated depth value of the pixel. The labels that minimize the associated mesh energy would produce a 3D surface or surface set that would satisfy smoothness and good SSD values between the pixel correspondences. The brute force implementation of the above deformable mesh optimization is NP-hard, hence an approximation method is required. Recently, graph cuts became popular in approximate function optimization after the introduction of  $\alpha$ -expansion algorithm[6], which proves the existence of an upper bound on error from the optimal result.

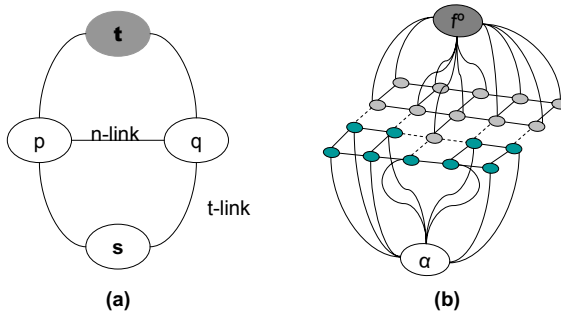
Graph cut based optimization methods need a special type of graph constructed first. We add a new node to the graph for each mesh element and since each mesh element represents a pixel, the special graph has a node for each pixel in the image. Two terminal nodes named the source ( $s$ ) and the sink ( $t$ ) are also added to the graph. Each node in the graph is connected to the terminal nodes with links called  $t$ -links. The weight of a  $t$ -link is chosen as the  $E_{Data}$  term of the mesh element corresponding to the node. Semantically, any graph node connected to  $s$  node has the depth label of  $s$ . Similarly, nodes connected to  $t$  node has the depth label of  $t$ . All pixel nodes are connected to their neighbors by n-links with the cost  $E_{Smoothness}$  of the pixel node(Fig. 2-a). The initial labeling of the graph is called  $f_0$ .

The  $\alpha$ -expansion algorithm uses  $z$  position values of the volume as the depth labels. For each depth label, a new graph is constructed with the sink node representing the current configuration and the source node representing the new depth label. Note that every time the graph is reconstructed, weights of the links are recalculated. The  $\alpha$ -expansion algorithm performs an  $s-t$  cut on this graph by using one of the max flow/min cut algorithms from the literature. After the partitioning, the nodes will have only one t-link, which means that each node will have only one label(Fig. 2-b). In other words, after one  $\alpha$ -expansion some mesh elements can change their labels to  $\alpha$ , and the others will keep their labels. Once all depth labels are tried as the source node, an iteration is completed. The  $\alpha$ -expansion continues with other iterations until there is no improvement in the total energy.

The above method was shown to be a very effective approximation method in assigning labels to pixels when there is a special type of energy functional involved. However, since it is still an approximation method, a better approximation is always helpful for a number of applications including stereo analysis. The next section explains how we use the dual mesh framework to achieve a better approximation.

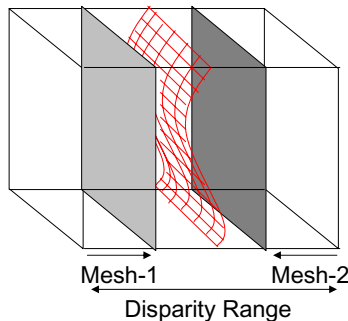
### 3.2 Graph Cuts Under Dual Mesh Framework

Dual mesh optimization framework[2,1] was originally developed for finding the approximate optimal values of contour positions in ultrasound or depth labels



**Fig. 2.** (a) The  $\alpha$ -expansion special graph with two nodes  $p$  and  $q$ . (b) The new labeling of the nodes after one  $\alpha$ -expansion step.

in stereo. The basic idea of dual mesh optimization is to pose the problem as a label assignment problem for each pixel or contour element. The continuity and smoothness of the labels are satisfied by optimizing an energy functional of a deformable mesh that assigns a depth position for each mesh element. The main argument of dual mesh approach is to employ two separate deformable mesh structures and initialize them at the opposite ends of the search space of the labels. By employing known optimization methods from the literature, the mesh energies are minimized separately and they start deforming independently (Figure 3). The deformable meshes usually stop deforming at different 3D positions due to local minima, which is a common problem in optimizing complex energy functionals. The dual mesh framework addresses the local minima problem by pushing the deformable mesh with the larger energy towards the other mesh. The biggest advantage of the dual mesh structure is that it can employ any optimization method to optimize each deformable mesh and the resulting 3D mesh positions will be better than what that specific optimization method can achieve.



**Fig. 3.** Two deformable mesh structure localizing the same 3D surface

We borrow the idea of the dual mesh framework to use it with graph cut based optimization methods. The dual mesh approach provides a facility for the graph cut method to improve the results by showing a better direction of

deformations which cannot be achieved by the graph cut method itself. The proposed optimization method stages are as follows:

- Start two graph cut optimization processes to minimize the dual mesh energies defined by Equation 1. The first optimization will use the maximum possible depth values as the initial label (mesh  $N$ ). The other optimization will use the minimum possible depth values as the initial label (mesh  $M$ ). The  $E_{Tension}$  term of the energy will not be used at this phase.
- After the optimization, take the final labeling of the corresponding mesh elements  $m_{ij}$  and  $n_{ij}$  in the two meshes and activate the  $E_{Tension}$  term for each mesh element that has different labeling in the other mesh. The  $E_{Tension}$  element is activated by adding its cost to the  $t - link$  of the mesh element.
- The above step will bias the mesh elements of the two meshes to find the same depth positions. However, it is not guaranteed that they will find the same positions due to other energy terms.
- If all of the two mesh elements find the same depth positions, then the optimizations ends. Otherwise, we deactivate the  $E_{Tension}$  term and start doing the same steps until convergence occurs or we see no improvement in the overall mesh energy.
- At the end of this process, the final labels are assigned to each stereo image pixels. If the corresponding mesh elements  $m_{ij}$  and  $n_{ij}$  have the same depth label, then pixel  $p_{ij}$  is assigned the same label. Otherwise, pixel  $p_{ij}$  gets the label of the smaller energy element.

The above procedure has several advantages. First, when compared to the graph cut methods, it produces noticeably closer results to the optimal value. This advantage is expected because we compare two almost identical graph cut optimization processes to bias the estimations towards the better one. The second advantage of this method is that, the corresponding mesh elements  $m_{ij}$  and  $n_{ij}$  that do not find the same depth positions would give us valuable information about the scene. These kinds of positions are actually problematic for stereo analysis because they correspond to occlusions, depth discontinuities, or textureless image areas. This information is very important in knowing what depth estimation values are more reliable than the others. Finally, unlike the original dual mesh method, our new method allows recovery of discontinuous 3D surface patches due to the employment of the Potts style smoothness function.

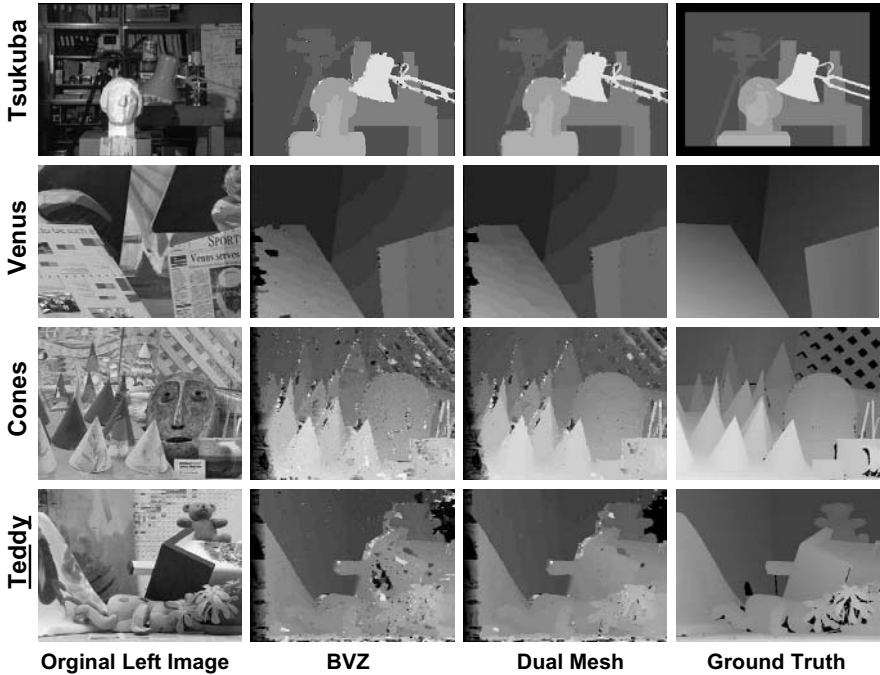
## 4 Experiments and System Validation

We implemented our system by using the graph cut library provided by [13]. We tested our system exhaustively to observe its performance in real world against the other methods and to validate the claims we made. For these experiments, we employed BVZ algorithm [5] as the underlying graph cut method of the dual mesh framework for its simplicity, though we could have used any graph cut method. For all the experiments, we used the stereo data and the ground truth provided by the Middlebury image base[14].

There are three main experiments we performed. First, we compared our results with the popular graph cut systems, BVZ[5], KZ1[12], and KZ2[11]. Table 1 shows that our method is always better than the BVZ, which is the underlying graph cut algorithm for our method. In some cases, the errors get very close to KZ1 and KZ2 algorithms which are much more sophisticated than the BVZ algorithm, which is very encouraging. Note that it is not fair to compare our dual mesh method to KZ1 and KZ2 methods directly because dual mesh method is dependent on the BVZ optimization method. We provided the numbers for the other methods just to show the scale of the difference between methods. We are

**Table 1.** Comparison of the dual mesh algorithm with other graph cut based algorithms. The numbers are percentage errors compared to ground truth on non-occluded, discontinuous, and all image regions.

Algorithm	Tsukuba			Venus			Teddy			Cones		
	Non-Occ	All	Disc	Non-Occ	All	Disc	Non-Occ	All	Disc	Non-Occ	All	Disc
BVZ	1.96	4.20	9.71	2.03	3.69	12.1	17.3	25.8	28.8	19.2	28.3	25.7
KZ1	1.83	2.48	6.42	1.06	1.52	5.53	12.0	17.9	22.4	5.78	12.9	13.2
KZ2	1.33	2.15	6.94	1.22	1.78	5.99	12.5	18.8	22.1	6.08	13.2	13.3
Dual Mesh	1.91	4.13	9.50	1.64	3.29	10.5	13.0	21.9	25.2	9.37	19.6	17.5



**Fig. 4.** The dual mesh disparity values compared with the ground truth and BVZ

working on implementing the KZ1 and KZ2 based dual mesh algorithms and we expect that such systems would produce better results than the KZ1 and KZ2 systems. Figure 4 shows the obtained disparity images from our system compared to BVZ method and the ground truth for the images of Table 3. Visual inspection of this data shows the accuracy and effectiveness of our system especially in images with discontinuities, occlusion and textureless regions. Note that we used exactly the same parameter set as the BVZ method when we compare our dual mesh method to BVZ in all the experiments.

For the second experiment, we tested the dual mesh capability of capturing the unreliable image areas. Textureless, occluded and discontinuous regions are problematic parts of the images for stereo analysis. Knowing such regions would make the subsequent processing more convenient. Dual mesh structure can be used for partial detection of these areas by checking the intermediate positions of the mesh elements during the optimization. If there are mesh elements  $m_{ij}$  and  $n_{ij}$  that do not find the same positions, then we mark these areas as problematic areas. Table 2 shows the percentage of the problematic pixels detected and the overlap of these pixels with the occluded image regions from the ground truth of the Tsukuba image. Notice that 38% of the problematic pixels are occluded. We visually verified that the rest of the problematic pixels are from textureless regions and depth discontinuities. We are working on producing the ground truth data to quantitatively verify this claim.

For the third experiment, we like to show that the results obtained by the dual mesh method cannot be obtained by a single optimization process. We run the BVZ method with the random labeling as suggested by [5]. Due to the randomness factor, we repeated the run 10 times and recorded the best, the worst and the average errors. We also modified the BVZ method so that it takes

**Table 2.** Detection of unreliable pixels. Occluded pixels are obtained from ground truth.

	Number	Percentage
Pixels	109921	100.0
Problematic Meshels of Dual Mesh	1312	1.19
Problematic And Occluded	504	0.45

**Table 3.** The effects of using different labeling methods

Algorithm	Non-occluded	All	Discontinuity
	Areas	Areas	Areas
BVZ Random (Best)	18.9	28.0	25.5
BVZ Random (Avg)	19.3	28.4	25.7
BVZ Random (Worst)	19.6	28.6	26.1
BVZ Regular-1 (One Mesh)	19.0	28.1	25.4
BVZ Regular-2 (One Mesh)	18.9	28.0	25.5
Dual Mesh	9.37	19.6	17.5

the same labeling order as our dual mesh labelings. Since we have two separate optimizations, there are two different labeling orders(regular-1 and regular-2). Table 3 shows the percentage errors from each run compared to our dual mesh on the Cones image. We obtained similar results on other images. As the results clearly show, the dual mesh approach produces better results.

## 5 Conclusions

Graph cut methods gained popularity in optimizing energy functionals of Computer Vision problems due to their effectiveness and closeness to the optimality. We presented an optimization framework that noticeably improves the graph cut based stereo methods. The system is based on deformable dual mesh optimization that employs two graph cut optimization processes. The processes communicate with each other to come up with a better 3D surface that cannot be achieved by a single optimization. Furthermore, the communication mechanism makes our system more robust against local minima. The method can also extract other useful information about the scene such as the unreliable image sections. Although we employed the BVZ algorithm as the underlying graph cut method, our framework can employ any graph cut methods to improve the results.

We validated the system by running several experiments and compared our results with the ground truth and other stereo algorithms. We quantitatively observed that our method noticeably improves the graph cut optimization results. We also visually observed the improvements. Overall, the results are very encouraging.

The system is open to further enhancements. We are working on implementing more sophisticated graph cut algorithms to be used as optimization methods. We are also working on making the system computationally more efficient by using a tighter communication mechanism between the meshes. It is also planned to use this system as a general label assignment method for image/volume segmentation, contour tracking, and motion analysis.

## Acknowledgements

This work is supported by TUBITAK Career Project 105E097.

## References

1. Y. Akgul, C. Kambhamettu, and M. Stone. A task-specific contour tracker for ultrasound. In *IEEE Workshop on Mathematical Methods in Biomedical Image Analysis*, 2000.
2. Yusuf Sinan Akgul and Chandra Kambhamettu. Recovery and tracking of continuous 3d surfaces from stereo data using a deformable dual-mesh. In *International Conference on Computer Vision*, pages 765–772, 1999.
3. M. Bleyer and M. Gelautz. A layered stereo matching algorithm using image segmentation and global visibility constraints. *ISPRS Journal of Photogrammetry and Remote Sensing*, 59(3):128–150, May 2005.



4. Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. In *EMMVCVPR02*, page 359 ff., 2002.
5. Y. Boykov, O. Veksler, and R. Zabih. Markov random fields with efficient approximations. In *IEEE Computer Vision and Pattern Recognition or CVPR*, pages 648–655, 1998.
6. Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222 – 1239, Nov 2001.
7. D. Freedman and P. Drineas. Energy minimization via graph cuts: settling what is possible. In *IEEE Computer Vision and Pattern Recognition*, pages II:939–946, 2004.
8. D. Greig, B. Porteous, and A. Seheult. Exact maximum a posteriori estimation for binary images. *Journal Royal Statistical Society, B*: 51(2):271–279, 1989.
9. B.K.P. Horn. *Robot Vision*. The MIT Press, 1986.
10. H. Ishikawa. Exact optimization for markov random fields with convex priors. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 25(10):1333–1336, October 2003.
11. V. Kolmogorov and R. Zabih. Computing visual correspondence with occlusions via graph cuts. In *International Conference on Computer Vision*, pages II: 508–515, 2001.
12. V. Kolmogorov and R. Zabih. Multi-camera scene reconstruction via graph cuts. In *European Conference on Computer Vision*, page III: 82 ff., 2002.
13. V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):147–159, 2004.
14. D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1-3):7–42, April 2002.

# Robust Variational Segmentation of 3D Objects from Multiple Views

Kalin Kolev, Thomas Brox, and Daniel Cremers

CVPR Group, University of Bonn  
Römerstr. 164, 53117 Bonn, Germany  
{kolev, brox, dcremers}@cs.uni-bonn.de

**Abstract.** We propose a probabilistic formulation of 3D segmentation given a series of images from calibrated cameras. Instead of segmenting each image separately in order to build a 3D surface consistent with these segmentations, we compute the most probable surface that gives rise to the images. Additionally, our method can reconstruct the mean intensity and variance of the extracted object and background. Although it is designed for scenes, where the objects can be distinguished visually from the background (i.e. images of piecewise homogeneous regions), the proposed algorithm can also cope with noisy data. We carry out the numerical implementation in the level set framework. Our experiments on synthetic data sets reveal favorable results compared to state-of-the-art methods, in particular in terms of robustness to noise and initialization.

## 1 Introduction

Recovering the spatial structure of a scene from multiple views is one of the oldest and most fundamental problems in computer vision with many applications in computer graphics, robot navigation, object recognition, and tracking. The literature on 3D reconstruction could be divided into four major classes: shape from stereo, shading, texture, and silhouettes.

Stereovision requires to match points from different images that correspond to the same point in the scene. The earliest algorithms that incorporate a large number of views use carving techniques to obtain a volumetric representation of the scene assuming Lambertian properties of the objects [18,10]. The space carving framework suffers from several limitations. Once a voxel is carved away, it cannot be recovered. Moreover, if one voxel is removed in error, further voxels can be erroneously removed in a cascade effect. These limitations are partially alleviated by the probabilistic space carving method [1]. Others have suggested to guide a deformable surface model by a measure based on local correspondences toward a steady state [6,5]. All these methods require a textured surface in order to match points.

Shape from shading methods, on the other hand, are mainly designed for homogeneous objects [8,9]. They are based on the diffusing properties of Lambertian surfaces and aim at reconstructing the object shape from light reflectance.

A difficulty of this concept is the requirement of a known illumination model or the necessity to estimate illumination together with the shape.

A similar problem appears with texture-based methods [12]. They need a known texture pattern in order to reconstruct a 3D surface by means of its distortion in the image.

In case of sparsely textured objects, which are known challenges to stereo- and texture-based techniques, silhouettes exhibit the dominant image feature. The algorithm presented in this paper belongs to this type of silhouette-based techniques. Such methods usually try to estimate the *visual hull* of the observed objects. The visual hull of an object is defined as the maximal shape that yields the same silhouette as the observed object [11]. The earliest attempts use a volumetric representation of the scene and are referred to as *volume intersection* techniques in the literature. That is, the space is discretized by a fixed voxel grid and each voxel is labeled as opaque or transparent. An early paper reporting a volumetric representation of the visual hull is due to Martin and Aggarwal [13]. They segment the input images in advance by a simple intensity thresholding and then back-project the estimated silhouettes to a surface representation. Since then, silhouettes have been used in many different algorithms. Octree-based representations have been employed by [15,19,7], and in [17] the authors presented a Hough-like voting scheme that back-projects image features into a volumetric space. In addition to volumetric approaches, some surface-based ones have been presented. In [3] and [20] apparent contours are used to reconstruct a 3D shape. Although the authors obtain better results, the reconstruction works only locally.

Yezzi and Soatto recently proposed *stereoscopic segmentation* as a variational framework for global 3D region segmentation from a collection of images of a scene [21]. They couple the segmentations of each image through the evolution of a single 3D surface rather than separate 2D contours, which makes their method robust to erroneous camera calibration. Upon a closer look, it turns out that stereoscopic segmentation has certain limitations. Its main drawback is the definition of the energy in the image domain that results in a very local evolution. Consequently, it needs an accurate initialization in order to capture the correct object topology. In addition, the algorithm is prone to noise as the strictly local surface evolution is mainly determined by single camera observations.

In this paper, we propose a probabilistic Bayesian formulation of 3D reconstruction which aims at estimating the most likely 3D shape given the observed images. In contrast to stereoscopic segmentation, this yields a more global evolution that makes better use of the available information from multiple cameras. As a consequence, our method has a larger radius of convergence and is more robust to noise than previous techniques.

**Paper organization.** In the next section, the probabilistic framework of the proposed method is presented and discussed. A variational formulation and a respective level set implementation are developed in Section 3. In Section 4 we show experimental results. Finally, we provide a conclusion in Section 5.

## 2 Probabilistic Volume Intersection

### 2.1 Bayesian Inference

Let  $V$  be a discretized volume and  $I_1, \dots, I_n : \Omega \mapsto \mathbb{R}$  a collection of calibrated input images with perspective projections  $\pi_1, \dots, \pi_n$ . Given the set of images, we are looking for the most probable surface  $\hat{S}$  that gives rise to these images, that is

$$\hat{S} = \arg \max_{S \in \Lambda} P(S | \{I_1, \dots, I_n\}), \tag{1}$$

where  $\Lambda$  is the set of all closed surfaces lying inside of the volume  $V$ . By means of the Bayes formula we obtain (omitting the normalization constant):

$$P(S | \{I_1, \dots, I_n\}) \propto P(\{I_1, \dots, I_n\} | S) \cdot P(S). \tag{2}$$

Assuming that all voxels are independent leads to

$$P(S | \{I_1, \dots, I_n\}) \propto \left[ \prod_{x_{ijk} \in V} P(\{I_l(\pi_l(x_{ijk}))\}_{l=1, \dots, n} | S) \right]^{dx} \cdot P(S), \tag{3}$$

where  $dx$  denotes the discretization step. The exponent  $dx$  is introduced to ensure the correct continuum limit. The resulting expression is then invariant to refinement of the grid.

According to a certain surface estimate  $S$ , the voxels can be divided into two classes: lying inside an object or belonging to the background. Hence, the volume  $V$  can be expressed as  $V = R_{obj}^S \cup R_{bck}^S$ . Considering this partitioning, we can proceed with

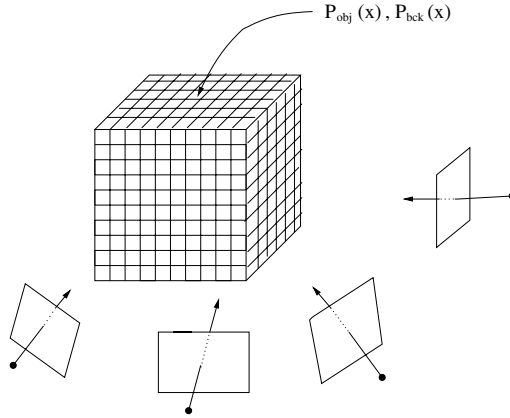
$$P(S | \{I_1, \dots, I_n\}) \propto \left[ \prod_{x_{ijk} \in R_{obj}^S} P(\{I_l(\pi_l(x_{ijk}))\}_{l=1, \dots, n} | x_{ijk} \in R_{obj}^S) \right]^{dx} \cdot \left[ \prod_{x_{ijk} \in R_{bck}^S} P(\{I_l(\pi_l(x_{ijk}))\}_{l=1, \dots, n} | x_{ijk} \in R_{bck}^S) \right]^{dx} \cdot P(S).$$

To simplify the notation, we denote

$$\begin{aligned} P_{obj}(x) &:= P(\{I_l(\pi_l(x))\}_{l=1, \dots, n} | x \in R_{obj}^S) \\ P_{bck}(x) &:= P(\{I_l(\pi_l(x))\}_{l=1, \dots, n} | x \in R_{bck}^S) \end{aligned} \tag{4}$$

for  $x \in V$  (see fig. 1) and come to the following expression

$$\hat{S} = \arg \max_{S \in \Lambda} \left[ \prod_{x_{ijk} \in R_{obj}^S} P_{obj}(x_{ijk}) \cdot \prod_{x_{ijk} \in R_{bck}^S} P_{bck}(x_{ijk}) \right]^{dx} \cdot P(S). \tag{5}$$



**Fig. 1.** Volume representation. Two probabilities  $P_{obj}, P_{bck}$  are assigned to each voxel for membership to one of the objects and background, respectively.

### 2.2 Joint Probabilities

In order to compute the joint probabilities  $P(\{I_l(\pi_l(x))\}_{l=1,\dots,n} \mid x \in R_{obj}^S)$  and  $P(\{I_l(\pi_l(x))\}_{l=1,\dots,n} \mid x \in R_{bck}^S)$ , we have to combine information from different images. This could be achieved by assuming independence of the image observations yielding

$$\begin{aligned}
 P_{obj}(x) &= \prod_{i=1}^n P(I_i(\pi_i(x)) \mid x \in R_{obj}^S) \\
 P_{bck}(x) &= 1 - \prod_{i=1}^n [1 - P(I_i(\pi_i(x)) \mid x \in R_{bck}^S)].
 \end{aligned}
 \tag{6}$$

Note the asymmetry in these expressions. The probability of a voxel being part of the foreground is equal to the probability that all cameras observe this voxel as foreground, whereas the probability of background membership describes the probability of at least one camera seeing background. However, this model has some disadvantages. In case of noisy images  $0 < P(I_i(\pi_i(x)) \mid x \in R_{obj}^S) < 1$  and  $0 < P(I_i(\pi_i(x)) \mid x \in R_{bck}^S) < 1$ , in general. Hence, for  $n \rightarrow \infty$  the joint probability  $P(\{I_l(\pi_l(x))\}_{l=1,\dots,n} \mid x \in R_{obj}^S)$  will converge to 0 and  $P(\{I_l(\pi_l(x))\}_{l=1,\dots,n} \mid x \in R_{bck}^S)$  to 1. To dispose this bias for increasing number of cameras, we have to take the dependency of the observations into account. In our model we used the geometric mean of the single probabilities:

$$\begin{aligned}
 P_{obj}(x) &= \sqrt[n]{\prod_{i=1}^n P(I_i(\pi_i(x)) \mid x \in R_{obj}^S)} \\
 P_{bck}(x) &= 1 - \sqrt[n]{\prod_{i=1}^n [1 - P(I_i(\pi_i(x)) \mid x \in R_{bck}^S)]}.
 \end{aligned}
 \tag{7}$$

They are modeled by Gaussian densities

$$\begin{aligned} P(I_i(\pi_i(x)) \mid x \in R_{obj}^S) &= \frac{1}{\sqrt{2\pi} \cdot \sigma} \cdot e^{-\frac{(I_i(\pi_i(x)) - \mu_{obj})^2}{2\sigma^2}} \\ P(I_i(\pi_i(x)) \mid x \in R_{bck}^S) &= \frac{1}{\sqrt{2\pi} \cdot \sigma} \cdot e^{-\frac{(I_i(\pi_i(x)) - \mu_{bck})^2}{2\sigma^2}}, \end{aligned} \quad (8)$$

where  $\mu_{obj}$ ,  $\mu_{bck}$  denote the mean intensities of object/background and  $\sigma$  is the respective standard deviation. We update these values in the course of evolution by projecting the current surface estimate onto the images as described in [21]. The standard deviation  $\sigma$  is set to the maximum of the deviations of the object and background regions. Alternatively, above probabilities could be modeled with two separate standard deviations. However, in our experiments the proposed model resulted in a faster convergence.

### 3 Variational Framework

#### 3.1 Variational Formulation

In this section we will convert the maximum a-posteriori estimation into an energy minimization problem. Applying the negative logarithm to (5) yields in a continuous formulation the following functional:

$$E(S) = - \int_{R_{obj}^S} \log P_{obj}(x) dx - \int_{R_{bck}^S} \log P_{bck}(x) dx - \log P(S). \quad (9)$$

Minimizing this energy functional is equivalent to maximizing the total a-posteriori probability of all voxel assignments. The first two terms are related to the external energy and measure the discrepancy between observed images and images predicted by the model. The last term exhibits the internal energy and describes the surface shape, thus allowing incorporation of prior knowledge on the geometry. Note that the functional also incorporates the intensity means and standard deviation, which are defined by the surface  $S$ . Since the unknowns, surface and radiances, live in an infinite-dimensional space (there exist multiple solutions  $S$  that explain the observed images), we need to impose regularization in order to make the minimization problem well-posed. This can be achieved by setting

$$P(S) = e^{-\nu|S|}, \quad (10)$$

where  $\nu$  is a weighting constant and  $|S|$  denotes the surface area. Inserting this expression into the above functional yields

$$E(S) = - \int_{R_{obj}^S} \log P_{obj}(x) dx - \int_{R_{bck}^S} \log P_{bck}(x) dx + \nu|S|. \quad (11)$$

In order to reconstruct the smoothest surface consistent with the images, we omit the data fidelity terms for points, which are visible from neither of the cameras. This is not restrictive, since no data is available for such points.

### 3.2 Level Set Implementation

The numerical implementation of the proposed energy functional (11) has been carried out within the level set framework [4,14] due to its stability and ability to handle topological changes automatically. In level set methods, the surface is implicitly represented by a function  $\phi : V \mapsto \mathbb{R}$ , whose values are the distances from the surface, and the interior and exterior of the surface are defined by  $\phi(x) < 0$  and  $\phi(x) \geq 0$ , respectively. Hence, we can use the Heaviside function

$$H(z) = \begin{cases} 1, & \text{if } z \geq 0 \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

to access these two regions. Expressing the energy functional (11) with respect to the level set function  $\phi$  yields

$$E(\phi) = - \int_V [\log P_{bck}(x)H(\phi(x)) + \log P_{obj}(x)(1 - H(\phi(x)))] dx + \nu \int_V |\nabla H(\phi(x))| dx. \tag{13}$$

This formulation has some nice properties. First, its Euler-Lagrange equations are easy to compute since the implicit function  $\phi$  occurs as an argument. Second, it leads to a stable volume-based surface flow. A similar energy functional was used in [2,16] for image segmentation purposes. The Euler-Lagrange equations of (13) read

$$\frac{\partial \phi(x)}{\partial t} = \delta(\phi(x)) \cdot [\log P_{bck}(x) - \log P_{obj}(x)] + \nu \delta(\phi(x)) \cdot \text{div} \left( \frac{\nabla \phi(x)}{|\nabla \phi(x)|} \right), \tag{14}$$

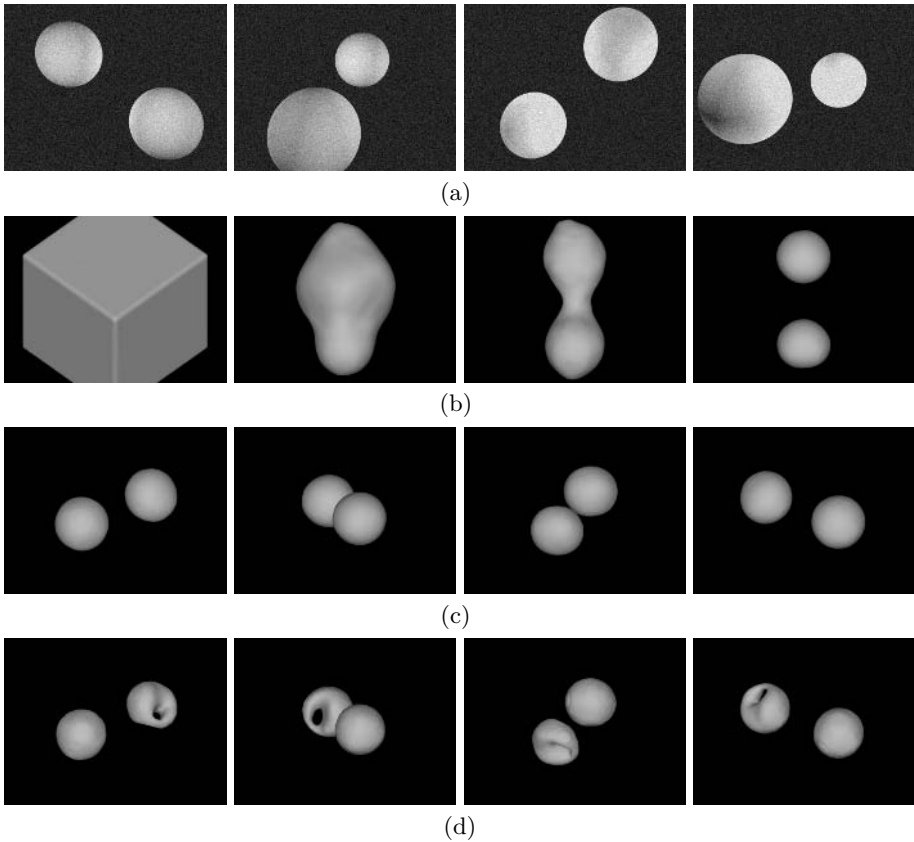
where  $\delta(\cdot)$  denotes the Dirac function

$$\delta(z) = \frac{d}{dz} H(z). \tag{15}$$

In practice, smoothed versions of  $H(\cdot)$  and  $\delta(\cdot)$  have to be applied [2].

## 4 Experiments

In Fig. 2 we show results obtained with the proposed algorithm applied to 20 noisy images, four of which are depicted in Fig. 2(a). Fig. 2(c) visualizes the final result from multiple viewing directions. Obviously, our method is able to deal with noise as well as lighting effects and leads to an accurate reconstruction of the two balls. In order to emphasize its robustness a reconstruction generated by carving techniques is presented for comparison. For the sake of fairness we added an identical smoothness term in the implementation of the shape carving method. The estimated mean intensities computed by our algorithm were used for segmenting the input images separately and independently. As clearly visible in Fig. 2(d), this approach is susceptible to noise and shading effects, since only

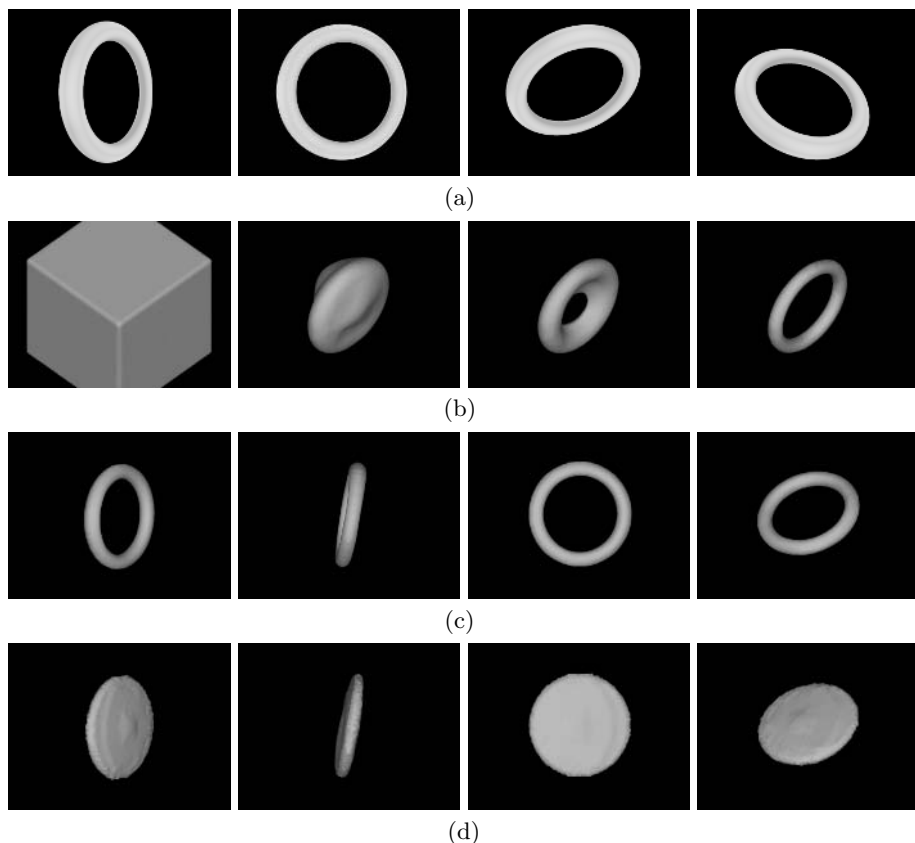


**Fig. 2.** Reconstruction of two spheres. (a) 4 out of 20 input images disturbed by noise, (b) surface during evolution, (c) reconstructed surface obtained with our probabilistic method, (d) result obtained with carving techniques.

single observations are taken into account for deciding whether a voxel should be carved away or not. In contrast, our method is quite robust to noise due to the averaging effect of integrating data from all views.

Fig. 3 demonstrates the ability of the proposed method to reconstruct complex topologies starting with an arbitrary initialization as opposed to stereoscopic segmentation, which requires an approximation of the real topology, as stated in [21]. The reconstructions of a torus obtained with our method and with stereoscopic segmentation from the same initial surface are depicted in Fig. 3(c) and Fig. 3(d), respectively. Note that, similar to stereoscopic segmentation, our method is bidirectional, i.e., surfaces can evolve inward as well as outward. In addition, our formulation leads to a surface evolution that allows for bigger time steps. In contrast to stereoscopic segmentation, the time step size is only restricted by the smoothness constraint.

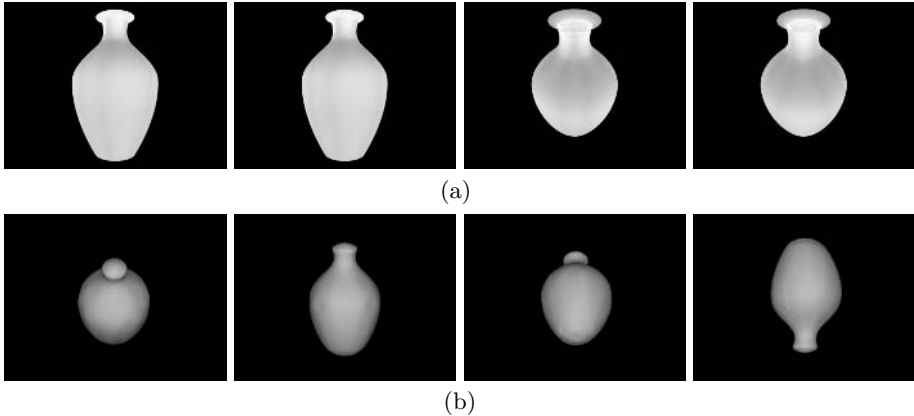




**Fig. 3.** Reconstruction of a torus. (a) 4 out of 20 input images, (b) surface during evolution, (c) reconstructed surface obtained with our method, (d) result obtained with stereoscopic segmentation [21] from the same initialization.

Finally, Fig. 4 illustrates the behavior of the presented algorithm when applied to a data set that exhibits ambiguous silhouette information. The cameras are arranged in such a way that none of them can see the bottom of the vase. Due to the geometric prior, the lacking information results in the smoothest shape that is photometrically consistent with the data (note the flat bottom and the neck of the vase).

All illustrated results were obtained from 20 images with  $640 \times 480$  pixels using a C++ implementation running on a Pentium IV with 3.4GHz. All cameras were situated on a bounding sphere enclosing the scene. For a cubic grid of  $128 \times 128 \times 128$  the algorithm takes between 20 and 30 minutes to converge, which is about a factor 3 faster than stereoscopic segmentation. Moreover, it can still be substantially accelerated when replacing our preliminary surface projection algorithm by a more sophisticated implementation.



**Fig. 4.** Reconstruction of a vase. (a) 4 out of 20 input images. Due to the rotational symmetry and the arrangement of the cameras, most images look the same. (b) Reconstructed surface from multiple views.

## 5 Summary

We have presented a new variational approach to reconstruct smooth shapes from a number of calibrated camera views. The variational formulation is derived from a probabilistic setting via Bayesian inference and uses the level set framework to represent the sought object surface. The mean radiance of object and background are estimated together with the surface. In comparison to previous methods, the probabilistic derivation and formulation of the energy on the volumetric instead of the image domain provides faster convergence and better robustness to noise or other violations of the assumption of constant object radiance. Moreover, the optimization is less prone to accurate initializations and allows to reconstruct more complex topologies. These properties have been confirmed in experimental evaluation. Future work is focused on applications to real data sets.

## Acknowledgments

This work was supported by the German Research Foundation, grant #CR-250/1-1.

## References

1. A. Broadhurst, T. W. Drummond, and R. Cipolla. A probabilistic framework for space carving. In *Proc. International Conference on Computer Vision*, pages 388–393, July 2001.
2. T. Chan and L. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, Feb. 2001.

3. R. Cipolla and A. Blake. Surface shape from the deformation of apparent contours. *International Journal of Computer Vision*, 9(2):83–112, 1992.
4. A. Dervieux and F. Thomasset. A finite element method for the simulation of Rayleigh–Taylor instability. In R. Rautman, editor, *Approximation Methods for Navier–Stokes Problems*, volume 771 of *Lecture Notes in Mathematics*, pages 145–158. Springer, Berlin, 1979.
5. Y. Duan, L. Yang, H. Qin, and D. Samaras. Shape reconstruction from 3D and 2D data using PDE-based deformable surfaces. In *Proc. European Conference on Computer Vision*, pages 238–251, 2004.
6. O. Faugeras and R. Keriven. Variational principles, surface evolution, PDE’s, level set methods, and the stereo problem. *IEEE Transactions on Image Processing*, 7(3):336–344, Mar. 1998.
7. B. Garcia and P. Brunet. 3D reconstruction with projective octrees and epipolar geometry. In *Proc. International Conference on Computer Vision*, pages 1067–1072, January 1998.
8. B. Horn and M. Brooks. *Shape from shading*. MIT Press, 1989.
9. H. Jin, D. Cremers, A. Yezzi, and S. Soatto. Shedding light on stereoscopic segmentation. In L. Davis, editor, *Proc. International Conference on Computer Vision and Pattern Recognition*, volume 1, pages 36–42, Washington, DC, 2004.
10. K. N. Kutulakos and S. M. Seitz. A theory of shape by space carving. *International Journal of Computer Vision*, 38(3):199–218, 2000.
11. A. Laurentini. The visual hull concept for visual-based image understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2):150–162, 1994.
12. J. Malik and R. Rosenholtz. A differential method for computing local shape-from-texture for planar and curved surfaces. In *Computer Vision and Pattern Recognition Conference*, pages 267–273, June 1993.
13. W. N. Martin and J. K. Aggarwal. Volumetric descriptions of objects from multiple views. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(2):150–158, 1983.
14. S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton–Jacobi formulations. *Journal of Computational Physics*, 79:12–49, 1988.
15. M. Potmesil. Generating octree models of 3D objects from their silhouettes from a sequence of images. *Computer Vision, Graphics, and Image Processing*, 40(1):1–29, 1987.
16. M. Rousson, T. Brox, and R. Deriche. Active unsupervised texture segmentation on a diffusion based feature space. In *Proc. International Conference on Computer Vision and Pattern Recognition*, pages 699–704, Madison, WI, June 2003.
17. S. Seitz and C. Dyer. Complete scene structure from four point correspondences. In *Proc. International Conference on Computer Vision*, pages 330–337, June 1995.
18. S. Seitz and C. Dyer. Photorealistic scene reconstruction by voxel coloring. In *Proc. International Conference on Computer Vision and Pattern Recognition*, pages 1067–1073, June 1997.
19. R. Szeliski. Rapid octree construction from image sequences. *Computer Vision, Graphics, and Image Processing*, 58(1):23–32, 1993.
20. R. Vaillant and O. Faugeras. Using extremal boundaries for 3D object modelling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):157–173, 1992.
21. A. Yezzi and S. Soatto. Stereoscopic segmentation. *International Journal of Computer Vision*, 53(1):31–43, 2003.

# Online Calibration of Two Zoom-Pan-Tilt Units for Planar Dynamic Events

Kurt Cornelis, Nico Cornelis, Maarten Aerts,  
Egemen Özden, and Luc Van Gool

Katholieke Universiteit Leuven  
Kasteelpark Arenberg 10, B-3001 Heverlee, Belgium  
`firstname.lastname@esat.kuleuven.be`

**Abstract.** In many sports and surveillance scenarios the action is dynamic and takes place on a planar surface, while being recorded by two or more zoom-pan-tilt cameras. Although their position is fixed, these cameras can typically rotate and zoom independently from each other. When rotation and zoom of each camera are known, one could reconstruct the dynamic event in 3D and generate different views of the action. Sensors exist which report zoom and orientation changes of pan-tilt units. In absence of such sensors, however, we prove that the varying zoom and rotation of two pan-tilt units can be extracted solely from the planar homography which exists between both cameras.

## 1 Introduction

When one mentions cameras and planes, one immediately thinks of planar homographies relating corresponding points between the different images. In this theoretical paper, we look at the special case of two time-synchronized zoom-pan-tilt cameras which observe an unknown dynamic planar event. Due to the dynamic nature of the event, the only planar homography which can be computed is the one between images from both cameras taken at the same time. Based on a-priori knowledge of the fixed pivot point around which each pan-tilt camera rotates, we prove that one can extract from this single homography all remaining unknowns: the focal length and rotation of each camera. To our knowledge, this is the first time that information has been extracted from a homography under these conditions, which are very realistic conditions since most sports and surveillance scenarios use this setup.

Our main goal is to introduce the mathematics necessary to extract the information. The theory will be verified on an artificial sport scenario. Due to the limited extent of this paper, those experiments have been selected which already demonstrate certain important behavior of the method. Future work will definitely extend our experiments to more error analysis of artificial and real video data. However, at this point the experiments already underline the validity of the presented theory.

## 2 Previous Work

Many authors already took a closer look at how to extract data from planar homographies. The algorithms can be classified based on a-priori knowledge. For instance, [4] assumes scene structure and internal camera calibration to be known. A relaxation of these assumptions has been implemented by [5,7] when only scene geometry is known and the camera internals are recovered together with the unknown camera externals. The other option is to assume only the internal calibration to be given beforehand as in [8]. Triggs [6] even takes it a step further and assumes the internal calibration to be unknown but the same for every image. Finally, we have the methods which assume no prior knowledge on scene geometry nor on internal camera calibration. Of the latter category, [2] and [3] are fine examples of homography decomposition algorithms.

The method we suggest describes an important case which we thought was still missing. The case of partial knowledge on the external camera parameters, more specifically a-priori knowledge on the fixed translations of pan-tilt cameras. As mentioned above, this case is of great practical importance for sport and surveillance events which are typically covered using independently zooming cameras which rotate around fixed pivot points. Some of the other algorithms mentioned above could be used in this scenario as well. However, none of them exploit the fact that camera translations should remain fixed over time and therefore lead to solutions with varying translations which are intrinsically wrong. Furthermore, many of the previous works require more than one homography to be given in order to solve for the unknowns, either by assuming multiple scene planes or by moving a single plane with respect to the camera. In our case of dynamic events recorded by two cameras, there is only a single homography which can be computed at each time instant. We will prove that under the assumption of fixed camera translations it is possible to extract the varying focal length and rotation of both cameras from this single homography.

## 3 Our Notation

A planar homography transfers image correspondences between two images of a planar surface, taken (at the same time in the case of dynamic events) from two different camera positions. The planar homography  $\mathbf{H}_{21}$  which transfers a point from image 2 to its corresponding point in image 1, is a  $3 \times 3$  matrix which is built up as follows:

$$x_1 \sim \mathbf{H}_{21}x_2 \tag{1}$$

$$\mathbf{H}_{21} = \lambda \mathbf{K}_1 \mathbf{R}_1^T \mathbf{B} \mathbf{R}_2 \mathbf{K}_2^{-1} \tag{2}$$

in which  $x_i = [X_i \ Y_i \ 1]^T$  denotes the corresponding homogeneous point in image 1 and 2 respectively, and in which the symbol  $\sim$  implies that equation 1 is only defined up to scale. Due to this scale ambiguity a scalar variable  $\lambda$  is present in equation 2.  $\mathbf{K}_i$  represents the  $3 \times 3$  internal calibration matrix of image 1 and

image 2 respectively.  $\mathbf{R}_i$  is a  $3 \times 3$  orthogonal matrix describing the rotation of the respective camera in the chosen world coordinate system.  $\mathbf{B}$  is a  $3 \times 3$  matrix which encapsulates the a-priori information on the camera positions and the plane in the following fashion:

$$\mathbf{B} = \mathbf{I} + \frac{(t_1 - t_2)}{(t_2^T n - d)} n n^T \quad (3)$$

in which  $\mathbf{I}$  is the  $3 \times 3$  identity matrix, and  $t_i$  is a  $3 \times 1$  vector representing the 3D focal point position of camera 1 and 2 respectively. The  $3 \times 1$  vector  $n$  and scalar  $d$  describe the plane. World points belonging to this plane satisfy the following equation:  $[X \ Y \ Z]n - d = 0$ . To model the camera zoom explicitly, we opt for the following parameterization of the internal calibration matrices:

$$\mathbf{K}_i = \begin{bmatrix} f_x & s & u \\ 0 & f_y & v \\ 0 & 0 & 1 \end{bmatrix}_i \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & f_i \end{bmatrix} = \mathbf{K}_i^{fixed} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & f_i \end{bmatrix} \quad (4)$$

in which  $f_x$  and  $f_y$  represent the focal length expressed in pixels along the  $x$  and  $y$  direction of the image; in which  $s$  represents the pixel skew; in which  $(u, v)$  represents the principal point and  $f_i$  the relative zoom factor. The influence of  $f_i$  can be easily understood as a scaling of the image around the principal point.

## 4 Our Setup: Knowns and Unknowns

Here we describe our setup and mention which parameters are assumed to be known, and which still need to be solved for. We have two zoom-pan-tilt cameras which are positioned at different locations and observe the same dynamic event taking place on a planar surface. First of all, we assume that for both cameras the internal calibration matrix  $\mathbf{K}_i^{fixed}$  is known beforehand, e.g. through a single prior calibration using a calibration grid. The only remaining internal unknowns during the envisioned application are therefore the zoom factors  $f_i$  of both cameras. Subsequently, we choose our right-handed world coordinate system such that the plane normal  $n$  equals the world's Z-axis and the scalar  $d$  equals 0. The 3D position  $t_i$  of each camera in the world coordinate system is assumed to be known a-priori. To this end, any localization method or a computer vision algorithm such as [1] can be used. The latter assumes the knowledge of at least three known world points, which for standardized sports fields and other areas are not hard to come by. During the envisioned application we assume that the cameras can only rotate and zoom. Therefore,  $t_i$  is a constant which only needs to be determined once. The previous assumptions therefore result in a constant and known matrix  $\mathbf{B}$ . So in the end, there remain nine unknowns: the scaling factor  $\lambda$ , two zooming factors  $f_i$  and six variables coming from both unknown rotations  $\mathbf{R}_i$ . This exactly matches the amount of entries in a  $3 \times 3$  planar homography matrix. The counting argument therefore states that it could be possible to retrieve these unknowns from a given  $\mathbf{H}_{21}$ .

## 5 Retrieving Our Unknowns

Using images taken simultaneously by both cameras, we can compute a planar homography  $\mathbf{H}_{21}$  for each time instant using the image correspondences between both images. Furthermore, if the video streams of both cameras are not yet synchronized, we can exploit the knowledge that the planar events are dynamic in order to synchronize both streams. Only the correct applied time-shift between both video streams will lead to planar homographies between both cameras which have a large number of supporting correspondences. Using the computed homography  $\mathbf{H}_{21}$  and equations 2 and 4, we can derive the following:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \frac{1}{f_1} \end{bmatrix} \mathbf{H}_{norm} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & f_2 \end{bmatrix} = \lambda \mathbf{R}_1^T \mathbf{B} \mathbf{R}_2 \tag{5}$$

$$\text{with } \mathbf{H}_{norm} = \left( \mathbf{K}_1^{fixed} \right)^{-1} \mathbf{H}_{21} \mathbf{K}_2^{fixed} \tag{6}$$

To determine the unknowns we use of the special eigenvalue properties of all square matrices:  $\det(\mathbf{A}) = \prod \text{eigenvalues}$  and  $\text{trace}(\mathbf{A}) = \sum \text{eigenvalues}$ . However, before we can use these we need to put equation 5 in the correct (eigenvalue decomposition) form by right or left multiplying each side of the equation with its transpose. For the case of right multiplication, we end up with the following equation, where  $f'_1$  replaces  $\frac{1}{f_1}$  and  $f'_2$  replaces  $f_2$  for ease and symmetry of notation:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & f'_1 \end{bmatrix} \mathbf{H}_{norm} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & f'_2 \end{bmatrix} \mathbf{H}_{norm}^T \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & f'_1 \end{bmatrix} = \lambda^2 \mathbf{R}_1^T \mathbf{B} \mathbf{B}^T \mathbf{R}_1 \tag{7}$$

The three eigenvalues of the right-hand side are the eigenvalues of  $\mathbf{B} \mathbf{B}^T$  multiplied by  $\lambda^2$ . For the equality to hold, they should equal the eigenvalues of the left-hand side. The eigenvalues  $\mu$  of a square matrix  $\mathbf{A}$  are found by solving  $\det(\mathbf{A} - \mu \mathbf{I}) = 0$  in which  $\mathbf{I}$  is the identity matrix. Using this, it can be easily proven that the three eigenvalues  $\mu$  of  $\mathbf{B} \mathbf{B}^T$  are not distinct (multiple identical eigenvalues) only when the two camera centres lie on a line parallel to the plane normal. However, this specific case is of no interest to us since there would be an obvious ambiguity (a rotation of both cameras around the plane normal) in the retrieval of the camera rotations. When assuming the following representation for  $\mathbf{H}_{norm}$ :

$$\mathbf{H}_{norm} = \begin{bmatrix} m_0 & m_1 & m_2 \\ m_3 & m_4 & m_5 \\ m_6 & m_7 & m_8 \end{bmatrix} \tag{8}$$

the three eigenvalues  $\mu$  of  $\mathbf{B} \mathbf{B}^T$  must make the following determinant zero:

$$\begin{vmatrix} m_{00} + m_{11} + m_{22} f_2'^2 - \lambda^2 \mu & m_{03} + m_{14} + m_{25} f_2'^2 & f_1'(m_{06} + m_{17} + m_{28} f_2'^2) \\ m_{03} + m_{14} + m_{25} f_2'^2 & m_{33} + m_{44} + m_{55} f_2'^2 - \lambda^2 \mu & f_1'(m_{36} + m_{47} + m_{58} f_2'^2) \\ f_1'(m_{06} + m_{17} + m_{28} f_2'^2) & f_1'(m_{36} + m_{47} + m_{58} f_2'^2) & f_1'^2(m_{66} + m_{77} + m_{88} f_2'^2) - \lambda^2 \mu \end{vmatrix} \tag{9}$$

in which  $m_{ij}$  is shorthand for  $m_i m_j$ . By filling in the three known and distinct eigenvalues  $\mu_1, \mu_2$  and  $\mu_3$  of  $\mathbf{B}\mathbf{B}^T$  we arrive at three independent equations in the three unknowns  $f_1', f_2'$  and  $\lambda$ . These are polynomials of the sixth power in  $\lambda$  and can seem very difficult to solve at first. However, two independent equations can already be obtained through simpler means: the special eigenvalue properties  $\det(\mathbf{A}) = \prod \text{eigenvalues}$  and  $\text{trace}(\mathbf{A}) = \sum \text{eigenvalues}$ . Using equations 7 and 9, and replacing  $f_i'^2$  by  $f_i''$  and  $\lambda^2$  by  $\lambda''$  we arrive at:

$$\lambda''^3 (\mu_1 \mu_2 \mu_3) = f_1'' f_2'' \det(\mathbf{H}_{norm})^2 \tag{10}$$

$$\lambda'' \sum_{1,2,3} \mu_i = \sum_{0,1,3,4} m_{ii} + f_1'' (m_{66} + m_{77}) + f_2'' (m_{22} + m_{55}) + f_1'' f_2'' m_{88} \tag{11}$$

The required third equation is found by using the two equations above to remove the highest order terms in  $\lambda''$  from equation 9. As expected, no matter if we set  $\mu$  equal to  $\mu_1, \mu_2$  or  $\mu_3$  in equation 9, we always end up with the same equation:

$$\lambda''^2 (\mu_{12} + \mu_{23} + \mu_{13}) = c_8^2 + f_1'' (c_2^2 + c_5^2) + f_2'' (c_6^2 + c_7^2) + f_1'' f_2'' \sum_{0,1,3,4} c_i^2 \tag{12}$$

in which  $\mu_{ij}$  is shorthand for  $\mu_i \mu_j$ , and  $c_i$  is shorthand for the cofactor of element  $m_i$  in equation 8. Cofactor  $c_i$  equals the determinant of the matrix which remains after deleting the row and column which contain element  $m_i$  in equation 8.

Together with equations 10 and 11, equation 12 forms a system of three independent non-linear equations in three variables. This system is solved in the following manner, for which the notation becomes too cumbersome to report it completely in terms of the known values of  $m_i$  and  $\mu_i$ . First, equation 10 is used to replace the product term  $f_1'' f_2''$  in equations 11 and 12. Next, after organizing the latter two equations so that terms in  $f_1''$  and  $f_2''$  show up on the left-hand side and terms in  $\lambda''$  on the right-hand side, we obtain the following system:

$$f_1'' \text{coef}_1 + f_2'' \text{coef}_2 = \text{poly}_1 \tag{13}$$

$$f_1'' \text{coef}_3 + f_2'' \text{coef}_4 = \text{poly}_2 \tag{14}$$

in which  $\text{coef}_i$  are constants which can be computed from the known values of  $m_i$ . The term  $\text{poly}_i$  is a third order polynomial in  $\lambda''$  with coefficients which are constant and computable from the known values of  $m_i$  and  $\mu_i$ . When both left-hand sides are linearly dependent, we can make the left-hand sides disappear by subtracting them. In that case, the right-hand sides form a third order polynomial in  $\lambda''$  which can be solved. Subsequently, the values of  $f_1''$  and  $f_2''$  can be determined by substituting the solutions for  $\lambda''$  in equations 10 and 11. When the left-hand sides are not linearly dependent, we use linear algebra to solve the system:

$$f_1'' = \frac{\begin{vmatrix} \text{poly}_1 & \text{coef}_2 \\ \text{poly}_2 & \text{coef}_4 \end{vmatrix}}{\begin{vmatrix} \text{coef}_1 & \text{coef}_2 \\ \text{coef}_3 & \text{coef}_4 \end{vmatrix}} \text{ and } f_2'' = \frac{\begin{vmatrix} \text{coef}_1 & \text{poly}_1 \\ \text{coef}_3 & \text{poly}_2 \end{vmatrix}}{\begin{vmatrix} \text{coef}_1 & \text{coef}_2 \\ \text{coef}_3 & \text{coef}_4 \end{vmatrix}} \tag{15}$$



which are both third order polynomials in  $\lambda''$  with coefficients dependent on the known values of  $m_i$  and  $\mu_i$ . When we insert equation 15 into equation 10, we get a sixth order polynomial in  $\lambda''$ . For each solution to  $\lambda''$  we can retrieve the corresponding values of  $f_1''$  and  $f_2''$  by using equation 15. Remembering all the substitutions, we can finally determine the original values of  $f_1 = \frac{1}{\sqrt{f_1''}}$ , of  $f_2 = \sqrt{f_2''}$  and of  $\lambda = \pm\sqrt{\lambda''}$  which need to be used in equation 5 to compute:

$$\mathbf{H}_{new} = \frac{1}{\lambda} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \frac{1}{f_1} \end{bmatrix} \mathbf{H}_{norm} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & f_2 \end{bmatrix} = \mathbf{R}_1^T \mathbf{B} \mathbf{R}_2 \tag{16}$$

Since  $f_1$  and  $f_2$  represent physical zoom factors they necessarily should have a positive value. The sign of  $\lambda$  can be determined from equation 16, because the determinant of both sides of the equation should be the same. From equation 16 the rotation matrices  $\mathbf{R}_1$  and  $\mathbf{R}_2$  can now be determined using the singular value decomposition of  $\mathbf{H}_{new} = \mathbf{U}_1 \mathbf{S}_1 \mathbf{V}_1^T$  and  $\mathbf{B} = \mathbf{U}_2 \mathbf{S}_2 \mathbf{V}_2^T$ , where  $\mathbf{U}_i$  and  $\mathbf{V}_i$  are orthogonal  $3 \times 3$  matrices, and  $\mathbf{S}_1$  and  $\mathbf{S}_2$  are identical  $3 \times 3$  diagonal matrices. It is important to note that a singular value decomposition is not unique in the sense that:

$$\mathbf{U} \mathbf{S} \mathbf{V}^T = \mathbf{U} \mathbf{D} \mathbf{S} \mathbf{D}^T \mathbf{V}^T = \mathbf{U}' \mathbf{S}' \mathbf{V}'^T \quad \text{with } \mathbf{D} = \begin{bmatrix} \pm 1 & 0 & 0 \\ 0 & \pm 1 & 0 \\ 0 & 0 & \pm 1 \end{bmatrix} \tag{17}$$

From this it follows that:

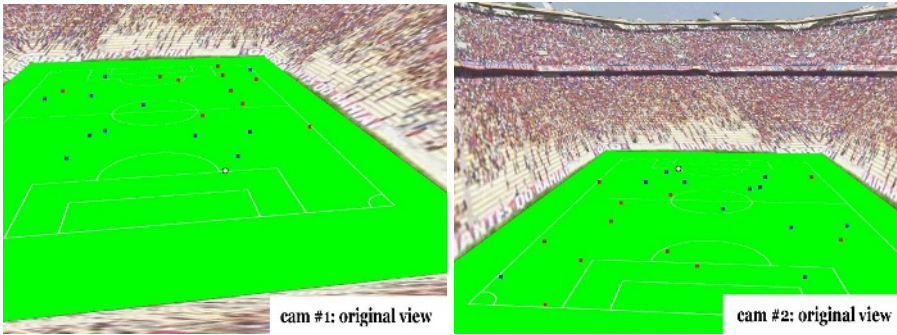
$$\mathbf{R}_1 = \mathbf{U}_2 \mathbf{D}_2 \mathbf{D}_1^T \mathbf{U}_1^T \quad \text{and} \quad \mathbf{R}_2 = \mathbf{V}_2 \mathbf{D}_2 \mathbf{D}_1^T \mathbf{V}_1^T \tag{18}$$

There are only four different combinations of  $\mathbf{D}_2 \mathbf{D}_1^T$  which ensure that the rotation matrices  $\mathbf{R}_1$  and  $\mathbf{R}_2$  are expressed in a right-handed world coordinate system, meaning that  $\det(\mathbf{R}_i)$  equals 1.

## 6 Solving Our Ambiguities

We need to solve a sixth order polynomial in  $\lambda''$  which can lead up to six different solutions for  $\lambda''$ . However, taking into account that the result should be a positive real value (since  $\lambda'' = \lambda^2$ ), it is already possible to eliminate some of the solutions. Furthermore, for the remaining set we can compute the corresponding values of  $f_1'' = \frac{1}{f_1^2}$  and  $f_2'' = f_2^2$  using equation 15. Here again, the values of  $f_1''$  and  $f_2''$  have to be positive and must be situated in a possible physical zoom range (which can be known beforehand). This allows us to remove even more implausible solutions.

As mentioned in the previous section, each possible solution for  $(\lambda, f_1, f_2)$  has associated with it four solutions for the rotation matrix  $\mathbf{R}_i$  of camera 1 and 2 respectively. The camera rotations should be such that both viewing frustums are directed towards the plane and overlap in the plane, since otherwise the planar homography  $\mathbf{H}_{21}$  could not have been determined in the first place. This



**Fig. 1.** View of camera 1 and camera 2 on the virtual soccer stadium. Players are rendered as points on the field. The ball is depicted by a white circle.

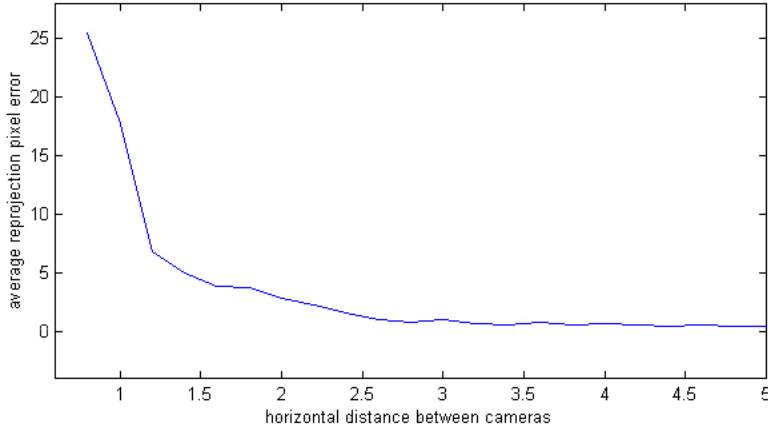
constraint is not necessarily enough to arrive at a single solution. However, in most practical scenarios, the dynamic events such as sports will be recorded with pan-tilt cameras. The word 'pan-tilt' implies that the camera cannot perform a roll motion around its optical axis. In mathematical terms this means that the X-axis of each camera (equal to the first column of  $\mathbf{R}_i$ ) will always be (close to) parallel with respect to the plane. An additional constraint is that, events are typically never recorded up-side down. Therefore, for those scenarios, we can easily single out the correct solution as the one which obeys the previously mentioned constraints the most.

## 7 Experiments

We verify our theory using an artificial soccer scenario. We constructed a virtual stadium in which two cameras are posted on either side of the soccer field at the same height with respect to the field, as shown in Figure 1. Since the global scale of the scene does not matter, the height of the camera positions may be assumed to equal 1. The resolution of the cameras was taken to be 640x480.

Both cameras rotate and zoom independently. At each time instant, the randomly moving players are rendered in both cameras as points. These points are detected in both images and used to estimate the planar homography at each time instant. Due to the discretization noise on the recovered image positions, the estimated homography does not perfectly match the ground truth homography. This allows us to investigate the effect of noise.

Using our method we extract from this homography the zoom and rotation of both cameras. Using the recovered parameters, the ground truth positions of the players are re-projected into the images and their projection is compared with the detected locations in the original images. The average pixel reprojection error which results is a good measure for the accuracy and suitability of the extracted zoom and rotation. What is very interesting to notice from Figure 2 is that the influence of the discretization noise increases when the horizontal

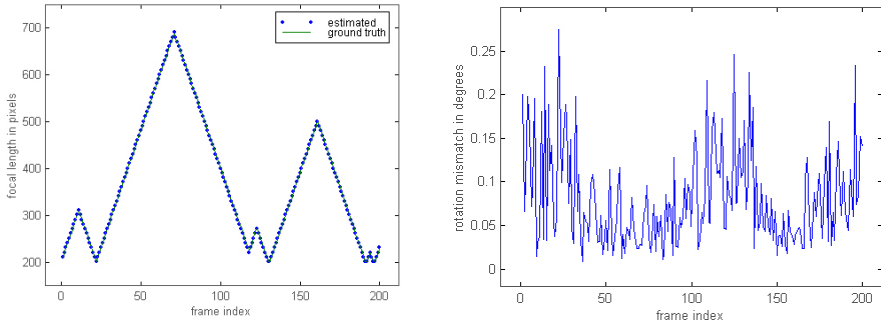


**Fig. 2.** The average reprojection pixel error in function of the horizontal distance between both cameras. The height of the camera above the plane is fixed and equals 1. For each distance, the average reprojection error was computed over 100 trial runs.

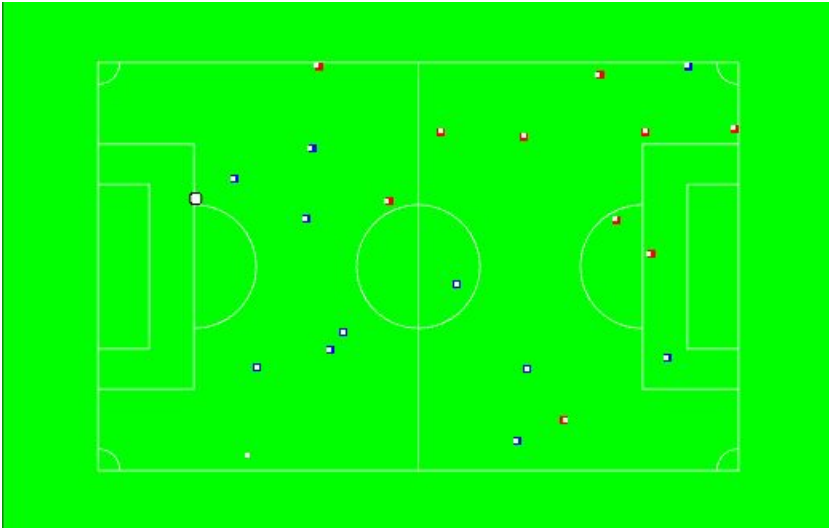
distance between both cameras decreases, while their vertical height stays fixed (equal to 1). Since the absolute scene scale is irrelevant, we can also deduce that the average reprojection error will increase when the horizontal distance remains fixed but the height of the cameras increases.

This behavior can be explained as follows. When matrix  $\mathbf{B}$  would be the identity matrix then the rotation matrices  $\mathbf{R}_1$  and  $\mathbf{R}_2$  merge into a single rotation matrix in equation 2. Obviously the closer the setup comes to this degenerate case the more results degrade. The cross-coupling between the rotations of both cameras diminishes the more matrix  $\mathbf{B}$  differs from the unity matrix. This can be achieved by moving the cameras further apart horizontally, bringing them closer to the plane or making the camera heights differ more, see equation 3. The idea is that the more the perspective effects of the scene differ between both images, the less coupling there is between the retrieved camera parameters. This result was anticipated in section 5, where we stated that the three eigenvalues  $\mu$  of  $\mathbf{B}\mathbf{B}^T$  would no longer be distinct in the case of no horizontal distance between both cameras. This degenerate case would render our three equations 10, 11 and 12 linearly dependent, meaning that no closed set of solutions can be found.

Another way of visualizing results is by comparing the ground truth focal lengths to the retrieved values in function of time, see Figure 3. The average relative error in focal length was found to be 0.3%. Also the mismatch angle (rotation angle needed to turn one coordinate system into another) between the local coordinate frame attached to the ground truth camera and the one determined by the retrieved rotation can be plotted in function of time. The average mismatch in rotation was found to be 0.07 degrees.



**Fig. 3.** Left: Comparison between the varying ground truth focal length (in pixels) and the recovered value for camera 1, in function of the frame index. Right: The rotational difference in degrees between the ground truth rotation and the recovered rotation of camera 1, in function of the frame index.



**Fig. 4.** A new virtual viewpoint of the reconstructed soccer game. The dark points represent the reconstructed player positions. The smaller white points contained within them, depict the ground truth positions of the players. The ball is depicted by a white circle.

The position of the players on the field can be determined from the detected image positions and the recovered camera rotation and zoom. Figure 4 demonstrates how this then enables us to view the action from a totally new virtual viewpoint, which would be the main application of our algorithm.

## 8 Conclusion and Future Work

In this paper, we put forth the mathematics able to extract the zoom and rotation of two cameras having fixed translation, from a single planar homography. It was shown how this particular case is very appealing since many scenarios such as sport events or surveillance lend themselves to this setting. The theory was tested on an artificial scenario, which demonstrated that the effect of noise on the final result depends on the setup of the cameras. We could deduce that the parameter coupling between both cameras becomes less the further the cameras are away from each other, and the closer they are to the plane. Intuitively, this corresponds to the reasoning that the more the perspective effects of the scene differ between both cameras, the better conditioned the homography decomposition is. This paper is very theoretic in nature and is mainly meant to introduce the mathematics for decomposing a planar homography based on a-priori knowledge on camera translations. The algorithm was tested on artificial data on which the effects of discretization noise could already be investigated. We will run additional experiments which will teach us about the influence of wrong a-priori information on the camera translations and matrices  $\mathbf{K}^{fixed}$ . We will continue our research by applying the new method to real-life situations, using real sport and surveillance scenarios.

## References

1. R. Haralick, C. Lee, K. Ottenberg, and M. Nölle. Review and analysis of solutions of the three point perspective pose estimation problem. *International Journal of Computer Vision*, 13(3):331–356, 1994.
2. E. Malis and R. Cipolla. Multi-view constraints between collineations: application to self-calibration from unknown planar structures. In *European Conference on Computer Vision*, 2000.
3. C. Rother, S. Carlsson, and D. Tell. Projective factorization of planes and cameras in multiple views. In *International Conference on Pattern Recognition*, 2002.
4. P. Sturm. Algorithms for plane-based pose estimation. In *Conference on Computer Vision and Pattern Recognition*, pages 706–711, 2000.
5. P. Sturm and S. Maybank. On plane-based camera calibration: A general algorithm, singularities, applications. In *Conference on Computer Vision and Pattern Recognition*, pages 432–437, 1999.
6. B. Triggs. Autocalibration from planar scenes. In *European Conference on Computer Vision*, pages 89–105, 1998.
7. Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.
8. Z. Zhang and A. Hanson. 3d reconstruction based on homography mapping. In *ARPA Image Understanding Workshop*, 1996.

# Dense Stereo by Triangular Meshing and Cross Validation

Peter Wey<sup>1</sup>, Bernd Fischer<sup>1</sup>, Herbert Bay<sup>2</sup>, and Joachim M. Buhmann<sup>1</sup>

<sup>1</sup> Institute of Computational Science

<sup>2</sup> Computer Vision Laboratory

ETH Zurich, Switzerland

<http://www.ml.inf.ethz.ch>, <http://www.vision.ee.ethz.ch>

**Abstract.** Dense depth maps can be estimated in a Bayesian sense from multiple calibrated still images of a rigid scene relative to a reference view [1]. This well-established probabilistic framework is extended by adaptively refining a triangular meshing procedure and by automatic cross-validation of model parameters. The adaptive refinement strategy locally adjusts the triangular meshing according to the measured image data. The new method substantially outperforms the competing techniques both in terms of robustness and accuracy.

## 1 Introduction

The demand for 3D models from 2D images has drastically increased during the last decade. Applications like web-publishing, surveillance and special effects for the movie industries rely on accurate representations of the imaged scene. Therefore, several approaches have been proposed to dense 3D reconstruction. Stereo matching has mainly been studied in the context of small-baseline stereo and for almost parallel planes [2].

The development of distinctive image features e.g. SIFT [3] that are able to find image correspondences between images taken under wide-baseline conditions, reduced the need for a large number of images to cover an important viewpoint change. High resolution images can be used that offer a high level of detail for the 3D modeling step. Strecha *et al.* [1] addressed the problem of dense stereo from such few high-resolution, wide-baseline images. The authors employ a probabilistic framework to model the image generation as a statistical process in order to gain the most probable scene represented by a depth map relative to a reference view. This estimate is achieved under the assumption that most surfaces obey the Lambertian model. Gargallo and Sturm [4] presented a similar approach using multiple depth maps. However, *dense* stereo still remains a hard problem and is viewed as the bottleneck of many applications of 3D reconstruction and view synthesis.

Our method extends the approach proposed by Strecha *et al.* by using a novel adaptive refinement technique based on a triangular mesh. The complexity of the mesh is automatically adjusted according to the measured image data. The

resulting representation then depends on the amount of texture in the scene and it does no longer use a predefined homogeneous pixel grid that finds no features for matching in homogeneous regions. The adaptive method clearly outperforms the competing methods like the approach of Strecha for scenes containing such locally homogeneous regions.

Various methods for dense 3D reconstruction depend on a variety of parameters to tune the prior knowledge, like the amount of smoothing in the model of Strecha or the number of triangles to represent the depth map in our adaptive refinement method. Therefore, we advocate a cross-validation method for model selection in 3D reconstruction. The cross-validation enables us to select the parameters (like smoothing priors) or model complexity in the triangular mesh. Furthermore, we adapted the mean squared error to compare the different 3D reconstruction methods.

In this paper we will give a brief description of Strecha’s probabilistic approach for 3D reconstruction (sec. 2). Our new adaptive refinement technique is described in section 3. In section 4 our new cross-validation technique is presented. The comparisons of the different methods and the results of the model selection are given in section 5.

## 2 Probabilistic Wide-Baseline Stereo

Strecha *et al.* [1] developed a performant approach for dense stereo reconstruction from a few wide-baseline images. Given a set of calibrated input images, the authors use a probabilistic framework for the estimation of the most probable depth map relative to a reference view. The procedure is based on a maximum a posteriori estimation by expectation maximization (EM). The cameras are calibrated beforehand using a robust interest point detection/description scheme like [3] and the calibration method described in [5] based on the estimation of the absolute quadric. An initial solution for EM is obtain from the detected interest points.

Given a set  $\mathcal{I} = \{I_1, \dots, I_N\}$  of  $N$  input images and their corresponding calibration matrices  $P_1, \dots, P_N$ , the aim is to estimate a depth map  $D_1$  for one of those images. Typically we have 3 to 10 images available. Without loss of generality, the first input image  $I_1$  can be chosen as the reference view for the estimation of  $D_1$ .

If the depth value of the pixel  $\mathbf{x}_1$  in image  $I_1$  is known, the pixel coordinates can be transformed into any other image  $I_i$  of the set by the mapping  $\mathbf{x}_i = l_i(\mathbf{x}_1; D_1, P_1, P_i)$ . This mapping is abbreviated as  $\mathbf{x}_i = l_i(\mathbf{x}_1)$ . The input images  $I_i$  are modeled as noisy measurements of the true image irradiance  $I_1^*$  having a normally distributed noise  $\epsilon$  with zero mean.

$$I_i(l_i(\mathbf{x}_1)) = I_1^*(\mathbf{x}_1) + \epsilon \quad \text{with} \quad \epsilon \sim \mathcal{N}(0, \Sigma), \quad (1)$$

where  $\Sigma$  is the covariance matrix. The parameters  $D_1$ ,  $I_1^*$ , and  $\Sigma$  are the parameters to be estimated.

The method is based on images that were taken under wide-baseline conditions and faces therefore the problem of occlusions and self-occlusions. In wide-baseline conditions, we cannot assume that the whole scene is visible in every image. Thus, only mutually visible information should be used for the computation of image correspondences. This problem is addressed by introducing a set of visibility maps  $V_1, \dots, V_N \in \{0, 1\}$ , where  $V_i(\mathbf{x}_i) = 1$  if the transformed pixel  $l_i(\mathbf{x}_1)$  is visible in the image  $I_i$  and  $V_i(\mathbf{x}_i) = 0$  otherwise. The posterior distribution of the depth map can be written as

$$p(D_1, I_1^*, \Sigma | \mathcal{I}) \sim \int p(\mathcal{I} | D_1, I_1^*, \Sigma, V) p(D_1, I_1^*, \Sigma | V) p(V) dV \quad (2)$$

$$p(\mathcal{I} | D_1, I_1^*, \Sigma, V) = \prod_{i=1}^N \prod_{\mathbf{x}_i} \mathcal{N}\{I_1^*(\mathbf{x}_i), \Sigma\}, \quad (3)$$

where  $p(\mathcal{I} | D_1, I_1^*, \Sigma, V)$  is the data likelihood. We choose a flat prior for  $I_1^*$  and  $\Sigma$ , which incorporates our expectations on the depth map  $p(D_1, I_1^*, \Sigma | V) = \exp\left(\frac{-\mathcal{R}(I_1^*, D_1)}{\lambda}\right)$ , where  $\lambda$  controls the amount of smoothing.  $\mathcal{R}(I_1^*, D_1)$  is a data-driven ‘regularizer’ of the depth map. As strong discontinuities should be allowed depending on the relation of the gradient of the estimated true image  $I_1^*$  and the changes in the depth map, this regularization term is defined as  $\mathcal{R}(I_1^*, D_1) = \nabla D_1^\top T(\nabla I_1^*) \nabla D_1$ .  $T(\nabla I_1^*)$  is the diffusion tensor defined as  $T(\nabla I_1^*) = \frac{1}{|\nabla I_1^*|^2 + 2\nu^2} (\nabla I_1^{*\perp} \nabla I_1^{*\perp\top} + \nu^2 \mathbf{1})$ , where  $\mathbf{1}$  is the identity matrix,  $\nu$  a parameter controlling the degree of anisotropy and  $\nabla I_1^{*\perp}$  is the vector perpendicular to  $\nabla I_1^*$ . The diffusion tensors controls the amount of punishment of depth discontinuities ( $\nabla D_1$ ). A high value of  $\nu$  sets  $T(\nabla I_1^*)$  towards the identity matrix and therefore all depth discontinuities will be treated the same way. Smaller values of  $\nu$  will lead to smaller values of  $\mathcal{R}$  if the image gradient is perpendicular to the current depth map discontinuity. An EM-algorithm that iteratively maximizes the posterior over all feasible depth maps and estimates the expectation of the visibility maps is used [1].

### 3 Adaptive Depth Map

The parameters for smoothness,  $\lambda$ , and anisotropy,  $\nu$  are globally chosen for the whole scene. However, the vast majority of scenes are combinations of rough and smooth parts. But, global values for  $\lambda$  and  $\nu$  can only be adjusted to one kind of scene. Locally chosen parameters lead to an extremely large number of parameters that have to be tuned (see section 4). Here, we present a novel approach to overcome these problems for scenes with variable smoothness. Our new approach locally determines the necessary amount of representation (the degrees of freedom) inside the depth map. This is done by adaptively refining a triangular representation of the depth map in those regions that need to capture fine details without over representing homogeneous regions. That means, small triangles for rough regions, but large and wide-spanning triangles for smooth regions.



At the end, the depth map consists of a triangular mesh with a small set of vertices. The vertices are connected using Delaunay triangulation [6]. The depth map is represented as a piecewise linear surface. Such a representation is motivated by the implicit prior knowledge that the scene contains locally planar patches. Moreover, it has the advantage to reduce the amount of memory space by an order of magnitude. Therefore the computation time is reduced also. The mesh refinement method is related to adaptive image polygonalization [7] for remote sensing.

For the estimation of the visibility maps, the true image irradiance  $I_1^*$ , and the covariance matrix  $\Sigma$ , we used the same method as in 2. The individual depth values of the pixels are sampled from the linearly interpolated depth map.

We have chosen a greedy approach to estimate the depth map. The optimization starts at a high scale resulting in a coarse approximation of the depth map. The depth map is now iteratively refined as follows. At each iteration, the resolution is locally increased only for the triangles for which depth discontinuities have not been captured at the current scale (in those regions that really need to be refined). New vertices have to be introduced on those triangles whose data likelihood could be substantially improved. From a computational point of view, this problem could be solved by testing every discrete position of the new vertex, optimizing its depth and remembering the best position. However, considering the resulting amount of computational work, this is not feasible.

The data log-likelihood of a triangular image patch  $P$  is

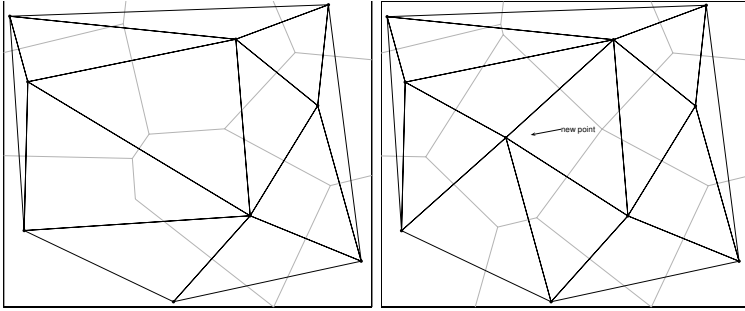
$$L_P(\mathcal{I}) = \frac{\sum_{\mathbf{x}_1 \in P} V_i(\mathbf{x}_1) (I_i(l_i(\mathbf{x}_1)) - I_1^*(\mathbf{x}_1))^t \Sigma^{-1} (I_i(l_i(\mathbf{x}_1)) - I_1^*(\mathbf{x}_1))}{\sum_{\mathbf{x}_1 \in P} V_i(\mathbf{x}_1)}. \quad (4)$$

In order to speed up the computation of the data likelihood for large triangles, only a subset of the pixels contained in the triangle are randomly selected and the data likelihood is estimated on this subset only. This sampling technique corresponds to the Nyström method used in numerical analysis. The greedy algorithm works as follows.

1. Determine the triangle with the smallest data likelihood.
2. Search for the position of the largest gradient magnitude inside this triangle.
3. Insert a new vertex at this position, adjust the triangulation and optimize the depth of this vertex according to the new triangulation.

If the triangle with the lowest likelihood is sub-divided, it is most probable that the resulting refinement will lead to a larger likelihood and therefore to an improved reconstruction. Accordingly, triangles that already represent the scene adequately, typically have high data likelihood values and will be left unchanged. Figure 1 shows the entity of the optimization problem. The Delaunay triangulation has to be adjusted locally. The depth value changes only for the newly inserted point.

Figure 2 shows the gradual refinement of the depth map for the city hall scene in Leuven. The upper left image shows the coarsest approximation on the depth map and the lower left image is the finest approximation.



**Fig. 1.** Inserting a new point in a given Delaunay triangulation. Left: Before insertion. Right: After insertion.

## 4 Model Selection

The adaptive mesh refinement method has only one free parameter: the number of vertices in the triangular mesh. This simplicity is in contrast to the approach followed by Strecha *et al.*, who have to adjust the amount of smoothing  $\lambda$  and the degree of local anisotropy  $\nu$ .

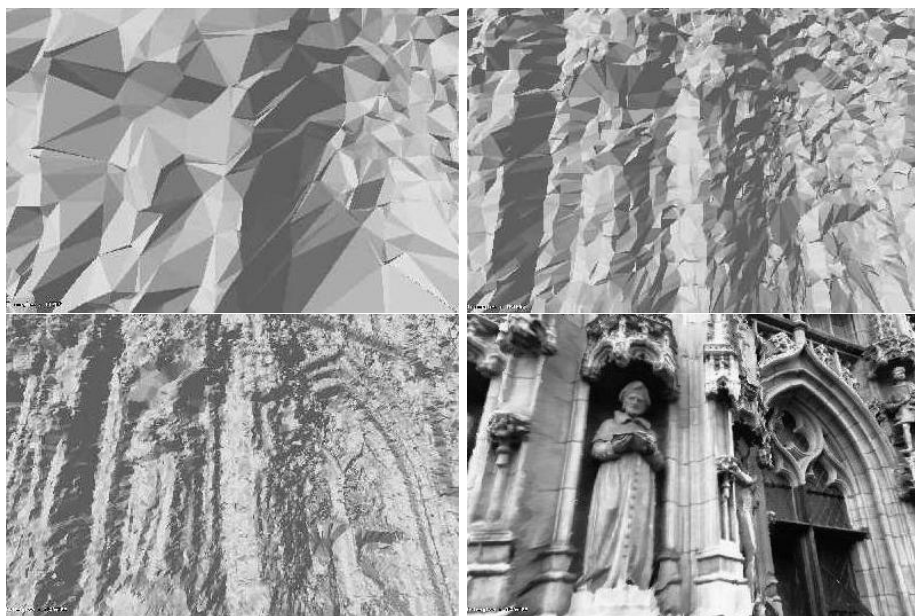
The goal is to automatically choose the parameters that maximize the quality of the depth map. Therefore, we need an appropriate loss function that estimates this quality for a specific parameter setting. A simple measure for the deviation of the two images, the original and the synthesized, is the mean squared error or in our case the mean squared color differences. Therefore, the error measure between an input image  $I_i$  and an virtual image  $\tilde{I}_i$  is the data log-likelihood

$$E(I_i, \tilde{I}_i) = \frac{\sum_{\mathbf{x}_1} V_i(\mathbf{x}_1) (\mathbf{y}_i - \tilde{\mathbf{y}}_i)^t \Sigma^{-1} (\mathbf{y}_i - \tilde{\mathbf{y}}_i)}{\sum_{\mathbf{x}_1} V_i(\mathbf{x}_1)}, \quad (5)$$

$$\text{where } \mathbf{y}_i = I_i(l_i(\mathbf{x}_1)) \text{ and } \tilde{\mathbf{y}}_i = \tilde{I}_i(l_i(\mathbf{x}_1)). \quad (6)$$

Here,  $\mathbf{y}_i = I_i(l_i(\mathbf{x}_1))$  is the vector of gray values for the different color channels. For each color channel the gray value is between 0 and 1. For RGB images, the range of this error measure lies between 0 and 3. The projected mesh does not cover every pixel in the new image. This fact is expressed by  $V_i(l_i(\mathbf{x}_1)) = 1$  (if visible) or  $V_i(l_i(\mathbf{x}_1)) = 0$  (not visible) for every pixel  $\mathbf{x}_1$  in the reference image. We are comparing only the visible image regions. This error measure is then validated by cross validation. The test error is the mean squared error on the test images. In the experiments presented we used the squared distance instead of the Mahalanobis distance.

To perform cross validation for a specific parameter setting, the input images are split up into two independent sets, a training set and a test set. Therefore, only a subset of the input images is used for the reconstruction algorithm. Then, using the obtained reconstruction, new virtual images seen from the cameras

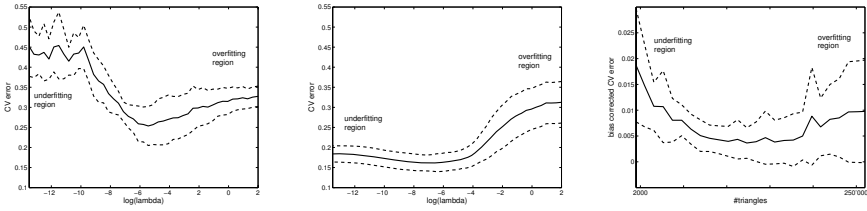


**Fig. 2.** Gradual refinement of the depth map for decreasing scales (top left to bottom left). Bottom right: Textured version of its neighbor on the left.

corresponding to the test images are created. These virtual images are compared with the original test images by the mean squared error.

To determine the optimal parameter setting for a specific algorithm, the whole parameter space could be sampled, keeping the best performing configuration. Such a procedure is redundant and time consuming to select the optimal parameters. In our greedy approach to increase the model complexity we do not have to recompute the whole depth map. Instead, we just interrupt the greedy method at a certain model complexity and compute the test error. This leads to a decrease of runtime compared to the anisotropy prior methods.

For the Leuven cityhall scene, we show the cross-validation error curves in figure 3 for the method proposed by Strecha *et al.* [1]. The setting here consists of seven images. These are split into a training set of four images and a test set of three images. The  $x$ -axis represents the parameter  $-\log \lambda$  and the  $y$ -axis the cross validation error. In both plots, one can nicely see the transition between the model over-fitting on the left hand side and under-fitting on the right hand side. In the over-fitting region not enough smoothing was performed. In the under-fitting area, the smoothness prior of the depth map dominates the information contained in the data likelihood. Low anisotropy leads to bad results due to the sensitivity of the smoothness penalty function towards image gradients (every image gradient permits the existence of depth discontinuities which lead to under-smoothing in flat, but highly textured areas of the scene). High values on the other hand ignore the information contained in the image



**Fig. 3.** Cross validation errors obtained with the Strecha’s method for two different choices of the parameter  $\nu$ . Left:  $\nu = 0.01$ , Middle:  $\nu = 100$ . Solid line: mean, dotted line: standard deviation. Right: bias corrected CV error for adaptive refinement.

gradients. The best results were achieved by setting  $\nu = 100$  and  $\lambda$  in the range of  $[250 \cdot 10^{-6} - 2000 \cdot 10^{-6}]$ .

Our method has a much lower tendency to produce an over- or under-fitting. Furthermore the standard deviation of our method is much smaller. To show that the adaptive refinement also generates an under- and over-fitting for small and large number of vertices, respectively, we subtracted an image bias from the error. The error is dependent on the base-line. The wider the baseline between training and test images, the larger the error. Shifting the error curve for each image down by the smallest value corrects this image dependent bias. Figure 3 (right) depicts the bias corrected cross-validation error. Here again one can clearly see the under- and also an over-fitting area.

## 5 Results and Comparisons

In this section, we show results of the adaptive refinement method compared to the method of Strecha *et al.*. Figure 4 shows the result on the Leuven cityhall scene. The mean cross validation error achieved is  $175.6 \cdot 10^{-6}$  and outperforms all the different variations of the method presented before. The best configuration with Strecha’s method achieved so far had the cross validation error  $181.3 \cdot 10^{-6}$ . Figure 4 shows both the untextured and textured views of the estimated dense reconstruction using our new adaptive refinement method.

Since the Leuven cityhall is a highly textured scene, figure 5 shows an example of a less-textured scene. It demonstrates that the result obtained with our method still provides a sufficient level of detail while making more sense for less-textured image regions. The first three columns of table 1 compare the cross-validation errors of the two methods for different resolutions. The first column represents the number of triangles of the adaptive depth map. The adaptive refinement method performs clearly better than the method of Strecha *et al.*

Figures 6 and 7 are other examples for our adaptive refinement method. This scene consist of 17 images of a metal lion. Half of the images have been used as training images and the other half as test images. Therefore, compared to the office scene before, much more images are used to reconstruct this scene. The cross validation errors of the two methods at different scale are presented in the last two columns of table 1. Here again, the adaptive refinement method clearly



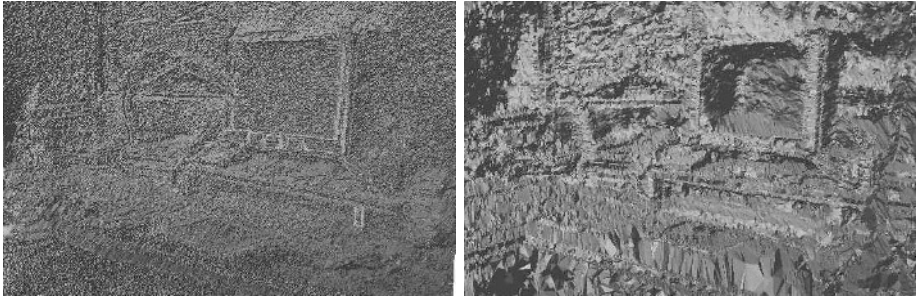
**Fig. 4.** Untextured and textured views of the reconstructed city hall using the adaptive triangulation method

outperforms its contender for all resolution levels. The large close-ups in figure 7 clearly show the difference between the depth map representation of the two methods. While the first method represents the scene in a uniformly sampled grid, the adaptive refinement method is able to capture the depth features with detail dependent resolution.

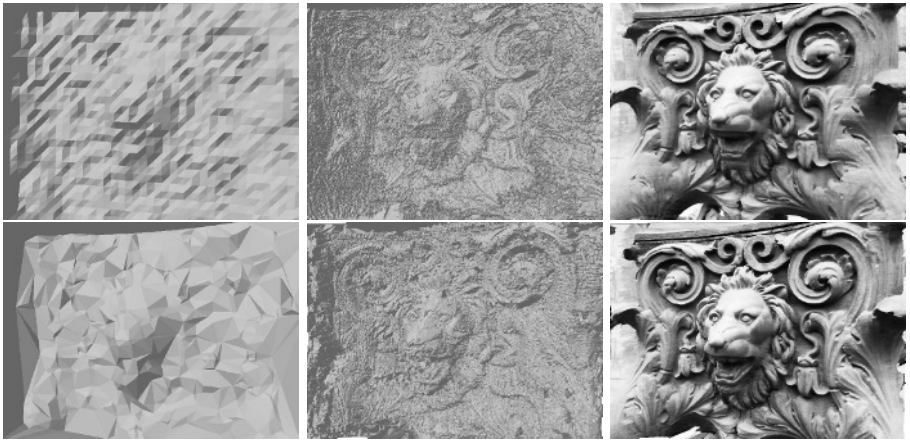
The reconstructions achieved with the method of Strecha *et al.* contain a high amount of noise (peaks). A higher amount of smoothing would have lead to worse results because important depth features would have been wipped out during the smoothing. The adaptive triangulation method is able to overcome this problem and it generates better results and therefore it outperforms the competing methods for all test cases; the reconstructed models achieve a higher quality using less triangles than alternative approaches.

**Table 1.** Cross validation error for different scales and resolutions. The ‘-’ denotes, that the reconstruction has not been performed for the corresponding resolution due to time limits.

resolution	Office Scene			Lion Scene		
	# triangles	Strecha	adapt. ref.	# triangles	Strecha	adapt. ref.
167x126	3'257	0.075	0.065	2'864	0.128	0.102
335x252	13'738	0.076	0.063	18'663	0.167	0.118
670x504	21'147	0.079	0.063	26'876	0.124	0.119
1340x1009	118'369	-	0.064	115'738	-	0.126



**Fig. 5.** Results of the method of Strecha (left) and with adaptive refinement (right)

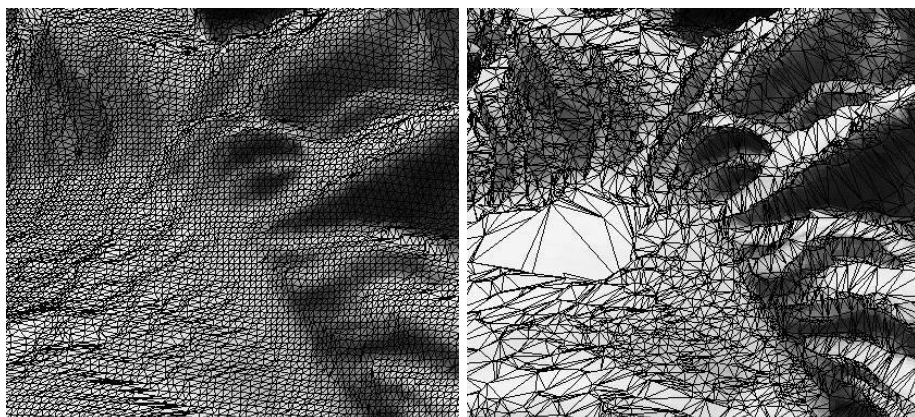


**Fig. 6.** Results of the method of Strecha *et al.* in the upper row and of the adaptive refinement method in the lower row. The images on the left denote the initial solution. The images on the right show the synthesized images.

## 6 Conclusion

This paper presented a novel approach for the estimation of dense depth maps, given a calibrated set of input images. A technique has been introduced to rate different parameter settings and to compare reconstructions using an objective measure based on cross validation.

Furthermore, we presented a new way of successively refining the depth map estimation. The adaptive refinement method based on triangular meshes was shown to outperform an established method for dense depth map estimation. The improved quality of the reconstructions is shown by a novel cross-validation method.



**Fig. 7.** Direct comparison of the two methods. The left image shows the result using the method of Strecha *et al.* The plot on the right hand side shows the same scene reconstructed with the adaptive refinement strategy.

## References

1. Strecha, C., Fransens, R., van Gool, L.: Wide-baseline stereo from multiple views: a probabilistic account. In: Conference on Computer Vision and Pattern Recognition (CVPR'04). Volume 1., IEEE Computer Society (2004) 552–559
2. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision* **47**(1) (2002) 7–42
3. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* **60**(2) (2004) 91–110
4. Gargallo, P., Sturm, P.: Bayesian 3d modeling from images using multiple depth maps. In: Conference on Computer Vision and Pattern Recognition. Volume 2., IEEE (2005) 885–891
5. Hartley, R.I., Zisserman, A.: *Multiple View Geometry in Computer Vision*. Second edn. Cambridge University Press, ISBN: 0521540518 (2004)
6. Faugeras, O.: *Three-Dimensional Computer Vision*. MIT Press (1993)
7. Hermes, L., Buhmann, J.M.: A minimum entropy approach to adaptive image polygonization. *IEEE Transactions on Image Processing* **12**(10) (2003) 1243 – 1258

# Low-Cost Laser Range Scanner and Fast Surface Registration Approach

Simon Winkelbach, Sven Molkenstruck, and Friedrich M. Wahl

Institute for Robotics and Process Control, Technical University of Braunschweig,  
Mühlenpfordtstr. 23, D-38106 Braunschweig, Germany  
{S.Winkelbach, S.Molkenstruck, F.Wahl}@tu-bs.de

**Abstract.** In the last twenty years many approaches for contact-free measurement techniques for object surfaces and approaches for 3d object reconstruction have been proposed; but often they still require complex and expensive equipment. Not least due to the rapidly increasing number of efficient 3d hard- and software system components, alternative low-cost solutions are in great demand. We propose such a low-cost system for 3d data acquisition and fast pairwise surface registration. The only hardware requirements are a simple commercial hand-held laser and a standard grayscale camera.

## 1 Introduction

Triangulation-based laser range finders and light-striping techniques are well-known since more than twenty years (e.g. [1], [2]). Beside other active techniques – like structured light, coded light, time of flight, Moiré interferometry, etc. (see e.g. [3] for an overview) – laser range scanners are commonly used for contactless measuring of surfaces and 3d scenes in a wide range of applications. The field of application comprises computer graphics, robotics, industrial design, medical diagnosis, archaeology, multimedia and web design, as well as rapid prototyping and computer-aided quality control. Most commercial laser scan systems use a camera and a laser beam or laser plane. The surface recovery is based on triangulation, i.e. the intersection of the illuminating laser beam and the rays projected back to the camera. Expensive high-precision actuators are often used for rotating/translating the laser plane or for rotating/translating the object.

Some alternative hand-held devices avoid expensive actuators and furthermore improve the flexibility of the scanning process. These approaches have to determine the position and orientation of the laser device on-line. Such an on-line tracking is done by various mechanisms like optical LED tracking, electromagnetic sensors or mechanical positioning arms (see e.g. [3], [4]).

Instead of an external tracking system, we propose a real-time self-calibration of a hand-held laser plane, which is based on a simple analysis of the laser stripes in the camera images. Thus, the laser line can be swept manually over the object during the scan, which has several advantages: (i) Only the lightweight laser has to be held, which allows a convenient scanning process. (ii) The low-cost



hardware requirements are even affordable for students and novice developers. (iii) The illumination direction is flexible and allows an interactive avoidance of laser shadow problems and outliers. The only precondition is a known background geometry, which serves as laser calibration target.

Thus, our approach can be regarded as a generalization of Zagorchev and Goshtasby [4]. They use a reference double-frame, which is placed around the object and acts as calibration target. In their approach, the laser is calibrated using the four visual intersection points of the laser and the double-frame. Detection of the red laser lines relies on an appropriate analysis of the red components in color images. However, a standard color camera may impair the detection accuracy, since only every fourth pixel can capture red light, the remaining pixels are interpolated. Our approach exhibits several advantages over [4]: A precise reference double-frame, which has to be adapted to the object size, is not needed. The calibration target can be almost arbitrarily shaped (e.g. an arbitrary background). Moreover, the laser calibration is more robust and accurate, since we use much more than four points to calibrate the laser, and we rely on subpixel analysis of grayscale difference images. Due to our subsequent fast surface registration, the object can be moved freely between different scans. Thus, a repositioning and recalibration of the camera to get different viewing directions is not necessary; it is easily possible to scan an object from *all* sides, even from the bottom.

An outline of the numerous publications dealing with registration techniques would go beyond the scope of this paper. Therefore, we only give a short overview of the most related work: A very popular surface registration approach is the *iterative closest point* (ICP) algorithm from Besl and McKay [5]. The algorithm iteratively improves an initial solution according to some fitness criterion. Although many enhancements to the original method have been suggested (e.g. [6], [7]), it still requires a good initial guess to find the global optimum. Most approaches are using surface features to find corresponding point pairs. Features vary from simple properties like *curvatures*, to complex vectors like *point signatures* [8], *surface curves* e.g.[9], [10], *spin-images* [11] or *salient points* [12]. However, their usage cannot guarantee unique point correspondences; nevertheless, it can highly constrain the search space. A well-known category dealing with object recognition and localization are the *pose clustering* approaches (also known as *hypothesis accumulation* or *generalized Hough transform* e.g. [13]). The drawback of voting tables is their high time and space complexity, particularly in case of large data sets and high-dimensional search spaces.

The authors of [14] give an thorough overview of current registration techniques and propose a new approach, which is based on a genetic algorithm to find pose hypotheses and a novel *surface interpenetration measure* as quality criterion. Unfortunately, this complex approach needs triangle meshes and a substantial run-time (5 minutes for 10000 points on a 1.7 GHz PC for a pairwise match). We will show that our approach is able to achieve an adequate match of bigger data sets in less than a second.

The surface registration method is a significant improvement of the *random sample matching* [15], which is an efficient and robust approach for matching

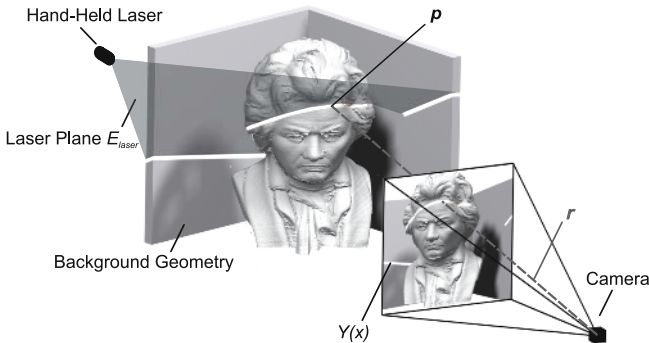
fragments of broken objects without knowing an initial solution. This method is based on the *RANSAC* algorithm introduced in [16]. The repeated procedure is simple but powerful: First, a likely hypothesis is generated randomly from the input data set. Subsequently, the quality of the hypothesis (number of contact points) is evaluated.

## 2 Hand-Held Laser Scanner

The basic idea of our self-calibrating laser scanner is quite simple. The laser ray, expanded to a plane by a cylindrical lens, has to intersect two things at the same time: the (unknown) surface, and the a priori known reference geometry (usually the background). The visible intersection with the background is used to calibrate the laser, i.e. to calculate the exact 3d pose of the laser plane  $E_{Laser}$ . With this knowledge we can triangulate new 3d point coordinates of the object's surface by intersecting the laser plane with the projecting rays. Certainly, the camera must have been calibrated so that its external and internal parameters are exactly known. In our setup we use markers on the background and Tsai's camera calibration method [17]. Thus, the exact coordinates of the background structure with respect to the camera coordinate system are implicitly known.

### 2.1 On-Line Laser Calibration

In many cases, a simple background (e.g. the natural environment) can be used for laser calibration. Under the assumption that the background geometry is known, we can obtain some point coordinates of the visible laser line by intersecting background and camera projection rays. Provided that these 3d intersection points are linearly independent, they constrain all degrees of the laser plane's pose. Although many background shapes are imaginable, the probably most applicable, available, and easy-to-use background will be the corner of a room, or two solid boards standing together in an exactly known angle.



**Fig. 1.** Laser triangulation: 3d scene and 2d camera image. The intersection of a projection ray  $r$  with the laser plane  $E_{Laser}$  results in new 3d point  $p$ .

It is important to find the coordinates of the laser line in the camera images  $I$  as precisely as possible. Thus, it is useful to take a reference image  $I_R$  without laser light and to use difference images  $I_d = I - I_R$  in the following. As the laser line will be rather horizontal or vertical, we can reduce the problem to a 1d detection of the laser line in each single column or row of the image, respectively. Without loss of generality, we assume in the following that the laser line is rather horizontal. We can find the line with subpixel accuracy by calculating the (weighted) average  $Y(x)$  of the “bright” pixel coordinates in each column  $x$ .

After obtaining the function  $Y(x)$  of the laser line, our next task is to calculate the 3d pose of the laser plane. We use the RANSAC method [16] to repeatedly select three random pixels  $Y(x_1), Y(x_2), Y(x_3)$ , and assume that they belong to the background. Since the camera’s internal and external parameters have been calibrated, we can obtain the equation of three “light rays”  $r_i$  for each of these pixels and intersect them with the known background geometry, resulting in three surface points  $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$ . Unless they are linearly dependent, they define a possible laser plane pose. These hypotheses can be quickly computed and evaluated using the number of inliers of  $Y(x)$  as a quality criterion.

### 2.2 Triangulation of 3D Points

From the previous step, we know the equation of the laser plane  $E_{Laser}$  and a number of image pixels from  $Y(x)$  that are both in that plane and on the object we are scanning (see Fig. 1). Again we can obtain the equation of a “light ray”  $r$  for each of those pixels. A new surface point of our object can be easily computed by the intersection  $\mathbf{p} = r \cap E_{Laser}$ .

In the process of scanning, the user generally sweeps the laser plane over the object multiple times. In this way, he can “brush over” outlying values and increase the precision where necessary. Thus, it often happens that the algorithm obtains several surface points for the same image pixel. These should be merged using averaging (fast and easy) or median filtering (memory consuming and slower, but very useful w.r.t. possible outliers).

All 3d points collected with this procedure form the visible surface of the object from one viewing direction. To obtain a full 360° model, scans from different directions have to be registered.

## 3 Fast Surface Registration

Given a set  $\mathcal{P}_A$  of 3d point coordinates  $\mathbf{p}_1, \dots, \mathbf{p}_k$  of the surface A and a set  $\mathcal{N}_A$  of corresponding 3d surface normals  $\mathbf{n}_1, \dots, \mathbf{n}_k$  (outward-pointing unit vectors) at these points. Referring to [11], we call the combination of a point with its normal an *oriented point*. This gives us the set of oriented points  $\mathcal{A}$  of surface A and the set of oriented points  $\mathcal{B}$  of the counter surface B

$$\mathcal{A} := \{ \mathbf{u} = [\mathbf{p}_u, \mathbf{n}_u] \mid \mathbf{p}_u \in \mathcal{P}_A \text{ and } \mathbf{n}_u \in \mathcal{N}_A \}, \tag{1}$$

$$\mathcal{B} := \{ \mathbf{v} = [\mathbf{p}_v, \mathbf{n}_v] \mid \mathbf{p}_v \in \mathcal{P}_B \text{ and } \mathbf{n}_v \in \mathcal{N}_B \}. \tag{2}$$

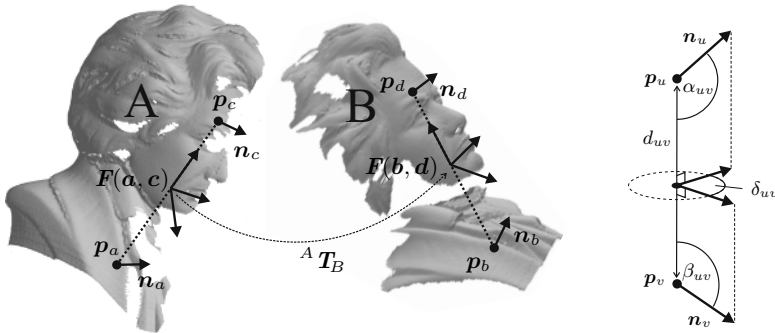
A *tangential contact* between two oriented points  $\mathbf{a} \in \mathcal{A}$  and  $\mathbf{b} \in \mathcal{B}$  means that the point coordinates and the respective surface normals coincide. We say  $\mathbf{a}$  is in tangential contact with  $\mathbf{b}$  if  $\mathbf{p}_a = {}^A T_B \cdot \mathbf{p}_b$  and  $\mathbf{n}_a = {}^A T_B \cdot \mathbf{n}_b$ , where  ${}^A T_B$  is the relative transformation in homogeneous coordinate notation. We can construct a pose hypothesis by assuming a contact between some points on each surface. More precisely, four given oriented surface points  $\mathbf{a}, \mathbf{c} \in \mathcal{A}$  and  $\mathbf{b}, \mathbf{d} \in \mathcal{B}$  are sufficient, if we assume a tangential contact between  $\mathbf{a}$  and  $\mathbf{b}$  as well as between  $\mathbf{c}$  and  $\mathbf{d}$ . This assumption constrains all degrees of freedom of the relative transformation. As illustrated in Fig. 2 (left), we can determine the homogeneous  $4 \times 4$  transformation matrix by multiplying two frames  ${}^A T_B = \mathbf{F}(\mathbf{a}, \mathbf{c})^{-1} \cdot \mathbf{F}(\mathbf{b}, \mathbf{d})$ , where the function  $\mathbf{F}(\mathbf{u}, \mathbf{v})$  represents a coordinate system lying between the oriented points  $\mathbf{u}$  and  $\mathbf{v}$

$$\mathbf{F}(\mathbf{u}, \mathbf{v}) := \begin{bmatrix} \frac{\mathbf{p}_{uv} \times \mathbf{n}_{uv}}{\|\mathbf{p}_{uv} \times \mathbf{n}_{uv}\|} \mathbf{p}_{uv} & \frac{\mathbf{p}_{uv} \times \mathbf{n}_{uv} \times \mathbf{p}_{uv}}{\|\mathbf{p}_{uv} \times \mathbf{n}_{uv} \times \mathbf{p}_{uv}\|} \frac{\mathbf{p}_u + \mathbf{p}_v}{2} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

with the difference vector  $\mathbf{p}_{uv} := (\mathbf{p}_v - \mathbf{p}_u) / \|\mathbf{p}_v - \mathbf{p}_u\|$  and the combined normal vector  $\mathbf{n}_{uv} := \mathbf{n}_u + \mathbf{n}_v$ . To avoid singular frames, we must ensure that the length of  $\mathbf{p}_{uv}$  and  $\mathbf{n}_{uv}$  is not zero. An exact coverage of both point pairs with opposed normals is only possible if their relative distances and angles are identical. To verify this constraint, we define a 4d relation vector of an oriented point pair

$$\text{rel}(\mathbf{u}, \mathbf{v}) := \begin{bmatrix} d_{uv} \\ \cos \alpha_{uv} \\ \cos \beta_{uv} \\ \delta_{uv} \end{bmatrix} := \begin{bmatrix} \|\mathbf{p}_v - \mathbf{p}_u\| \\ \mathbf{n}_u \cdot \mathbf{p}_{uv} \\ \mathbf{n}_v \cdot \mathbf{p}_{uv} \\ \text{atan2}(\mathbf{n}_u \cdot (\mathbf{p}_{uv} \times \mathbf{n}_v), (\mathbf{n}_u \times \mathbf{p}_{uv}) \cdot (\mathbf{p}_{uv} \times \mathbf{n}_v)) \end{bmatrix}, \quad (4)$$

consisting of the Euclidean point distance  $d_{uv}$ , the angles of inclination  $\alpha_{uv}$  and  $\beta_{uv}$  between the normals  $\mathbf{n}_u$  and  $\mathbf{n}_v$ , the line connecting  $\mathbf{p}_u$  and  $\mathbf{p}_v$ , and finally the rotation angle  $\delta_{uv}$  between the normals around the connection line. The four relations are also illustrated in Fig. 2 (right). Note that the relation vector



**Fig. 2.** (Left) relative transformation  ${}^A T_B$  between supposed contact points; (right) relations between the oriented points  $\mathbf{u}$  and  $\mathbf{v}$

is invariant under rotation and translation. Incidentally, Wahl, Hillenbrand and Hirzinger [18] showed that similar relation vectors can be accumulated in feature histograms for rapid 3d-shape classification. Using these relation vectors, the set of valid pose hypotheses  $\mathcal{H}$  can be specified by

$$\mathcal{H} := \{(\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}) \mid \text{rel}(\mathbf{a}, \mathbf{c}) = \text{rel}(\mathbf{b}, \mathbf{d}); \mathbf{a}, \mathbf{c} \in \mathcal{A}; \mathbf{b}, \mathbf{d} \in \mathcal{B}\} . \tag{5}$$

### 3.1 Rapid Generation of Likely Pose Hypotheses

In [15] we have proposed a highly efficient method for generating likely pose hypothesis by using 'spin-tables'. The following improved approach considerably accelerates the run-time of the hole matching algorithm. In our experiments we observe an acceleration factor of 40 to 100.

How long does it take to find two corresponding point pairs (one for A and one for B)? Assume, that we have two identical surfaces, each with  $n$  surface points. Having chosen a point pair of A, the probability to select the corresponding point pair of B by chance is  $1/n^2$ . Thus, we have to compare an average of  $n^2 + 1$  point pairs, which results in an expensive run-time complexity of  $O(n^2)$ . But with a simple trick, the problem can be computed much faster:

Assume we alternately choose random point pairs of A and B, and store them in a hash table, using rotational invariants as table indices. Under the assumption that the invariants are unique, we only need to process an average of  $1.2 \cdot n$  pairs until a hash collision occurs. This will provide the much better run-time complexity of  $O(n)$ . This approach complies with the 'birthday attack' [19] - an efficient cryptological strategy to generate two different documents with similar digital signatures (hash values). Let us concretize the algorithm. Instead of a hash table, we use 4d relation tables (one per surface), and the four invariant relations (4) as table indices. This leads to the following search loop:

1. Randomly choose an oriented point pair  $\mathbf{a}, \mathbf{c} \in \mathcal{A}$  and calculate  $\text{rel}(\mathbf{a}, \mathbf{c})$ .
2. Insert the point pair into the relation table:  $R_A[\text{rel}(\mathbf{a}, \mathbf{c})] = (\mathbf{a}, \mathbf{c})$ .
3. Read out same position of the opposite relation table:  $(\mathbf{b}, \mathbf{d}) = R_B[\text{rel}(\mathbf{a}, \mathbf{c})]$ ; if there is an entry  $\Rightarrow$  new pose hypothesis  $(\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d})$ .
4. Randomly choose an oriented point pair  $\mathbf{b}, \mathbf{d} \in \mathcal{B}$  and calculate  $\text{rel}(\mathbf{b}, \mathbf{d})$ .
5. Insert the point pair into relation table:  $R_B[\text{rel}(\mathbf{b}, \mathbf{d})] = (\mathbf{b}, \mathbf{d})$ .
6. Read out same position of the opposite relation table:  $(\mathbf{a}, \mathbf{c}) = R_A[\text{rel}(\mathbf{b}, \mathbf{d})]$ ; if there is an entry  $\Rightarrow$  new pose hypothesis  $(\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d})$ .

These steps will be repeated until the hypothesis is good enough, all combinations are tested, or the time exceeds a predefined limit. Optionally the hypotheses selection in step 3 and 6 can be improved further by comparing local features; i.e. we only select hypotheses that satisfy  $feature(a) = feature(b)$  and  $feature(c) = feature(d)$ . In our experiments we use the local mean curvature, which enables us to reject over 95%. We found that 4d relation tables with  $32^4$  entries offer a good trade-off between accuracy and efficiency. Using  $2 \times 2$  bytes per entry, one relation table requires a reasonable memory capacity of four megabytes. The proposed algorithm offers a run-time complexity of  $O(n)$

for the first hypothesis, but since the relation tables get filled continuously, the complexity converges to  $O(1)$  for further hypotheses.

### 3.2 Fast Hypotheses Verification

After generating a pose hypothesis we must measure its matching quality. For this we adopt the approach of [15], where the proportion of overlapping area  $\Omega$  (where surface  $A$  is in contact with the opposite surface  $B$ ) is estimated. We assume that the surfaces are in contact at areas where the distances between surface points are smaller than some predefined  $\varepsilon$ . In contrast to [15] we do not have to consider fragment penetrations. Suppose that  $\mathbf{x}_1, \dots, \mathbf{x}_n \in A$  are independent random points. Let  $contact_B(x)$  be a function which determines whether a point  $\mathbf{x}$  is in contact with surface  $B$

$$contact_B(x) = \begin{cases} 1 & \text{if } dist_B(\mathbf{x}) < \varepsilon, \\ 0 & \text{else} \end{cases} \quad \text{with } dist_B(\mathbf{x}) = \min_{\mathbf{y} \in B} \|\mathbf{x} - {}^A\mathbf{T}_B \cdot \mathbf{y}\|. \quad (6)$$

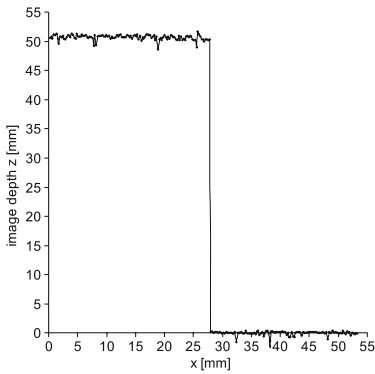
The function  $dist_B(\mathbf{x})$  returns the minimal distance of a point  $\mathbf{x}$  w.r.t. surface  $B$ . It can be implemented efficiently by using a *kd-tree* data structure (see [20]), which offers a logarithmical time complexity for the closest point search. Now  $\Omega$  can be approximated up to an arbitrary level of confidence. Considering the margin of error, for every additional random point, the approximation of  $\Omega$  can be recomputed as

$$\Omega \approx \frac{\sum_{i=1}^n contact_B(\mathbf{x}_i)}{n} \pm \frac{1.96}{2\sqrt{n}} \quad (7)$$

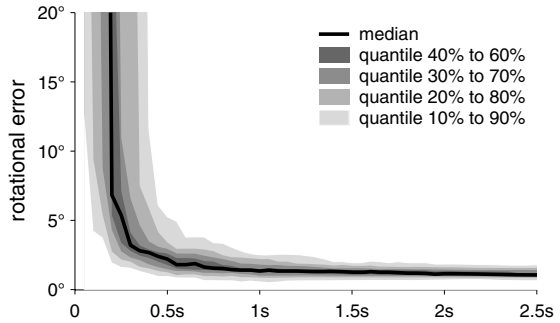
with a 95% level of confidence.  $\Omega$  can also be regarded as the probability that a random point  $x \in A$  is in contact with the opposite surface  $B$ . Thus  $\Omega$  can be forecasted by an efficient *Monte-Carlo* strategy using a sequence of random points, combined with a dropout if the upper bound of the confidence interval is considerably worse than the last best match. In this manner the quality estimation gets faster and faster, whenever the hypothesis is improved.

## 4 Experimental Results and Conclusion

For experimental evaluation, we used a grayscale CCD camera with XGA resolution, connected to a standard AMD-Athlon PC with 2.2 GHz. The scanning accuracy naturally depends on the exactness of the camera calibration and on the triangulation angle. To evaluate the accuracy of our laser scanner, we have scanned a well-known test object under a reasonable triangulation angle of about 30-35°, and a distance of 600 mm to the camera. The object's front surface consists of two planar faces with a 50.25 mm step in depth. The scan result contains this step within a tolerance of less than 0.4 mm. The measured (unfiltered) depth values of each surface are very accurate and show an RMS error of only 0.37 mm. Fig. 3 shows a scan line of the unfiltered depth values. The accuracy can be further improved by using appropriate time and space filtering (average and/or median).



**Fig. 3.** Scan line of two planar faces with a 50.25mm step in depth



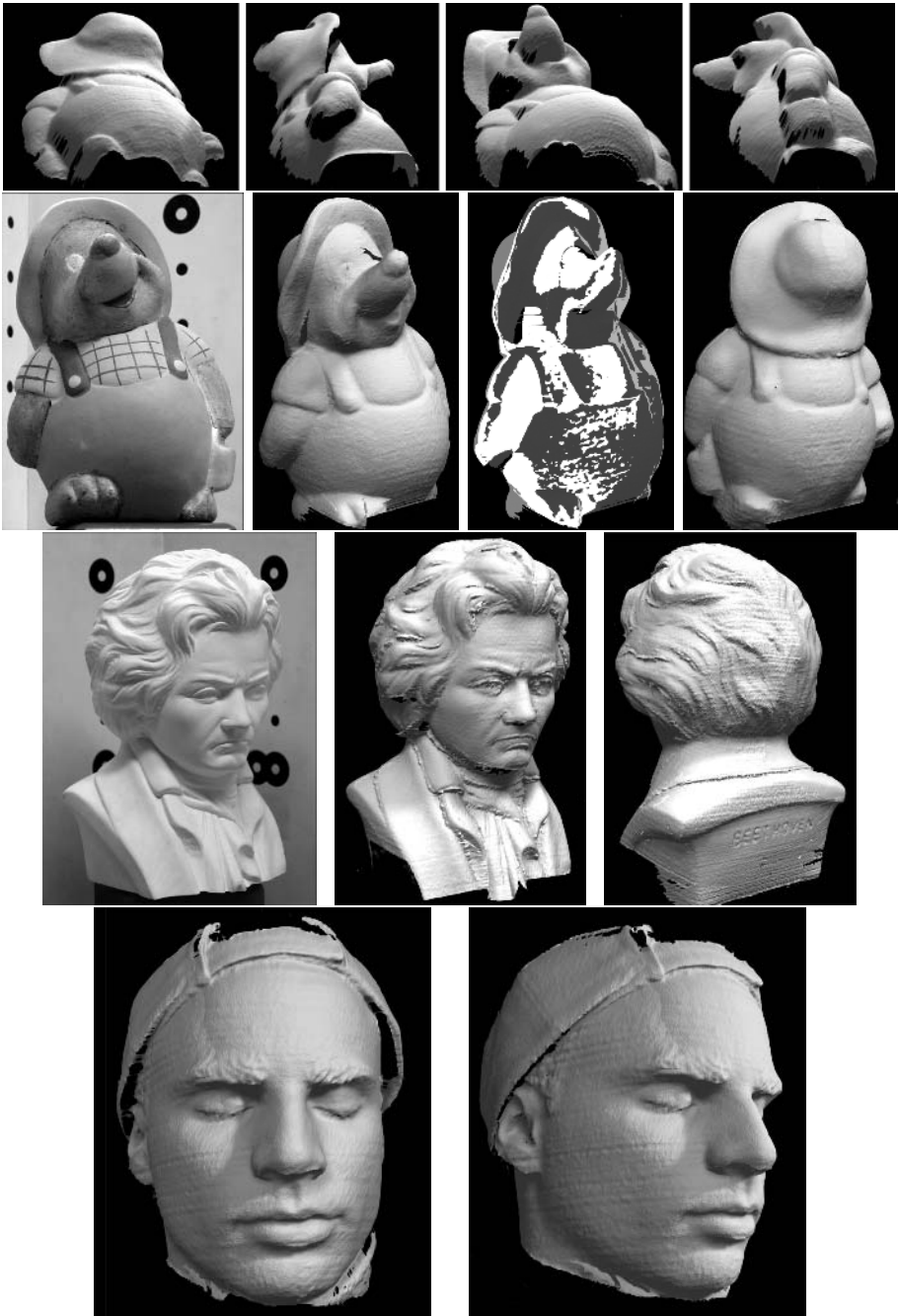
**Fig. 4.** Rotational registration error over time, median±quantiles of 100 test runs

Our surface registration approach performs very well with all of our test objects. Since the registration algorithm is a random process, we carried out 100 test series of each pairwise match. Fig. 4 shows the median rotational error over time in the case of two scans of the Beethoven bust (shown in Fig. 2). Both surfaces consist of approximately 60000 points with a surface overlap of 35%. As can be seen, after an execution time of only 0.5 seconds, 50% of all passes have already achieved a rotational accuracy of less than 2°. For accuracy evaluation we used a high-precision turn table, which provides the 'ground truth'. The outstanding registration efficiency is also documented in Table 1. The table presents a comparative study of the root mean square (RMS) distance, rotational error, and execution time between the registration approach in [15] and our substantial improvement. As can be seen, we achieve an impressive speed-up of factor 80 in this case. The results of all other test series are similar (factor 40–100 depending on data size and overlap).

Fig. 5 presents some result images of our scanning and registration method. In the first row one can see a side view of four scans of a clay mole (height 145 mm) under different angles of rotation of about 90° from one to the next. The second row shows a camera image of the mole and three images of the registration result of all four scans. In the third image of this row, the different scans have been colored in different shades of gray in order to show the surface interpenetration

**Table 1.** Registration performance of 100 test runs: comparison of root mean square (RMS) distance, rotational error (RotE), and execution time to achieve an RMS distance of less than 3 mm

	using approach of [15]				after our improvement			
	mean	median	min	max	mean	median	min	max
RMS [mm]	2.22	1.94	0.64	5.00	1.03	0.98	0.69	1.64
RotE [°]	2.36	2.23	0.37	6.16	1.10	1.03	0.11	2.09
Time [s]	28.85	28.05	0.60	>60	0.38	0.35	0.05	0.95



**Fig. 5.** (Row 1) scans of a test object. (Row 2) camera image; registered scans; each scan in a different color; back view. (Row 3) camera image and two views consisting of four registered scans. (Row 4) two views of a head consisting of two registered scans.



and the fusion edges. The third row shows a Beethoven bust (height 170 mm) – a camera image, and the registration result from two different sides. Although it is easily possible to texturize the scenes using the camera images, we only show the untexturized scans to demonstrate that even small shape details are acquired accurately. For example, the “BEETHOVEN” engraving on the back can be read very well although it is only 0.4 to 0.7 mm in depth. In the last row we present scan and registration results of one of the authors’ head. He was sitting on a chair in the corner of a room, the walls served as background for laser calibration. Between the two scans, the camera has not been moved, instead the person has turned his head.

The horizontal artifacts (stripes) that are visible on flat surfaces in most scan results are caused by slight inaccuracies during on-line laser calibration. They can be effectively reduced using space and/or time filtering. Gaps in the scanned surface appear at places the camera cannot see (“shadow”), when the corresponding area is too dark, or when it has been scanned so quickly that the laser did not intersect it in any image.

With these impressive results, we have demonstrated that simple low-cost equipment is sufficient to build up a system for 360°-object-reconstruction, which is superior to other techniques (e.g. scanning flexibility and registration efficiency).

## References

1. Pipitone, F.J., Marshall, T.G.: A wide-field scanning triangulation rangefinder for machine vision. *International Journal of Robotics Research* **2**(1) (1983) 39–49
2. Hall, E.L., Tio, J.B.K., MCPerson, C.A.: Measuring curved surfaces for robot vision. *Computer* **15**(12) (1982) 42–54
3. Blais, F.: Review of 20 years range sensor development. *Journal of Electronic Imaging* **13**(1) (2004)
4. Zagorchev, L., Goshtasby, A.: A paintbrush laser range scanner. *Computer Vision and Image Understanding* **101** (2006) 65–85
5. Besl, P.J., McKay, N.D.: A method for registration of 3-D shapes. *IEEE Trans. Pattern Anal. Machine Intell.* **14**(2) (1992) 239–258
6. Krebs, B., Sieverding, P., Korn, B.: A fuzzy icp algorithm for 3d free form object recognition. In: *International Conf. on Pattern Recognition*. (1996) 539–543
7. Dalley, G., Flynn, P.: Pair-wise range image registration: a study in outlier classification. *Comput. Vis. Image Underst.* **87**(1-3) (2002) 104–115
8. Chua, C.S., Jarvis, R.: Point signatures: A new representation for 3d object recognition. *International Journal of Computer Vision* **25**(1) (1997) 63–85
9. Papaioannou, G., Theoharis, T.: Fast fragment assemblage using boundary line and surface matching. In: *IEEE/CVPR Workshop on Applicat. of Computer Vision in Archaeology*. (1999)
10. Krebs, B., Korn, B., Wahl, F.M.: Plausibilistic preprocessing of sparse range images. In: *Proc. of the 8th Int. Conf. on Image Anal. and Processing*. (1995) 361–366
11. Johnson, A., Hebert, M.: Recognizing objects by matching oriented points. In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition*. (1997) 684–689
12. Schön, N., Häusler, G.: Automatic coarse registration of 3d surfaces. In: *Vision, Modeling, and Visualization 2005*. (2005)

13. Barequet, G., Sharir, M.: Partial surface matching by using directed footprints. In: 12th annual symposium on Computational geometry. (1996) 409–410
14. Silva, L., Bellon, O.R.P., Boyer, K.L.: Robust Range Image Registration Using Genetic Algorithms and the Surface Interpenetration Measure. Volume 60 of Machine Perception Artificial Intelligence. World Scientific (2005)
15. Winkelbach, S., Rilk, M., Schönfelder, C., Wahl, F.M.: Fast random sample matching of 3d fragments. In: Pattern Recognition, 26th DAGM Symposium. Volume 3175 of Lecture Notes in Computer Science., Springer (2004) 129–136
16. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM* **24**(6) (1981) 381–395
17. Tsai, R.Y.: An efficient and accurate camera calibration technique for 3d machine vision. In: IEEE Conf. Computer Vision and Pattern Recognition. (1986) 364–374
18. Wahl, E., Hillenbrand, U., Hirzinger, G.: Surflet-pair-relation histograms: A statistical 3d-shape representation for rapid classification. In: Proc. 4th International Conf. on 3-D Digital Imaging and Modeling (3DIM'03). (2003) 474–481
19. Weisstein, E.W.: Birthday Attack. (From MathWorld—A Wolfram Web Resource. <http://mathworld.wolfram.com/BirthdayAttack.html>)
20. Friedman, J.H., Bentley, J.L., Finkel, R.A.: An algorithm for finding best matches in logarithmic expected time. *ACM Trans. on Math. Software* **3**(3) (1977) 209–226

# A Point-Based Approach to PDE-Based Surface Reconstruction

Christian Linz<sup>1</sup>, Bastian Goldlücke<sup>2</sup>, and Marcus Magnor<sup>1</sup>

<sup>1</sup> Institut für Computergraphik  
TU Braunschweig

Mühlenpfordtstr. 23, 38106 Braunschweig, Germany

{linz, magnor}@cg.cs.tu-bs.de

<sup>2</sup> Graphics - Optics - Vision

Max-Planck-Institut für Informatik

Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany

bg@mpi.de

**Abstract.** Variational techniques are a popular approach for reconstructing the surface of an object. In previous work, the surface is represented either implicitly by the use of level sets or explicitly as a triangle mesh. In this paper we describe new formulations and develop fast algorithms for surface reconstruction based on partial differential equations (PDEs) derived from variational calculus using an explicit, purely point-based surface representation. The method is based on a Moving Least-Squares surface approximation of the sample points. Our new approach automatically copes with complicated topology and deformations, without the need for explicit treatment. In contrast to level sets, it requires no postprocessing, easily adapts to varying spatial resolutions and is invariant under rigid body motion. We demonstrate the versatility of our method using several synthetic data sets and show how our technique can be used to reconstruct object surfaces from real-world multi-view footage.

## 1 Introduction

Many interesting problems in computer vision can be formulated as minimisation problems of an energy functional given as a surface or curve integral over a scalar-valued weight function. The variational formulation of these kinds of problems lead to a curve or surface evolution PDE. Among the well-known variational methods successfully applied in computer vision are *Geodesic Active Contours* [3]. While originally designed for segmentation of objects in 2D it can be easily generalised to 3D [4]. Caselles et al., Zhao et al. and Savadijev et al. use this approach to model surfaces from unstructured point clouds [4,22,19]. Geodesic active contours were also employed for the detection and tracking of moving objects in 2D [16]. Furthermore, minimal surfaces may be employed for 3D reconstruction of static objects from multiple views, as proposed by Faugeras and Keriven [6].

All of these problems fit into one unifying framework [9]. There, a mathematical analysis of weighted minimal hypersurfaces is given in arbitrary dimension and for a general class of weight functions. An Euler-Lagrange equation is derived that yields the necessary minimality condition. As an application example, the static 3D reconstruction of a surface is generalised towards a global space-time reconstruction of the evolving surface [8]. A common feature of the aforementioned approaches is that object geometry is implicitly defined as the zero level-set of a function extending over the entire space. For an explicit representation, the object shape has to be extracted using marching-cube-like techniques [15] in a post-processing stage.

In contrast to the Eulerian approach, Duan et al. propose a PDE-based deformable model that takes the Lagrangian approach [5], i.e., shape and topology of the deformable object is always explicitly represented throughout the computation. The surface is typically represented as a triangle mesh. This model is used for surface reconstruction from volumetric images, point clouds and reconstruction from 2D multiple views. Goldlücke and Magnor recently also incorporated an explicit surface representation into their framework for space-time coherent reconstruction [7]. While this technique to solve PDEs directly yields an explicit representation of the solution, the topology information encoded in the mesh connectivity requires explicit handling when the surface topology changes. Complex local mesh operations such as the deletion and creation of edges, faces, or vertices render this approach hard to implement robustly.

Recently, purely point-based models have gained increasing popularity in traditional computer graphics as well as in the field of geometric modelling. Those models offer great flexibility since they neither store, nor have to maintain, any connectivity information. In this paper, we make use of this new modelling paradigm in the context of computer vision. Specially, we apply point-based geometry representation to the problem of PDE-based surface reconstruction. We unite several algorithms for point-based geometry processing under a common framework for PDE-based surface evolution. Our approach combines the implicit recovery of the surface topology inherent to level sets with the flexibility of a point based geometry representation stemming from the lack of connectivity information. In particular, our point-based PDE solver does not require any post-processing nor explicit handling of topology changes and easily adapts to varying spatial resolutions. Moreover, it is also invariant under rigid body motion while level sets are vulnerable to numerical diffusion under such circumstances.

The rest of this paper is organised as follows: Section 2 reviews some prerequisites concerning point-based geometry representation that are at the very heart of our work. In Sect. 3, we briefly review the mathematical framework of weighted minimal hypersurfaces and introduce our PDE-based surface reconstruction algorithm. We apply it to the problem of reconstructing surfaces from unstructured point clouds and deal with the problem of surface reconstruction from multiple views in Sect. 4. Section 5 concludes our work and presents some ideas for future work.

## 2 Review of Point-Based Models

This section briefly summarises some well-known algorithms from point-based modelling. Each of these algorithms is designed for its very special purpose, for example normal estimation or outlier detection. In the following sections, we unite them under a common framework for PDE-based surface evolution on point-based models.

**Point Sample Neighbourhood and Normal Estimation.** Our surface representation consists of an unstructured point cloud  $\mathcal{P}$  in 3D space, made up of  $n$  oriented disks that describe an underlying manifold surface  $S$ . Contrary to polygonal representations, in a point-based setting all local computations are based on spatial proximity between samples, instead of geodesic proximity and the known connections between mesh vertices. For dense samples and small Euclidean neighbourhoods, both notions are similar [2]. Geodesic neighbourhoods are proposed by Klein and Zachmann [12]. While this work more reliably estimates topologically correct neighbours, it is only applicable to static point sets since the computational cost for building the underlying datastructures is too high for the applications we have in mind. Therefore, our neighbourhood structure relies on the notion of the  $k$ -nearest neighbours with respect to the Euclidean distance, denoted  $\mathcal{N}_k$ , which was already successfully used in [18].

This neighbourhood structure can be computed efficiently using a hierarchical space partitioning technique, for example  $k$ D-trees.

Since normal vectors are not necessarily given a-priori or may change if the shape of the model changes, they have to be estimated by analysing the local neighbourhood of a sample point. As has been demonstrated in [11], a surface normal can then be estimated by performing an eigenanalysis of the covariance matrix of the local neighbourhood  $\mathcal{N}_k$ . The eigenvector with the smallest eigenvalue defines the least-squares plane through the centroid of the neighbourhood  $\mathcal{N}_k$  and can therefore serve as an approximation to the local surface normal.

**MLS Projection.** The set of oriented disks defining our model does not provide a mathematically smooth surface definition. To compute a smooth surface that approximates the sample points  $\mathcal{P}$ , Levin [13,14] introduces a projection operator based on a Moving Least-Squares (MLS) optimisation. This approach has first been applied to point-based geometry in  $\mathbb{R}^3$  by Alexa et al. [1]. The MLS projection takes a point  $\mathbf{r}$  in space and projects it onto a polynomial that locally approximates the underlying surface in the vicinity of  $\mathbf{r}$ . The computation of the polynomial can be split in two steps. First of all, a reference plane  $H$  is fitted to the surface samples around  $\mathbf{r}$  using a weighted least-squares optimisation. This reference plane provides a local parameterisation of the sample points and is used in a second least-squares fit to compute a bivariate polynomial. A global approximation is built by blending the local polynomials.

Both the computation of the reference domain and the polynomial approximation employ a radially symmetric Gaussian weighting function  $\theta(d) = e^{-d^2/h^2}$ . The parameter  $h$  corresponds to the anticipated spacing between neighbouring samples. In what follows, we always adapt the bandwidth to the local sampling density of the surface as proposed in [17].

**Surface Refinement.** The quality of the surface reconstruction from an unstructured point set heavily depends on the sampling density of the point set. If the object is undersampled, the reconstructed surface will not be able to recover details present in the original object. Based on the surface definition reviewed in the preceding paragraph, several methods for up- and downsampling of point sets have been proposed [1,17,18]. However, these methods are expensive due to the nature of the projection operator and typically generate oversampling. An algorithm that overcomes these problems was proposed by Guennebaud et al. [10]. We employ this method in our framework to ensure a sufficient sampling density.

To achieve a uniform distribution of all samples, we let neighbouring point samples repel each other. We use an algorithm for point relaxation introduced by Turk [20] for resampling a surface defined by polygons. This approach has been adapted to point-based geometries by Pauly et al. [17].

**Outlier Detection.** Noise and outliers are almost always present in a point-sampled geometry. Weyrich et al. proposed a set of fast heuristics to detect outliers in point sets [21]. The underlying criteria all deliver an estimator  $\chi(\mathbf{p}) \in [0, 1]$  which specifies the likelihood for a point sample  $\mathbf{p}$  to be an outlier. All criteria are solely based on the analysis of the  $k$ -nearest neighbours  $\mathcal{N}_k$  of  $\mathbf{p}$ . The final classification is then computed as a weighted average of the heuristics. The weighting of the criteria depends on the type of the underlying surface. We refer the interested reader to [21] for details on this.

**Overview.** In the next section, we integrate the point-based graphics tools reviewed in this section in a framework for PDE-based surface evolution algorithms. Using the point-based models, it is easier and more elegant to obtain a solution to these evolutions. In particular, our approach overcomes the need to keep the surface in a consistent manifold state as it is the case with evolution algorithms based on triangle meshes. Moreover, compared to implicit level set representations, the point-based surface easily adapts to varying spatial resolutions and may readily be rendered without the need for prior surface extraction.

### 3 PDE-Based Surface Reconstruction and Point-Based Models

We now turn to the mathematical framework we build our work upon. In [9], a mathematical analysis of weighted minimal hypersurfaces is given in arbitrary dimension and for a general class of weight functions. The aim is to find a  $k$ -dimensional regular hypersurface  $\Sigma \subset \mathbb{R}^n$  which minimises the energy functional

$$\mathcal{A}(\Sigma) := \int_{\Sigma} \Phi(\mathbf{s}) dA(\mathbf{s}). \quad (1)$$

We restrict the weight function  $\Phi$  to depend solely on the surface point  $\mathbf{s}$ . The necessary condition for a surface to be a minimum of this functional is to satisfy the Euler-Lagrange equation

$$\Psi := \langle \Phi_{\mathbf{s}}, \mathbf{n} \rangle - \text{Tr}(\mathbf{S}), \quad (2)$$

where  $\mathbf{S}$  is the shape operator of the surface. The result presented in [9] is more general in that the weight function may also depend on the surface normal  $\mathbf{n}$ . We do not consider this general case in this paper.

One of the fundamental questions in practise is how to solve the Euler-Lagrange equation (2). Only in a very limited number of simple cases can an analytic solution be derived directly. In all other cases, one has to numerically solve the surface evolution equation

$$\frac{\partial}{\partial \tau} \Sigma_\tau = \Psi \mathbf{n}, \tag{3}$$

where  $\Sigma_\tau$  represents the surface  $\Sigma \subset \mathbb{R}^n$  and  $\tau$  is the evolution parameter. If we start with an initial surface  $\Sigma_0$  and let it evolve using (3), it will eventually converge to a steady state, yielding a solution to the Euler-Lagrange equation.

We will now present our framework to solve (3) using a point-based approach. For validation, we first test our solver with a surface reconstruction from unorganised 3D sample points, which are distributed on synthetic objects whose geometry is precisely known. Our reconstruction technique and error function is similar to the work of Zhao et al. [22] and Caselles et al. [4], yet we are using the more general framework of Goldlücke and Magnor [9]. The target point cloud defines a point-based model in the sense of Sect. 2, and our surface evolution is implemented using a purely point-based model as well.

The error functional is modelled as the signed distance function  $\mathcal{D}(\mathbf{s})$  for each surface sample  $\mathbf{s}$  of the evolving surface to the closest point  $\mathbf{t}$  on the target surface:

$$\mathcal{A}(\Sigma) := \int_{\Sigma} \Phi(\mathbf{s}) dA(\mathbf{s}), \tag{4}$$

$$\text{where } \Phi(\mathbf{s}) := \mathcal{D}(\mathbf{s}). \tag{5}$$

The signed distance  $\mathcal{D}(\mathbf{s})$  from an arbitrary point  $\mathbf{s} \in \mathbb{R}^3$  to a known surface  $\Sigma$  is the distance between  $\mathbf{s}$  and the closest point  $\mathbf{t} \in \Sigma$ , multiplied by  $\pm 1$ , depending on which side of the surface  $\mathbf{s}$  lies. In a point-based setting, the distance is hence computed as

$$\mathcal{D}(\mathbf{s}) = (\mathbf{s} - \mathbf{t}) \cdot \mathbf{n}_{\mathbf{t}}, \tag{6}$$

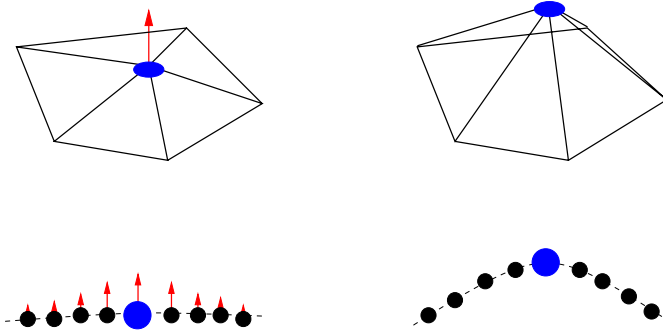
that is the normal component of the distance to the closest point on the target surface. In fact, we compute the signed distance function of  $\mathbf{s}$  to a local tangent plane in  $\mathbf{t}$ . Since our surface  $\Sigma$  is closed, this simple rule works well.

According to [5], we add an extra term to (3), yielding

$$\frac{\partial}{\partial \tau} \Sigma_\tau = [v\Phi + \langle \Phi_{\mathbf{s}}, \mathbf{n} \rangle - \text{Tr}(\mathbf{S})\Phi] \mathbf{n}. \tag{7}$$

The term  $v\Phi$  allows the model to capture arbitrary non-convex shapes and avoids that the model gets stuck into local minima during deformation.  $v$  is a constant velocity. Using Euler integration, we yield the following iterative formulation of the evolution process:

$$\mathbf{p}^{\tau+\Delta} = \mathbf{p}^\tau + \Delta \Psi \mathbf{n}_{\mathbf{p}}, \tag{8}$$



**Fig. 1.** Evolution on triangle meshes compared to evolution in a point-based setting. The evolution force  $\Psi$  (red vector) has to be distributed to the neighbouring sample points to achieve a behaviour comparable to triangle meshes. The dashed line illustrates the idealised MLS approximation of the sample points.

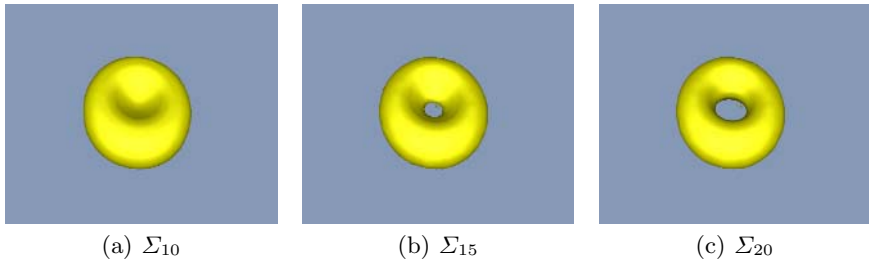
where  $\mathbf{p}^\tau$  denotes the position of the surface point of the deformable model  $\Sigma_\tau$  at time instant  $\tau$  and  $\mathbf{n}_\mathbf{p}$  its normal.  $\Delta$  denotes the time step and may be used to control the evolution speed.

We approximate the value  $\text{Tr}(\mathbf{S})$  in (7) by the mean curvature values obtained from the MLS approximation presented in Sect. 2. Likewise,  $\langle \Phi_\mathbf{s}, \mathbf{n} \rangle$  is approximated using fourth-order accurate central differences. Therefore, we first compute the one-ring neighbourhood of a sample  $\mathbf{s}$  and displace the entire neighbourhood structure by a fixed amount in positive and negative normal direction. Since the evolution along the signed distance function does not yield a uniform point distribution and, moreover, often produces undersampled regions, we apply the upsampling scheme in combination with the point relaxation introduced in Sect. 2 to the evolving surface to ensure a good surface approximation in the next iteration. Moreover, we detect outliers that originate from an overshooting evolution force using the heuristics introduced in Sect. 2 to avoid incorrect normals and curvature values. Since the evolution is based on these values, errors would also amplify during the iterative process. Compared to evolutions on triangle meshes, we have to take care that the per-point evolution force  $\Psi$  also affects the sample points in a small neighbourhood, Fig. 1. By weighting the forces using a Gaussian kernel, we are able to mimic an evolution behaviour similar to triangle meshes.

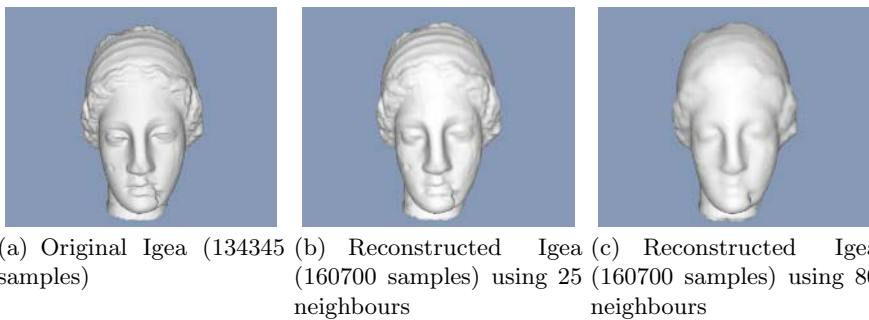
## 4 Results and Applications

We validate our point-based approach using several models. As a first step, we choose a model of a torus that requires an explicit topology change, Fig. 2. The surface topology is recovered implicitly by the MLS surface approximation without the need for additional operations. Using an explicit surface representation with connectivity information such as triangle meshes would have required complex local mesh operations which render this approach hard to implement





**Fig. 2.** Shape recovery with implicit topology change. The initial point surface  $\Sigma_0$  was a sphere surrounding the target surface.



**Fig. 3.** Reconstruction with varying spatial resolutions: The original model is shown on the left, a reconstruction based on a small neighbourhood is depicted in the middle. The reconstruction on the right uses a larger neighbourhood. The initial points are distributed on an sphere enclosing the geometry.

robustly. Figure 3 shows the results of the evolution on a more complex model. It also shows the adaptivity of the point-based model to different spatial resolutions by a varying interpolation radius, determined by the size of the neighbourhood structure. Higher spatial resolutions are easily obtained by placing more sample points in the desired regions. Grid-based level sets on the contrary require a more complex restructuring of the underlying grid. In both cases, the initial point surface  $\Sigma_0$  was an appropriately scaled sphere surrounding the object.

In a second step, we use our point-based PDE solver to reconstruct real-world object geometry from multiple 2D images. First, however, we need some additional notation for colour and visibility of surface samples. Let  $I_k$  denote the image associated with camera  $k$ . Each camera projects the scene onto the image plane via a fixed projection of the form  $\pi_k : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ . Then,  $I_k \circ \pi_k(\mathbf{s})$  denotes the colour of the projection of  $\mathbf{s}$  into the image taken by camera  $k$ . For each surface point  $\mathbf{s} \in \mathbb{R}^3$ , let  $\nu_k(\mathbf{s})$  denote whether  $\mathbf{s}$  is visible in camera  $k$  in the presence of a surface  $\Sigma$  or not. An error measure, taking care of photo-consistency of the evolving surface with the input images, can now be defined as

$$\Phi^C(\mathbf{s}) := \frac{1}{|\mathcal{V}_s|(|\mathcal{V}_s| - 1)} \sum_{i,j=1}^l \nu_i(\mathbf{s})\nu_j(\mathbf{s}) \cdot \chi_{i,j}(\mathbf{s}, \mathcal{N}_k) \quad (9)$$

$$\chi_{i,j}(\mathbf{s}, \mathcal{N}_k) := \frac{1}{|\mathcal{N}_k|} \sum_{\mathbf{q} \in \mathcal{N}_k} ((I_i \circ \pi_i)(\mathbf{q}) - \bar{I}_i^{\mathcal{N}_k}) \cdot ((I_j \circ \pi_j)(\mathbf{q}) - \bar{I}_j^{\mathcal{N}_k}). \quad (10)$$

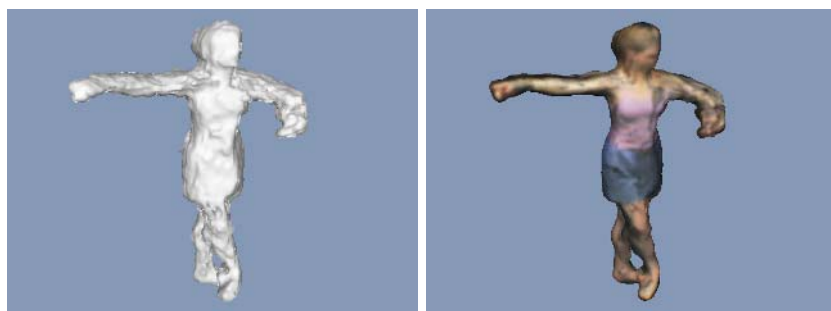
$\bar{I}_i^{\mathcal{N}_k}$  denotes the mean colour value in the  $k$ -neighbourhood  $\mathcal{N}_k$  of a surface sample. This functional is a reasonable discretization of the error functional introduced in [9] for point-based models.

Using this definition of the error functional, we are able to reconstruct the surface of an object given multiple views. We test our method on multi-view footage of a dancer, recorded from 8 cameras distributed around the scene. All input images are segmented into foreground and background using a thresholding technique. The results obtained with our point-based approach on a fixed frame of the dancer sequence are shown in Fig. 4. Our point-based PDE solver clearly smoothes the initial surface  $\Sigma_0$  obtained from a space-carving approach and improves photo-consistency. Compared to the approach taken in [8], the use of a point-based model gives comparable results at lower implementation complexity since explicit handling of topology changes is completely avoided.

## 5 Summary and Conclusions

In this paper, we have introduced a purely point-based technique to reconstruct explicit surfaces from implicit PDE definition. We demonstrated that this representation in combination with the powerful Moving Least-Squares surface approximation unites the advantages of a level set-based representation, i.e., implicit recovery of surface topology, with direct accessibility of an explicit model based on triangle meshes. Our representation does not depend on an underlying grid-structure and hence easily adapts to varying spatial resolutions and is invariant under rigid body motions. We showed the general applicability of point-based geometry representation to surface reconstruction using synthetic data sets as well as real-world data. Compared to a direct representation based on triangle meshes, the point-based model used in our work is more flexible, especially when topology changes are involved. It is thus the more natural choice for iterative surface evolution. Compared to level set-based surface representations, the point-based models are far less memory-consuming. Our, yet unoptimised, point-based implementation already outperforms a similar implementation using level sets.

We believe that with growing interest in point-based models in the research community, the flexibility of this surface representation will be exploited for various tasks in computer vision. One could, for example, extend the implementation of reconstruction from multiple views described in Sect. 4 to reconstructions in space-time as has been outlined in [7,8]. Furthermore, a detailed analysis of the convergence properties and a quantification of the approximation quality as compared to grid-based level sets would be helpful. Also, the computational complexity of our approach needs more investigation.



(a) Initial model



(b) Final result after 180 iterations

**Fig. 4.** Initial and final point set for a fixed frame, coloured with per-point colour information derived from the best two cameras. Photo-consistency has clearly improved in the final result as can be judged by the decrease of black and grey areas on the arms of the dancer.

## References

1. Marc Alexa, Johannes Behr, Daniel Cohen-Or, Shachar Fleishman, David Levin, and Claudio T. Silva. Computing and rendering point set surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 9(1):3–15, 2003.
2. Nina Amenta, Marshall Bern, and Manolis Kamvyselis. A new Voronoi-based surface reconstruction algorithm. In *Proc. SIGGRAPH '98*, pages 415–421, New York, NY, USA, 1998. ACM Press.
3. Vincent Caselles, Ron Kimmel, and Guillermo Sapiro. Geodesic active contours. In *Proc. ICCV*, pages 694–699, 1995.
4. Vincent Caselles, Ron Kimmel, Guillermo Sapiro, and Catalina Sbert. Minimal surfaces based object segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4):394–398, 1997.
5. Ye Duan, Liu Yang, Hong Qin, and Dimitris Samaras. Shape Reconstruction from 3D and 2D Data Using PDE-Based Deformable Surfaces. In *Proc. ECCV (3)*, volume 3023 of *LNCS*, pages 238–251. Springer, 2004.
6. O. Faugeras and R. Keriven. Variational principles, surface evolution, PDE's, level set methods and the stereo problem. In *IEEE Transactions on Image Processing*, volume 3, pages 336–344, 1998.

7. Bastian Goldlücke and Marcus Magnor. Spacetime-continuous Geometry Meshes from Multiple-View Video Sequences. In *Proc. IEEE International Conference on Image Processing (ICIP'05)*, Genoa, Italy, 2005. accepted.
8. Bastian Goldlücke and Marcus Magnor. Space-time isosurface evolution for temporally coherent 3d reconstruction. In *Proc. CVPR*, volume I, pages 350–355, Washington, D.C., USA, July 2004.
9. Bastian Goldlücke and Marcus Magnor. Weighted minimal hypersurfaces and their applications in computer vision. In *Proc. ECCV (2)*, volume 3022 of *Lecture Notes in Computer Science*, pages 366–378. Springer, 2004.
10. Gael Guennebaud, Loc Barthe, and Mathias Paulin. Real-Time Point Cloud Refinement. In *Symposium on Point-Based Graphics*, pages 41–49, 2004.
11. Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. Surface reconstruction from unorganized points. In *Proc. SIGGRAPH '92*, pages 71–78. ACM Press, 1992.
12. Jan Klein and Gabriel Zachmann. Point Cloud Surfaces using Geometric Proximity Graphs. *Computers & Graphics*, 28(6):839–850, 2004.
13. David Levin. The approximation power of moving least-squares. *Math. Comput.*, 67(224):1517–1531, 1998.
14. David Levin. Mesh-independent surface interpolation. In *Geometric Modeling for Scientific Visualization*, pages 37–49. Springer Verlag, 2003.
15. William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. In *Proc. SIGGRAPH '87*, pages 163–169, New York, NY, USA, 1987. ACM Press.
16. Nikos Paragios and Rachid Deriche. Geodesic active contours and level sets for the detection and tracking of moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(3):266–280, 2000.
17. Mark Pauly, Markus Gross, and Leif Kobbelt. Efficient Simplification of Point-Sampled Surfaces. In *VIS '02: Proceedings of the conference on Visualization '02*, pages 163–170. IEEE Computer Society, 2002.
18. Mark Pauly, Leif Kobbelt, and Markus Gross. Multiresolution Modeling of Point-Sampled Geometry. Technical Report 378, Computer Science Department, ETH Zurich, Switzerland, Computer Science Department RWTH Aachen, Germany, September 2002.
19. Peter Savadkijev, Frank P. Ferrie, and Kaleem Siddiqi. Surface Recovery from 3D Point Data Using a Combined Parametric and Geometric Flow Approach. In *Proc. EMMCVPR*, volume 2683 of *LNCS*, pages 325–340. Springer, 2003.
20. Greg Turk. Re-tiling polygonal surfaces. In *Proc. SIGGRAPH '92*, pages 55–64, New York, NY, USA, 1992. ACM Press.
21. Tim Weyrich, Mark Pauly, Simon Heinzle, Richard Keiser, Sascha Scandella, and Markus Gross. Post-processing of Scanned 3D Surface Data. In *Symposium on Point-Based Graphics*, pages 85–94, 2004.
22. Hong-Kai Zhao, Stanley Osher, and Ronald Fedkiw. Fast surface reconstruction using the level set method. In *VLSM '01: Proceedings of the IEEE Workshop on Variational and Level Set Methods*, page 194, Washington, DC, USA, 2001. IEEE Computer Society.

# Robust Feature Representation for Efficient Camera Registration

Kevin Köser, Volker Härtel, and Reinhard Koch

Institute of Computer Science and Applied Mathematics  
Hermann-Rodewald-Str. 3,  
24098 Kiel, Germany  
{koeser, vhaertel, rk}@mip.informatik.uni-kiel.de

**Abstract.** This paper shows an approach for automatic learning of efficient representations for robust image features. A video sequence of a 3D scene is processed using structure-from-motion algorithms, which provides a long validated track of robust 2D features for each tracked scene region. Thus each tracked scene region defines a class of similar feature vectors forming a volume in feature space. The variance within each class results from different viewing conditions, e.g. perspective, lighting conditions, against which the feature is not invariant. We show on synthetic and on real data that making use of this class information in subspace methods, a much sparser representation can be used. Furthermore, less computational effort is needed and more correct correspondences can be retrieved for efficient computation of the pose of an unknown camera image than in previous methods.

## 1 Introduction

Registering a camera against a previously learned scene requires an efficient representation of the scene content for lookup. We show how feature-based camera pose computation can benefit from a class-based representation of learned scene features: During an offline phase, a camera is moved within a scenario or around an object. By using structure from motion techniques [9] we track a natural feature across many images and process its appearances, i.e. we learn how a feature typically changes (e.g. due to perspective, light). The camera trajectory as well as the 3D locations in the scene are reconstructed and the image appearances (2D features) of the 3D locations are learned and organized in a database optimized for fast lookup and high recognition rates, which are main new contributions.

Such robust or invariant 2D features have been quite a busy research topic during the last years. However, they have usually been studied in 2D environments or on planes only [7], because an objective evaluation in cluttered 3D scenes is difficult. As a difference compared to previous feature evaluations, we propose the following method: In the online phase a database query is performed for every 2D feature of a camera image returning the likeliest learned 3D feature class. We estimate the camera pose based on these 2D-3D correspondences using robust estimation techniques. Afterwards we check how many of the 2D-3D

correspondences are inliers, which is a quite natural criterion of the fitness of the feature representation, since it directly resembles the pose estimation problem.

The paper is structured as follows: In the next section we compare different image registration techniques as well as we briefly review the work on feature evaluation. Afterwards we show the steps of our method, scene learning and feature organisation in the offline phase and registration in the online phase. In the last section we compare results of different representations.

## 2 Previous Work

Early marker-less approaches on registering views in Augmented Reality scenarios tried to compute orientation only, e.g. using the Fourier-Mellin-Transform [4]. This is a global transformation of the whole image, which breaks if different parts of the image undergo different perspective distortions, clutter and occlusion. To solve for this, various local features have been studied during the last years. They have been applied successfully for image-to-image matching, panorama registration and more recently also 6DOF camera pose computation [3]. The principle is that a particular detector/descriptor combination produces the same - or a slightly different - *feature vector* for the same 3D region under different conditions, while producing other vectors for differently looking regions. The feature vector can be interpreted as a signature. Among all the detector/descriptor pairs, DoG/SIFT [5] is known to perform well [7] and can be computed quite fast. Although being invariant only against scale, rotation and affine brightness change of a 2D image, it is *robust* against mislocation, perspective effects and several other distortions. Robust means that small violations of the invariance assumptions will cause only small disturbances of the feature vector. In that case the feature vectors occupy a small continuous area in the feature space. This makes it well-suited for our purposes and we will use it throughout this paper, though the proposed techniques can also be applied to other features with such properties.

For a 2D feature to provide significant information to discriminate it from others, the descriptions must be quite high-dimensional. On the other hand, when one seeks to find a similar feature vector in the space of all possible features, we run into the curse of dimensionality if the description vector is too large. One way of organizing points in high dimensional spaces is space-partitioning using kd-trees[8]. For the typical SIFT feature dimension of 128 a complete binary space partitioning would create a tree with  $2^{128}$  (more than  $10^{38}$ ) leaves. Therefore we compare different methods of learning the relevant parts of the high-dimensional feature descriptions, which have been applied successfully in face recognition [2] and other classification tasks, often however only on the raw image signal: Multiple discriminant analysis (“fisher-faces”) and principle component analysis (“eigen-faces”) and show the advantages over the technique of using vector entries with largest variance, which is a common feature space matching technique today [3]. In contrast to PCA-SIFT [6], we are not interested in the subspace of all feature descriptions a DoG/SIFT operator can produce

on the set of all images ever possible. This encodes what all descriptors do have in common. We explicitly want to learn what is *different* between the clusters of features *in our scene*. The learning is deliberately based on the knowledge that different representatives belong to the same class like in [13], while we do not only seek for one representative per class but we also look for a transformed small representation to make a fast distinction between the classes possible.

In that sense the idea is somewhat related to the Randomized Trees approach [10], which does not rely on high-level features but on massive simple tests. Instead of performing a nearest neighbor search in one tree and applying a decision, they propose a soft-classification by using several trees, where each tree node encodes class probabilities. The final classification is performed by combining the probabilities. While this is an interesting approach in the handling of the probabilities, the authors have proposed it only for recognition and pose computation of single objects, presumably because the simple decisions made in the trees sacrifice discriminative power for the sake of speed. They have not evaluated whether the approach does also extend to larger scale scenarios.

### 3 Scene Database

To register a view in the online phase we use a database of features which has to be set up offline. To create the database we take an image sequence of the scene and use a feature-based structure from motion system similar to [9] to calibrate the images and to reconstruct 3D points from corresponding 2D interest points. Each interest point is assigned a descriptor (in the simplest case the grey values of the image region around the point or, more complex, the SIFT descriptor), which can be represented as a high-dimensional vector.

#### 3.1 3D Features

If the descriptors for corresponding 2D points do not vary too much across several images, we can assume that the invariance/robustness properties of the feature type are still satisfied, e.g. for SIFT features that the 2D image regions are projections of a three dimensional locally continuous surface from similar viewpoints and that all projections of this surface result in similar descriptors. We call the surface a 3D feature and assign it a 3D point. However, we do not want to reconstruct the surface here but the class of descriptors it produces, which is a novelty compared to previous approaches. They form a continuous area in descriptor space and their differences are e.g. due to small localisation distortions or transformations against which the descriptor is not completely invariant. Combining incremental structure from motion (in contrast to the reference image technique of [3]) allows to process long image sequences with lots of descriptor measurements.

If each class of descriptors covers a coherent and relatively small part in the high-dimensional descriptor space, and any two distinct classes are at different locations in this space, we can view the matching process as a classification

problem. For each 2D feature detected in the online phase we try to find the best matching class in descriptor space. Beis and Lowe proposed an approximate nearest neighbor search on a kd-tree partitioning the descriptor space [8]. The partitioning should at best represent the distribution of the various classes, therefore some parts in the feature space are more interesting than others. To traverse a balanced binary tree of depth  $d$  (e.g.  $d = 15$ ) we have to pass  $d$  decision hyperplanes, which divide the feature space. This tree has  $2^d$  leaves (distinct areas in feature space). If  $d$  is too large (for instance the original vector size 128), this leads to an unmanageable number of bins ( $2^{128}$ ). Even for depths not much larger than 20, the tree is over-fitted and only sparsely populated, unless one uses a huge number of features. For a small  $d$  on the other hand the question is extremely important, which is the best partitioning of the space and what are good dividing hyperplanes. Beis and Lowe solve the problem by computing the variance of each descriptor dimension across all features and select only the most variant entries. Instead, we propose to apply classical methods of dimensionality reduction from pattern recognition. These methods are compared next.

### 3.2 Dimensionality Reduction

From the offline phase we have many 3D features, which we saw in several images. Each 3D feature defines a class with mean and scatter in feature space. Let  $D_c^i \in \mathbb{R}^h$  be the  $i$ th (of  $n_c$ ) descriptor vector for class  $c$  (of a total of  $n$  classes). Since it has  $h$  entries, we have an  $h$ -dimensional descriptor space (e.g. for SIFT typically  $h = 128$ ). We want to find the reduction transformation  $R(D_c^i) = d_c^i : \mathbb{R}^h \mapsto \mathbb{R}^l$ , which shrinks our descriptor to a low dimension number  $l$  (e.g.  $l = 15$ ). However, the descriptor should not lose too much discriminative information needed for matching.

**Principle Components Analysis.** The most popular approach to dimensionality reduction is principle component analysis (PCA). PCA computes the mean and scatter of all descriptors (see [11]). We define different means as follows:

$$\mu_c = \frac{1}{n_c} \sum_i D_c^i \qquad \mu = \frac{1}{\sum_c n_c} \sum_c \sum_i D_c^i \qquad \mu_{Means} = \frac{1}{n} \sum_c \mu_c \quad (1)$$

$$\Sigma = \sum_c \left( \sum_i ((D_c^i - \mu)(D_c^i - \mu)^T) \right) \quad (2)$$

The principal components are now the eigenvectors of  $\Sigma$  according to [11]:

$$\Sigma \mathbf{e}_\Sigma^j = \lambda_\Sigma^j \mathbf{e}_\Sigma^j \quad (3)$$

where  $\mathbf{e}_\Sigma^j$  are sorted according to their eigenvalues  $\lambda_\Sigma^j$ ,  $\lambda_\Sigma^0$  being the largest. Let  $\hat{\mathbf{e}}_\Sigma^j = \frac{\mathbf{e}_\Sigma^j}{\sqrt{\lambda_\Sigma^j}}$ . Finally, we define the reduction transformation for PCA as:

$$R_{PCA}(D_c^i) = (\hat{\mathbf{e}}_\Sigma^0 \hat{\mathbf{e}}_\Sigma^1 \dots \hat{\mathbf{e}}_\Sigma^l)^T (D_c^i) \quad (4)$$



A slight modification of PCA, which we call PCA-Means takes into account classes and is computed only using the means of the classes, which gives an equal weight to each class and does not prefer strongly populated classes over small ones. The only difference in computation is that we replace equation (2) by equation (5), where  $\Sigma_{Means}$  is also called the inter class scatter matrix:

$$\Sigma_{Means} = \sum_c ((\mu_c - \mu_{Means})(\mu_c - \mu_{Means})^T) \quad (5)$$

Compared to classical PCA definition, the mean is neglected in our PCA reduction methods (eq. 4). However, since our reduction transformation is linear, the mean also transforms linear and introduces a constant offset for all features, which can be ignored since we are only looking for the nearest neighbor.

PCA is designed to minimize the reconstruction error, therefore it is suitable for compression and un-compression of similar vectors in high-dimensional space. However, it does not account for classes and does not aim at preserving separability of vectors in reduced space. In other words, PCA preserves what is common between two classes, not what is different. The goal of finding a linear transformation that maximizes class separability is the topic of discriminant analysis.

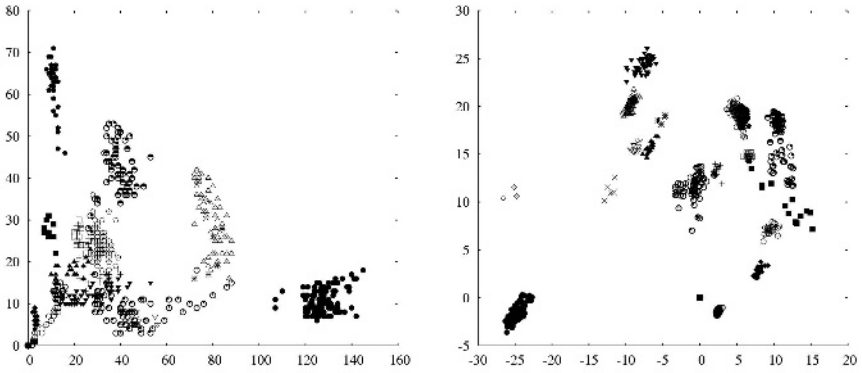
**Multiple Discriminant Analysis.** We propose an extension of multiple discriminant analysis (MDA) [11], which falls back smoothly to PCA in case only sparse within class information is available. The idea of MDA is to represent each class of descriptors by a mean and scatter and find a transformation  $R$  that minimizes within class scatter while maximizing the scatter of all class means. The within class scatter  $\Sigma_c$  and the total scatter matrix  $\Sigma_{total}$  (imagine as an average within class distribution) are defined as:

$$\Sigma_c = \frac{1}{(n_c - 1)} \sum_i ((D_c^i - \mu_c)(D_c^i - \mu_c)^T) \quad \Sigma_{total} = \frac{1}{n} \sum_c \Sigma_c$$

The rows of the reduction transformation matrix are the solutions  $\mathbf{e}^j$  to the generalized eigenvalue problem [11]:

$$\Sigma_{Means} \mathbf{e}^j = \lambda^j \Sigma_{total} \mathbf{e}^j \quad (6)$$

If  $\Sigma_{total}$  is nonsingular, the system can be converted to a standard eigenvalue problem like equation (3). However, particularly when  $\Sigma_{total}$  is estimated from few samples in the high-dimensional space, it will be singular, mainly because of missing data. A full rank can be enforced by applying ridge regularization [12] to the total scatter matrix, i.e. we add  $diag(\sigma^2)$  (a diagonal matrix with entries  $\sigma^2$ ). Small values of  $\sigma$  do not affect the shape of  $\Sigma_{total}$ , while larger ones make the diagonal dominate the matrix and very large values make it in fact a multiple of the identity matrix. In that case, equation (6) is the same as equation (3) for PCA-Means, therefore the value of sigma controls between MDA and pure PCA behavior. Since we do not want to lose within class shape information, we compute a minimum noise level, as the smallest existing eigenvalue of equation (6). This leads to a smooth transition from PCA-Means to MDA as soon as within class scatter is available.



**Fig. 1.** Exemplary distribution of features of 20 classes (randomly chosen) of a real video sequence of 400 images projected to the first two axes (left), the first two MDA axes (right). MDA representation shows more distinct local clusters .

**Most Variant Entries.** The approach chosen by Beis and Lowe [8] can also be viewed in the context of dimensionality reduction. They compute the variance for each vector entry separately. This corresponds to only taking into account the diagonal elements of  $\Sigma$  of equation (2) and sort vector components by these values. By disregarding the off-diagonal elements, the relations between the vector entries are thrown away. This is suboptimal for descriptors whose components are correlated, which is certainly the case for the SIFT descriptor, because the soft-binning technique distributes gradients into different vector entries upon mislocalization. In other words, the entries of the SIFT descriptor are not uncorrelated as a strictly diagonal scatter matrix would imply. The resulting reduction matrix is a pure permutation of the columns of the identity matrix.

### 3.3 Database Representation

Using a transformation of the previous section we can transform the original vectors into a space where the dimensions are sorted by importance. We build a kdtree in that space and choose the depth  $d = \log_2(c)$  such that in average each bin holds a class. In our novel transformed space we follow the method of [8].

Once the database is set up, we can extract 2D features from an unknown image. Each feature is transformed according to our reduction and traverses the tree using the backtracking strategy [8] until a better match in reduced space cannot be found, a maximum error in reduced space is reached or - if real-time is an issue - a (constant) maximum number of comparisons has been reached. The best match so far or “no match” is returned. The “no match” statement is particularly important because it decreases the false positive rate. Fewer outliers again speed up robust pose computation.

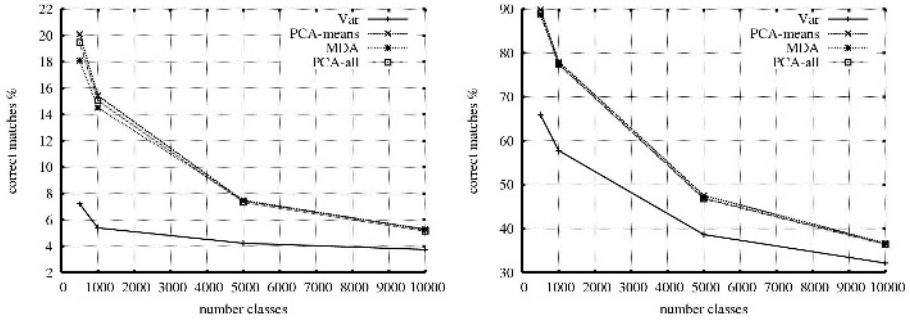
## 4 Experiments

In this section we show results on synthetic data as well as on videos of 3D scenes and compare the different reduction techniques, particularly PCA, the largest variance method [3] and ridge-regularized MDA.

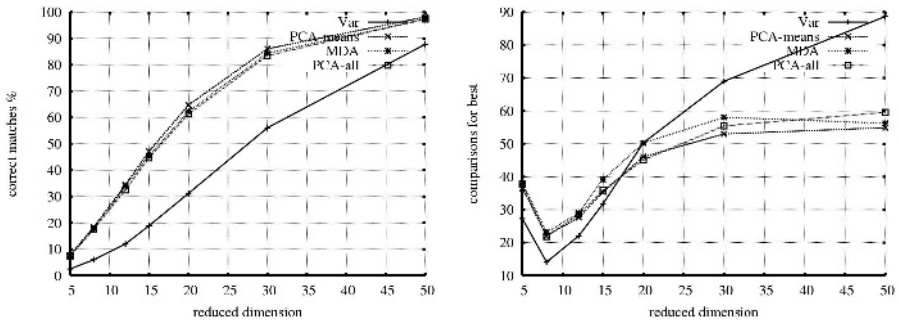
**Synthetic Data.** By empirical tests on real images of several scenes, we found that the SIFT descriptors of most 3D features do indeed cluster. For a series of feature tracks we found that in our scenarios the descriptors (vectors from  $[0; 200]^{128}$ ) of the same 3D feature usually have a maximum standard deviations of up to 15 (in principal component direction), seldomly higher. As pointed out before the entries of the SIFT descriptor are not independent if we consider the unavoidable 3D scale space mislocation (because of the soft-binning). Therefore, when creating synthetic data, we do not enforce independence of these components when creating synthetic features. The class of descriptors referring to a 3D feature is simulated by an anisotropic gaussian distribution in 128D with standard deviation along the principal axes of  $\sigma_j \in [1; 15]$ . The class means are sampled from a uniform distribution across the whole feature space  $[0; 200]^{128}$ .

We compare the average number of comparisons until the nearest neighbor has been found and the percentage of correctly classified points. As can be seen in figure (3) PCA, PCA-Means and MDA represent the data in a way that more features are classified correctly than in the variance method. Furthermore they need less comparisons on average, which means that search can be stopped after a lower fixed total number of comparisons in the tree. This improves camera registration speed. MDA and PCA methods do perform nearly the same, however it seems that MDA suffers slightly from overfitting onto the uncertain covariance matrices caused by the few measurements per class. If we generate the synthetic data in a way that the  $\Sigma_{total}$  is not isotropic but that all  $\Sigma_c$  are slightly biased towards a main orientation, MDA outperforms all other methods.

**3D Scenes.** Using the structure from motion approach similar to [9] we set up several 3D scenes. The left part of figure (4) shows a rendered sequence with real texture, for which the camera poses are perfectly known but for which feature positions in 3D space have been created by structure from motion. The right part of figure (4) shows a livingroom scene, for which real images sequences have been captured and reconstructed. We perform an exhaustive search in the feature space to evaluate the database, therefore we always get the nearest neighbor in transformed space. A feature is defined to be correctly classified if the 3D point is projected by the camera pose within 5 pixel distance to the detected 2D feature. We give only the mean number of correct 2D-3D matches as opposed to a mean percentage because the unknown test views produce features which have not been seen before, therefore the fraction is less significant than the total number. Figure (4) shows that MDA performs slightly better than the PCA methods but clearly better than variance and that a reduction to 15 dimensions can be successfully applied not only without loss of matches but also with improved speed, since we can work on the smaller vectors.

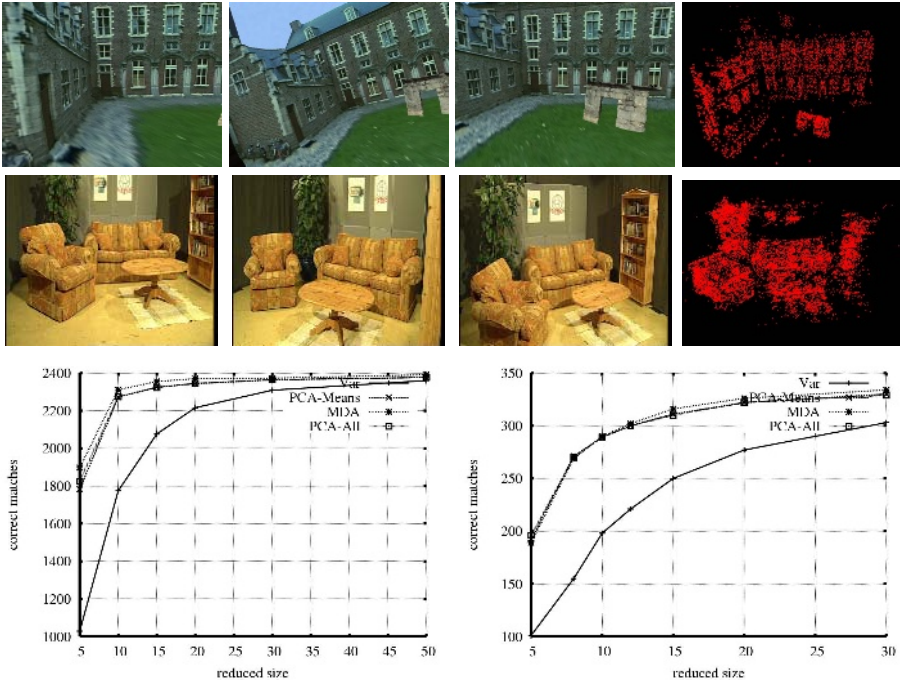


**Fig. 2.** Percentage of correctly classified synthetic features after exhaustive search in reduced space. Left: All features were placed in the tree. Right: Only class means were placed in the tree.



**Fig. 3.** Left: Percentage of correctly classified synthetic features after exhaustive search in reduced space using 500 classes of 20-50 features each. Right: Average number of comparisons needed until nearest neighbor was found.

To evaluate the quality of the representation for real-time pose estimation applications, we limit the number of comparisons per feature to 50 and compare our methods to the method used in [3] (Var). Using a fixed reduced vector size of  $l = 20$  leads to about the same number of correct matches in our scenario. For a fair comparison of runtime, we set  $l = 20$  and count the total number of basic comparisons the cpu had to perform on the sofa scenario in order to compute tentative 2D-3D correspondences needed by a robust pose estimation algorithm. As can be seen in figure (5) (rows 3 and 4) our methods produce more correct correspondences, while requiring only 15 percent of operations. When reducing size to  $l = 10$  the number of correct matches reduces by about 10 percent, the number of operations decreases even to 8 percent. The amount of memory required is also drastically reduced in our method, since we only use low-dimensional representatives instead of all the high-dimensional ones. In this comparison it can be seen (upper two rows) that the number of correct matches is slightly lower when only the class means are used in the tree, this is



**Fig. 4.** Upper Row: Rendered (semi-artificial) castle sequence (640x480 pixels). 3D feature locations of 4612 classes. Lower Row: Real sofa sequence (352x288 pixels). 3D feature locations of 2748 classes. Lower left (castle) and right (sofa): Number of correct nearest neighbors in database depending on reduction, standard deviation is about 15 percent of measurement.

Method	mean num. comp.	std.dev. comp.	num. inliers
MDA	982	4	311.5
PCA-Means	964	9	314.2
MDA (all features)	972	10	336.9
PCA-Means (all features)	960	10	337.1
Var (all features)	6397	13	324.5

**Fig. 5.** Comparison of basic operations performed in the sofa scenario. Upper two rows: only means are stored in the tree. Lower three rows: all features stored in the tree.

presumably because of too few measurements per class resulting in inaccurate mean and scatter estimation. However, in certain applications using only one class representative might be preferable.

## 5 Conclusion

We proposed exploiting class information for representation of features in a database for fast retrieval. Furthermore we described a method for learning rel-

evant variances of robust features in a particular scene in which registration is desired. We used an inlier criterion for 2D-3D correspondences to evaluate the representation on real 3D environments which resembles the pose estimation problem. As expected, on feature vectors where components are not independent of one another (presumably as most features) PCA and MDA outperform maximum variance as used in the literature [3] [8] and allow reducing feature size by a large factor. We showed that using a sparse database rather than putting all features into the database classifies better if gaussian distributions can be assumed and estimated reliably. Future work should evaluate other class shapes, which might be represented by several clusters. MDA on the other hand did not show up as significantly better than the PCA methods on real images, maybe because of missing feature data for scatter estimation within the classes. This means that when speed is an issue in the offline phase, the PCA methods can be used without losing much performance, because the PCA transformation needs less time to be determined.

## Acknowledgments

This work has been partially funded by the European Union in project MATRIS IST-002013 ([www.ist-matris.org](http://www.ist-matris.org)).

## References

1. V. Lepetit, L. Vacchetti, D. Thalmann, P.Fua “Fully Automated and Stable Registration for Augmented Reality Applications” ISMAR2003, p. 93, The Second IEEE and ACM International Symposium on Mixed and Augmented Reality, 2003
2. P. Belhumeur, J. Hespanha, D. Kriegman “Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection” IEEE Transactions on PAMI, Special Issue on Face Recognition, 19(7), pp. 711-720, July 1997
3. Iryna Skrypnyk, David G. Lowe “Scene Modelling, Recognition and Tracking with Invariant Image Features” IEEE and ACM International Symposium on Mixed and Augmented Reality 2004: 110-119
4. D. Stricker, T. Kettenbach. “Real-time Markerless Vision-based Tracking for Outdoor Augmented Reality Applications” IEEE and ACM International Symposium on Augmented Reality (ISAR 2001), New York, USA, 29-30 October 2001.
5. David G. Lowe “Distinctive image features from scale-invariant keypoints” International Journal of Computer Vision, 60, 2 (2004), pp. 91-110
6. Ke, Y., Sukthankar, R. “PCA-SIFT: A more distinctive representation for local image descriptors” CVPR 2004. Volume 1., Washington, DC, USA, IEEE Computer Society (2004) 511-517
7. Krystian Mikolajczyk, Cordelia Schmid “ A performance evaluation of local descriptors” IEEE Transactions on Pattern Analysis & Machine Intelligence (10) vol. 27, pp.1615-1630, 2005
8. Jeffrey S. Beis , David G. Lowe “Shape Indexing Using Approximate Nearest-Neighbour Search in High-Dimensional Spaces” Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR '97), p.1000, June 17-19, 1997

9. M. Pollefeys, L. Van Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, R. Koch “Visual modeling with a hand-held camera” *International Journal of Computer Vision* 59(3), 207-232, 2004
10. V. Lepetit, P. Lagger and P. Fua, “Randomized Trees for Real-Time Keypoint Recognition” *Conference on Computer Vision and Pattern Recognition*, San Diego, CA, June 2005
11. Richard O. Duda, Peter E. Hart, David G. Stork “Pattern Classification”, Second Edition Wiley Interscience, 2001
12. M. Skurichina, “Stabilizing weak classifiers” Ph.D. thesis, Delft University of Technology, Delft, 2001, October 15
13. M. Grabner, H. Bischof, “Object recognition based on local feature trajectories” 1st Cognitive Vision Workshop, OCG Oesterreichische Computer Gesellschaft, 2005

# Reconstruction of Façade Structures Using a Formal Grammar and RjMCMC

Nora Ripperda and Claus Brenner

Institute of Cartography and Geoinformatics,  
University of Hannover, Germany  
{Nora.Ripperda, Claus.Brenner}@ikg.uni-hannover.de

**Abstract.** Today's processes to extract man-made objects from measurement data are quite traditional. Often, they are still point based, with the exception of a few systems which allow to automatically fit simple primitives to measurement data. At the same time, demands on the data are steadily growing. The need to be able to automatically transform object representations, for example, in order to generalize their geometry, enforces a structurally rich object description. Likewise, the trend towards more and more detailed representations requires to exploit structurally repetitive and symmetric patterns present in man-made objects, in order to make extraction cost-effective. In this paper, we address the extraction of building façades in terms of a structural description. As has been described previously by other authors, we use a formal grammar to derive a structural façade description in the form of a derivation tree. We use a process based on reversible jump Markov Chain Monte Carlo (rjMCMC) to guide the application of derivation steps during the construction of the tree.

## 1 Introduction

The extraction of man-made objects from sensor data has a long history in research [1]. Especially for the modelling of 3D buildings, numerous approaches have been reported, based on monoscopic, stereoscopic, multi-image, and laser scan techniques. While most of the effort has gone into sensor-specific extraction procedures, very little work has been done on the structural description of objects.

Modelling structure though is very important for downstream usability of the data, especially for the automatic derivation of coarser levels of detail (LoD) from detailed models (a process called generalization). Being able to deliver different LoDs tailored to different customers needs, to context-adapted visualizations, such as on mobile displays, or simply to cut down rendering time of large models is essential for 3D models to enter the market. The Sig3D group has defined five levels of detail for building models [3]. However, the definition of discrete LoDs alone does not imply any path to derive one level from the other in an automated way. Experience from 2D map generalization in cartography shows that generalization purely based on geometric information is indeed a hard problem, which becomes even worse in 3D.



Representing structure is not only important for the later usability of the derived data, but also as a means to support the extraction process itself. A fixed set of structural patterns allows to span a certain subspace of all possible object patterns, thus forms the model required to interpret the scene. Especially for man-made structures such as building façades, a large number of regularity conditions hold. In interactive measurement processes, introducing structural descriptions can cut down acquisition time, since repeated or mirrored parts can be introduced in one step.

This paper elaborates on the grammar-based extraction of façade descriptions. The grammar is used to guide the generation of possible façade layouts using a reversible jump Markov Chain Monte Carlo (rjMCMC) process to explore solution space.

## 2 Related Work

Grammars have been extensively used to model structures. For modelling plants, Lindenmayer systems were developed by the biologist Aristid Lindenmayer [7]. They have also been used for modelling streets and buildings [6,4]. But Lindenmayer systems are not necessarily appropriate for modelling buildings. Buildings differ in structure from plants and streets, in that they don't grow in free space and modelling is more a partition of space than a growth-like process.

For this reason, other types of grammars have been proposed for architectural objects. Stiny introduced shape grammars which operate on shapes directly [8]. The rules replace patterns at a point marked by a special symbol. Mitchell describes how grammars are used in architecture [5]. The derivation is usually done manually, which is why the grammars are not readily applicable for automatic modelling tools.

Wonka et al. developed a method for automatic modelling which allows to reconstruct different kinds of buildings using one rule set [9]. The approach is composed of a split grammar, a large set of rules which divide the building into parts, and a control grammar which guides the propagation and distribution of attributes. During construction, a stochastic process selects among all applicable rules.

Dick et al. introduce a method which generates building models from measured data, i.e. several images [2]. This approach is also based on the rjMCMC method. In a stochastic process, 3D models with semantic information are built.

## 3 Grammar-Based Façade Reconstruction

In this section, the basic concept of our method is described. The distinctive feature of our approach is that we combine a grammar-based façade description with a rjMCMC-based exploration of the derivation tree. Thus, compared to existing rjMCMC approaches, we gain the ability to explicitly model superstructures, such as regularity and symmetry, in a hierarchical way. Compared to

existing grammar-based approaches, we use `rjMCMC` and the associated evaluation functions to guide the application of derivation rules and thus achieve a measurement data driven instantiation of the derivation tree.

For our experiments, we use terrestrial laser scan data and images. For the moment, we concentrate on façades, i.e., the measurement data consists of point clouds and orthorectified images of single façades (fig. 1).



**Fig. 1.** Point cloud and orthorectified image of a façade

### 3.1 Model Description Using a Façade Grammar

The façade model is described in terms of a recursive partition of space. We obtain a partition from the application of a derivation rule of the split grammar. The overall façade partition is represented by a derivation tree. Each node corresponds to one of the symbols of the grammar. There are two kinds of symbols, the first one being nonterminals (tab. 1). Geometrically, nonterminals do not represent façade geometry directly but serve as containers which hold other objects, represented in the derivation tree by nonterminal or terminal children. Some of this containers imply that their children have identical properties while others don't. `SYMMETRICFAÇADE` indicates symmetries in the façade and can be derived in `SYMMETRICFAÇADESIDE` which represent the left side and the mirrored right side of the façade and an optional `SYMMETRICFAÇADEMIDDLE`. The second group contains the terminal symbols, which represent façade geometry and cannot be subdivided further (tab. 2). The start symbol is the

**Table 1.** Nonterminal symbols corresponding to containers

<code>ABOVEDOOR</code>	<code>FAÇADEROW</code>	<code>SYMMETRICPARTFAÇADE</code>
<code>ABOVEWINDOW</code>	<code>GABLE</code>	<code>SYMMETRICPARTFAÇADEMIDDLE</code>
<code>FAÇADE</code>	<code>GROUND FLOOR</code>	<code>SYMMETRICPARTFAÇADESIDE</code>
<code>FAÇADEARRAY</code>	<code>IDENTICALFAÇADEARRAY</code>	<code>SYMMETRICFAÇADE</code>
<code>FAÇADECOLUMN</code>	<code>PARTFAÇADE</code>	<code>SYMMETRICFAÇADEMIDDLE</code>
<code>FAÇADEELEMENT</code>	<code>STAIRCASECOLUMN</code>	<code>SYMMETRICFAÇADESIDE</code>

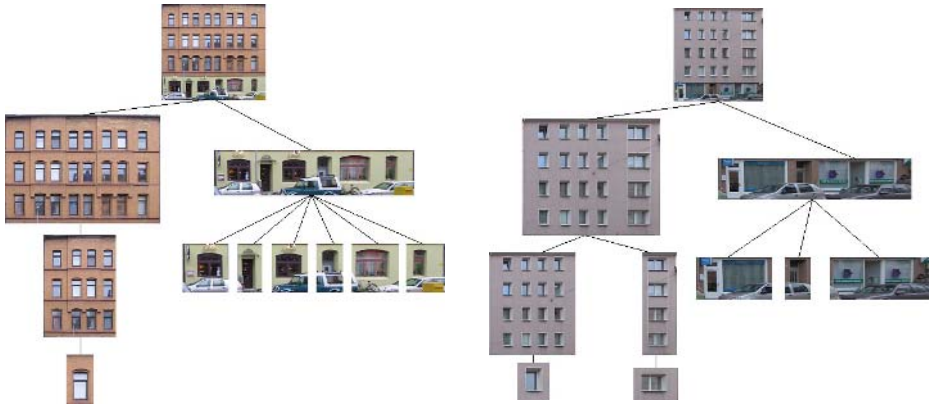
symbol `FAÇADE`. Starting from it, the subdivision can be made by rules similar to the ones introduced by [9]. The model is expressed as a derivation tree with `FAÇADE` as root. Derivation rules have a left side which consists of one

**Table 2.** Terminal symbols corresponding to façade geometry

DOOR	STAIRCASEWINDOW	WINDOW
DOORARCH	WALL	WINDOWARCH

symbol and a right side which may comprise several symbols in a certain spatial layout. As an example, a grammar rule splits FAÇADE into GROUND FLOOR and PARTFAÇADE. Fig. 2 shows two examples of the subdivision of façades. In both cases the façade is subdivided into GROUND FLOOR and the upper floors represented by PARTFAÇADE. The GROUND FLOOR is partitioned in different FAÇADEELEMENTS which contain a DOOR or a WINDOW each. The upper floors are modeled in different ways. In the first case it is a SYMMETRICPARTFAÇADE with a IDENTICALFAÇADEARRAY of WINDOWS inside. The second consists of two different IDENTICALFAÇADEARRAYS with different types of WINDOWS.

The model is described by a parameter vector  $\theta$  which contains the derivation tree and the attributes of the symbols.



**Fig. 2.** Example subdivision of façades

### 3.2 Exploration of the Derivation Tree Using RjMCMC

For the model generation we use a stochastic method. We are searching for the model with parameter vector  $\theta$  with the highest probability  $p(\theta|D_S D_I)$  under given scan ( $D_S$ ) and image data ( $D_I$ ). The parameter vector  $\theta$  encodes the current state of the derivation tree, including attributes.

To obtain the value of  $\theta$ , we use a Markov Chain simulation. This simulates a random walk in the space of  $\theta$ . The process is led by a transition kernel  $P(\theta_t|\theta_{t-1})$  and converges to a stationary distribution  $p(\theta|D_S D_I)$ .

During the simulation façade elements are added, deleted or changed. The first two operations change the number of elements on the façade and thus the dimension of the parameter vector  $\theta$ . The basic Markov Chain Monte Carlo method

doesn't support dimension changes of  $\theta$  and therefore we use the rjMCMC method. This method allows a change in the dimension of the parameter vector  $\theta$  and thereby the number of façade elements can vary during the simulation. The rjMCMC method requires reversibility. For each change from state  $\theta_1$  to state  $\theta_2$  there must exist a reverse change from  $\theta_2$  to  $\theta_1$ .

The rjMCMC method uses a scoring function for the evaluation of changes in the parameter vector to accept or reject a proposed jump, as well as a jumping distribution that proposes these jumps. The work flow of the method is as follows. Beginning with a start value of  $\theta$  (corresponding to the grammar symbol FAÇADE) a jump is determined by the jumping distribution according to the current state (expressed by  $\theta$ ). This jump is executed and the scoring function is used to decide if it is accepted. If the jump is rejected the changes are undone.

### 3.3 Jumping Distribution

A change is proposed depending on the jumping distribution  $J_t(\theta_t|\theta_{t-1})$  that expresses the likelihood for each change. At the moment, the probability is assigned to each change manually depending on an assumed likelihood of the result. For example, a change FAÇADE  $\rightarrow$  IDENTICALFAÇADEARRAY is more likely than FAÇADE  $\rightarrow$  FAÇADEARRAY because façades build regular structures. We expect to improve the distribution for some changes with further analysis of façade structure. Each state change is in one of the following categories:

- Application of a split rule from the grammar. Façade elements are divided horizontally, vertically or in both directions and each part becomes a new symbol (see fig. 3). The split indicates a change in the façade. If the ground floor differs from the rest of the façade, a split is applied.

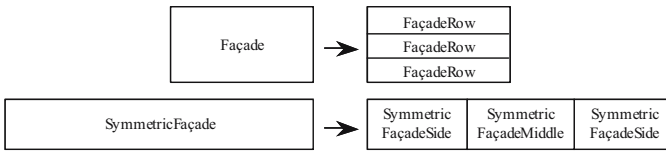
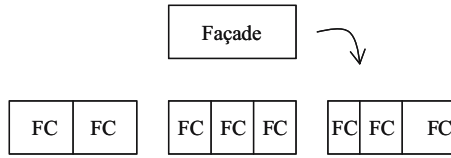


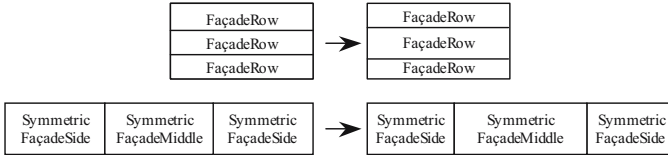
Fig. 3. Split rules

In fact, one grammar rule comprises a set of changes to the parameter vector  $\theta$ , since the associated attributes have to be chosen, such as the number and size of children. Fig. 4 shows an example where one rule splits the symbol FAÇADE into FAÇADECOLUMNS. The number of columns and their width is determined randomly. If a FAÇADE can be divided into several FAÇADECOLUMNS the general rule stands for all rules of this kind with any number and position of columns.

- Changes in structure. Even after derivation of new containers according to the previous step, a second set of state changes allows to modify parameters,



**Fig. 4.** One parameterized split rule represents splits which differ in number and geometry



**Fig. 5.** Changes which modify splits

e.g. the number of columns or the position of the parting lines between columns (see fig. 5). The same can be done starting from a child symbol. The position and extent of a symbol may change. In this case, the neighbor symbols which are involved in the change have to be changed as well.

- Replacement of symbols. This allows to interchange one symbol in the derivation tree by another symbol. In this case, the geometry stays the same, but the denotation changes. This is for example used in the case of the symbols FAÇADE and ABOVEWINDOW. The rules

$$\begin{aligned}
 \text{FAÇADE} &\rightarrow \text{SYMMETRICFAÇADE} \\
 \text{ABOVEWINDOW} &\rightarrow \text{WINDOWARCH} \\
 \text{ABOVEWINDOW} &\rightarrow \text{WALL}
 \end{aligned}$$

allow to replace this symbols by other symbols.

To ensure reversibility, each change can be applied from left to right and vice versa. This is a difference to the way split grammars are used, but is a requirement for the rjMCMC approach.

### 3.4 Scoring Functions

For the evaluation of changes, we use different methods which can be divided into two groups. The first group contains methods which test the general plausibility of the model of the façade. The second group evaluates how good the model fits the data by comparing it to range and image data. In any case, the evaluation functions return a score which is used to decide if the change is accepted or rejected.

The general plausibility depends on the alignment, the extent and the position of the façade elements. Windows are usually arranged in rows and columns. Therefore, such layouts are assigned a high acceptance score. The same holds for symmetric layouts. We consider the size and the aspect ratio of façade elements to rate their probability. We also use the size for the rating of the subdivision into rows, columns and arrays. A row which is five meters high is not very likely and thus has a low acceptance score. The last general criterion is the position of the elements. A door in the third floor is not very likely, so only doors in the ground floor are assigned a high score. For the general plausibility we can use the same scoring functions as given in [2]. The alignment is rated by

$$f_{align}(\theta) = \sum_{r=1}^R [Var(\mathbf{t}_r) + Var(\mathbf{b}_r) + Var(\mathbf{r}_r - \mathbf{l}_r)] + \sum_{c=1}^C [Var(\mathbf{l}_c) + Var(\mathbf{r}_c) + Var(\mathbf{t}_c - \mathbf{b}_c)]$$

where  $\mathbf{t}_r, \mathbf{b}_r, \mathbf{l}_r, \mathbf{r}_r$  are top, bottom, left and right coordinates of the façade element (window or door) belonging to the row  $r$  and  $\mathbf{t}_c, \mathbf{b}_c, \mathbf{l}_c, \mathbf{r}_c$  the same for the column  $c$ . The function  $Var(\mathbf{x})$  denotes the empirical variance of the elements of  $\mathbf{x}$ . The symmetry is evaluated by

$$s_{sym}(\theta) = \sum_{r=1}^R [(\mathbf{l}_r - \mathbf{l}) + (\mathbf{r}_r - \mathbf{r})]^2$$

where  $\mathbf{l}_r$  are the leftmost points of the elements of row  $r$ ,  $\mathbf{r}_r$  are the rightmost points of the elements of row  $r$ , and  $\mathbf{l}$  and  $\mathbf{r}$  are the left and right coordinates of the façade.

These evaluation functions test the configuration of windows, thus they can be applied only after terminal symbols exist in the model. The acceptance is defined by a threshold. Fig. 6 shows two configurations of façade elements. The left and right configurations have an alignment score of 0.013 and 0.1, respectively. The symmetry scores are 0.06 and 0.2.

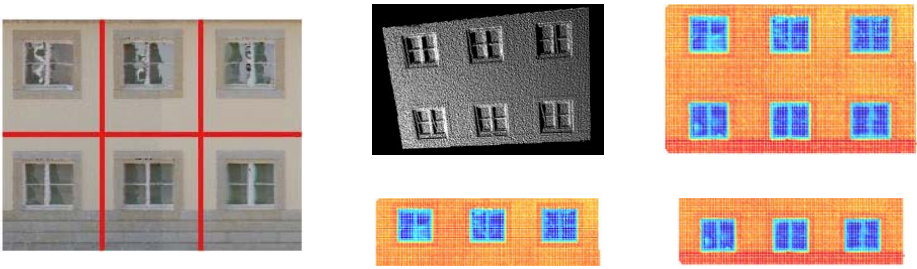


Fig. 6. Façade elements aligned and disturbed

To evaluate the match of the data to the model, scan and image data are used. In the first case, the fact that window points typically lie behind the façade is exploited. In the second case, color difference has been used since windows typically appear darker than the surrounding façade. In both cases, the

information is used for the subdivision into rows, columns, and arrays as well. For example, upon division into rows, the resulting row strips are correlated to obtain an acceptance score. Additionally, in image data a color change may indicate a changeover of ground floor and first floor.

A proposed split of a container can demand that the children have the same properties. In this case we calculate the correlation of each children in the image data and a mesh of the scan data. The correlation value determines whether the split is accepted or not. Fig. 7 shows a proposed partition in a IDENTICALFAÇADEARRAY in the left. The right part shows the mesh of the scan data and the partition in the upper and lower part. In this example image and scan correlation result in high values and the change is accepted.



**Fig. 7.** Proposed partition in an IDENTICALFAÇADEARRAY, point cloud of the façade, mesh of the point cloud and partition in upper and lower part

For a split that differentiates the children another criterion is used. Here we use color changes as obtained from strong image gradients or the region boundaries of planar segmentation. We score the split of a FAÇADEELEMENT into DOOR or WINDOW and surrounding elements with edge detection in image data and the gradient of the mesh of the point cloud. Additionally we use the mean distance of points and the façade plane.

## 4 Results

We tested this approach on façade data. The input data is the 3D point cloud and an orthorectified image calculated using the point cloud. Fig. 8 shows the result of our method applied to the data shown in fig. 1 and the derivation tree is shown in fig. 9. The façade is found to be symmetric so the root of the derivation tree is SYMMETRICFAÇADE. The two children are SYMMETRICFAÇADESIDE and SYMMETRICFAÇADEMIDDLE whereas the first child represents both the left and the mirrored right side of the façade. In this case a two times three subdivision is proposed for each side. And each part contains a WINDOW of the same size and position based on a fixed point. The middle part is subdivided in FAÇADEELEMENTS which contain a DOOR and a WINDOW respectively. The WALL parts are not displayed in image and tree.

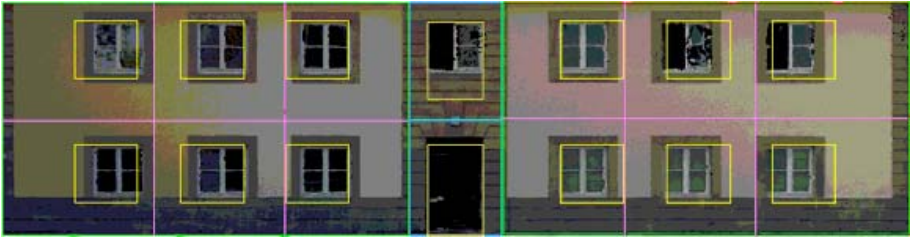


Fig. 8. Resulting façade image

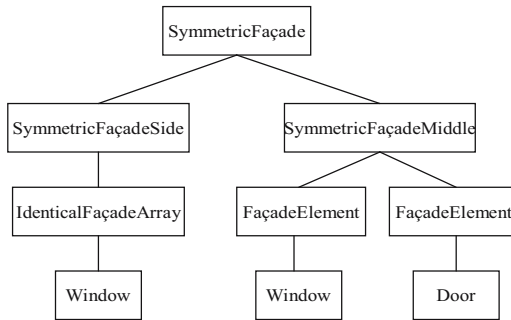


Fig. 9. Derivation tree of the façade shown in fig. 1

## 5 Conclusions and Outlook

In this paper, we have proposed a new approach for the reconstruction of façade structures. It combines two previously reported approaches, namely the generation of artificial façade structures using grammars, and the reconstruction of façades using rjMCMC. Compared to existing grammar-based approaches, we gain the ability to reconstruct façades based on measurement data. Compared to existing rjMCMC approaches, by using a grammar, we obtain a hierarchical façade description and the ability to evaluate superstructures such as regularity and symmetry at an early stage, i.e., before terminal symbols such as WINDOW are instantiated.

We have shown first results in terms of symbols, rules, and score functions. However, much remains to be done. First, we plan to enlarge our set of derivation rules as well as to improve our scoring functions. A systematic capture of façade images is under way, by which we hope to gain more insight into typical façade patterns, which will help us to improve the definition of derivation rules of our grammar. Finally, we are looking into ways how the required jumping distributions can be derived from training data.

**Acknowledgements.** This work was done within in the scope of the junior research group “Automatic methods for the fusion, reduction and consistent



combination of complex, heterogeneous geoinformation”, funded by the VolkswagenStiftung, Germany.

## References

1. E. P. Baltsavias. Object extraction and revision by image analysis using existing geodata and knowledge: current status and steps towards operational systems. *ISPRS Journal of Photogrammetry and Remote Sensing*, 58:129–151, 2004.
2. A.R. Dick, P.H.S. Torr, R. Cipolla, and William Ribarsky. Modelling and interpretation of architecture from several images. *International Journal of Computer Vision*, 60(2):111–134, 2004.
3. T. H. Kolbe, G. Gröger, and L. Plümer. CityGML – interoperable access to 3D city models. In P. van Oosterom, S. Zlatanova, and E. Fendel, editors, *Proc. of the Int. Symposium on Geo-information for Disaster Management, Delft, March 21.-23.* Springer, 2005.
4. Jean-Eudes Marvie, Julien Perret, and Kadi Bouatouch. The FL-system: a functional L-system for procedural geometric modeling. *The Visual Computer*, 21(5):329 – 339, June 2005.
5. William John Mitchell. *The Logic of Architecture : Design, Computation, and Cognition*. Cambridge, Mass.: The MIT Press, 1990.
6. Y.I.H. Parish and P. Mller. Procedural modeling of cities. In E. Fiume, editor, *ACM SIGGRAPH*. ACM Press, 2001.
7. Przemyslaw Prusinkiewicz and Aristid Lindenmayer. *The algorithmic beauty of plants*. New York, NY: Springer, 1990.
8. G. Stiny and J. Gips. *Shape Grammars and the Generative Specification of Painting and Sculpture*, pages 125–135. Auerbach, Philadelphia, 1972.
9. Peter Wonka, Michael Wimmer, Franois Sillion, and William Ribarsky. Instant architecture. *ACM Transaction on Graphics*, 22(3):669–677, July 2003.

# Stable Wave Detector of Blobs in Images

Jan Dupač<sup>1</sup> and Václav Hlaváč<sup>2</sup>

<sup>1</sup> RS Dynamics s.r.o., 149 00 Prague 4, Starochodovská 1359, Czech Republic  
dupac@rsdynamics.com

<http://www.rsdynamics.com>

<sup>2</sup> Czech Technical University, Faculty of Electrical Engineering  
Department for Cybernetics, Center for Machine Perception  
121 35 Prague 2, Karlovo náměstí 13, Czech Republic

hlavac@fel.cvut.cz

<http://cmp.felk.cvut.cz>

**Abstract.** Stable Wave Detector (SWD) is a new multiscale landmark detector in the intensity image. SWD belongs to a group of interest-point-like operators aiming at detecting repeatedly distinguished entities regardless of their semantics. The speed and the robustness of landmark detection and the precision of landmark localization are main issues. The target landmarks are blobs which correspond to local maxima/minima of intensity (positive and negative peaks). The detector is based on the phase of the first harmonic wave in the moving window. The localization is a result of an integral transformation rather than a derivative. Thus, the blob detector is inherently robust to noise. The SWD provides subpixel localization of blobs together with the estimate of its precision, the measure of the strength/significance and the estimate of the size/scale for each blob.

## 1 Motivation

This work is a part of a larger project leading to a hand-held instrument for on-line 3D positioning of geophysical measurements. The proposed method uses video sequences captured by a calibrated stereo rig. The required precision is in order of 0.1 meter in the area of several hundreds meters. The requirement for online processing and a hand-held battery-powered system induces hard constraints to computational complexity of used algorithms which should simultaneously meet challenging precision requirements. The subpixel measurement is required because enlarging camera resolution leads to quadratically growing amount of data compared to the linear gain in precision.

What are the landmarks to search? We were inspired by the nature. It is known that dogs are short-sighted and cannot see sharply. Nevertheless, they live in the 3D world and can hunt. The first author is short-sighted too, however, even without glasses he can also feel a 3D structure of the world. Another observation concerns fast movements. If something moves very fast then we are unable to recognize details of it. However, we feel its position and speed. These observations are our arguments to seek same integral units as blobs rather than differential structures as corners or edges, i.e., the whole window instead of its corners and its center instead of the border.

## 2 State-of-the-Art

The proposed method can be broadly classified among interest-point-like operators aiming at detecting repeatedly distinguished entities without considering their semantics. These entities are mainly used for matching in different images, stereo, image retrieval, categorization, object recognition, etc. There is a popular group of corner detectors originated in Moravec detector [11] improved by Harris [4]. Fast implementation of Harris detector is provided in [12]. Comparison of interested point detectors is given in [15]. An overview of existing interest point detectors can be found in [9].

Corners are not inherently scale invariant, i.e., a multi-scale Harris detector does not localize the same local structure at the same point in different scale. Mikolajczyk [9] choose the ‘correct’ scale as maxima of Laplacian-of-Gaussian in the scale space. Deriche [1] approached this problem by fitting the line through the locations at different scales and searched where the series of points converge. Lowe’s detector searches maxima of Difference of Gaussian [7] in the scale space.

It was noticed by many that richer structures in images compared to local, ‘derivative-based’ corners can bring additional benefit. Proposed methods are detectors based on affine normalization around Harris points [9,14], a detector of ‘maximally stable extremal regions’ [8], an edge-based region detector [16], a detector based on intensity extrema [17], and the detectors of ‘salient regions’ based on entropy [5] or wavelet transform [6]. The performance of above region detectors is compared in [3,10] where the performance to change in viewpoint, scale, illumination, defocus and image compression are considered.

The proposed blob detector was motivated by practical observations and a signal processing theory. Its robustness and speed is derived from basic properties of Fourier transformation, dot product and phase of the first harmonics. A global difference of phase of the first harmonics between two omnidirectional images was used to find their relative orientation [13].

## 3 Stable Wave Detector

The target patterns in the images are blobs, i.e., local minima or maxima of intensity. In 1D, the blob corresponds to a positive or a negative peak. Let us start our explanation with an experiment in 1D which motivated our approach and gave it the name. The examples of ideal peaks will be shown in Section 3.2. The single scale SWD algorithm will be introduced in Section 3.3. Main components of SWD are described in Sections 3.4 and 3.5. The extension of SWD to multiscale is given in Section 3.6. Section 3.7 describes how to apply SWD to detect blobs in 2D.

### 3.1 What Is Stable Wave?

The word ‘wave’ comes from Fourier transform which is our basic tool. The term ‘stable’ says that we seek the wave in the signal (or image) which is stable / well

localized in some sense. The idea of the stable wave originates from a simple experiment in 1D. Let us consider data of length  $N > 2T$ ,  $T \geq 4$  containing a peak of width about  $T/2$ . Then suppose that the peak is located somewhere in the closed interval  $[x_0 - T/2, x_0 + T/2]$ , where  $T < x_0 < (N - T)$ . The following iterative algorithm can localize the peak:

1. Define frame  $F_1$ , as  $[x_0 - T/2, x_0 + T/2 - 1]$ .
2. For frame  $F_i$ , compute Fourier coefficients  $a, b$  and phase  $\varphi_i \in (-\pi, \pi]$  of the first harmonic wave (i.e., with a period  $T$ ).
3. Estimate the peak location relatively to frame  $F_i$  as

$$\begin{aligned} x_i &= \frac{T}{4}(1 - 2\frac{\varphi_i}{\pi}); \varphi_i < 0, \text{ (maxima)} \\ x_i &= \frac{T}{4}(3 - 2\frac{\varphi_i}{\pi}); \varphi_i > 0, \text{ (minima)} \end{aligned} \tag{1}$$

4. Define a new frame  $F_{i+1}$  centered around  $x_i$ .

Repeat steps 2-4 until the process converges, i.e., until  $F_{i+1}$  is the same as  $F_i$  (the peak is stable) or diverges  $|x_i - x_0| > T/2$  or oscillates  $i > T$ .

The process typically converges in the first or the second iteration for the ideal peak without noise (Section 3.2). The algorithm works well also for edges for which  $x_i$  is a zero crossing instead of an extreme.

An interesting observation is that the precise knowledge of the peak width is not crucial. The algorithm works well up to about an octave below and over the optimal  $T$ . The amplitude

$$A = \sqrt{(a^2 + b^2)} \tag{2}$$

can measure the strength of response (suitability) of a period  $T$  for a given peak.

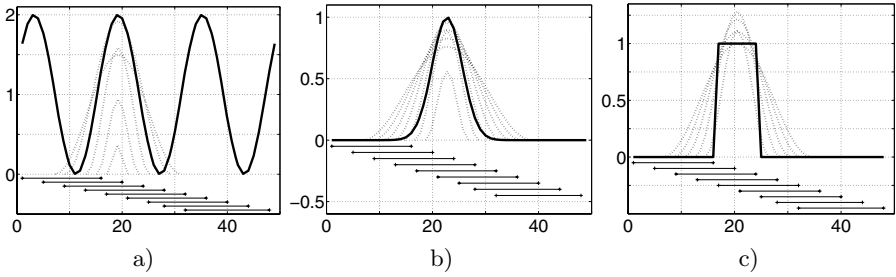
### 3.2 Ideal Peaks

Examples of ideal shapes from the SWD point of view are cosine, Gaussian, rectangle and other similar symmetric shapes. The detector localizes peaks of ‘ideal shape’ with the subsample precision similarly as humans would localize them intuitively when looking at them. For example, the detector finds the maximum of Gaussian or the center of a rectangle.

Figure 1 shows the examples of ideal peaks (solid curves) suitable for SWD with frames of width 16 (line segments bellow the curves). The dotted curves visualize the results of SWD algorithm at variable frame lengths. The dotted curves are cosine waves with the amplitude found by SWD centered at the peak location found by a single scale SWD. The cosine curves are shifted up to have positive values for better visualization. SWD algorithm successfully found peaks even if the frame length was far from the period of the signal or from the double of the peak width.

### 3.3 Outline of the Single-Scale SWD Algorithm

The algorithm described in Section 3.1 just motivated SWD. A practical SWD algorithm is sketched in this section for the single-scale. Let us consider data to be a row vector  $D$  of length  $N$  and the expected width of the peaks be near  $T/2$ .



**Fig. 1.** Examples of ideal peaks (*solid curves*), frames (*line segments*) of length  $T = 16$ , and stable waves found by SWD at different periods ( $T \in \{4, 8, 12, 16, 20, 24, 28, 32\}$ , *dotted curves*). a) Sine wave ( $T = 16, \varphi = 0.69$ ). b) Gaussian  $y = \exp(-(x-22.76)^2/16)$ . c) 8-sample-wide rectangle.

The SWD algorithm consists of four forward steps.

1. The input data are divided into  $n$  overlapping frames  $F_i$  of length  $T$ . The frames should overlap more than  $T/2$ . Efficient choice of the overlap is described in Section 3.4.
2. For each frame  $F_i$ , Fourier coefficients  $a_i, b_i$  are computed. (Section 3.4)
3. Candidate frames are found as cosine frames meeting Consistent Neighboring Frame (CNFr) criterion (Section 3.5).
4. Having a candidate frame  $F_i$ , a subsample location  $x_i$  of the peak inside the frame can be found using Equation (1).

### 3.4 The Efficient Computation of Fourier Coefficients

In the discrete case, Fourier coefficients are computed as dot products  $a_i = F_i \cdot S$ ,  $b_i = F_i \cdot C$ , where  $F_i$  are intensity data in the frame  $i$  and

$$S_t = \sin(2\pi(t/T)), \quad C_t = \cos(2\pi(t/T)), \quad t = 0, \dots, T - 1.$$

To compute all coefficient  $a_i, b_i$  for whole data of the length  $N$ , the frame length  $T$ , and an arbitrary overlap  $\Omega$ , the number of multiplication would be

$$\mathcal{O}(N, \Omega) = 2N \frac{T}{T - \Omega}.$$

For example  $\mathcal{O}(N, 2/3T) = 6N$ ,  $\mathcal{O}(N, 3/4T) = 8N$ ,  $\mathcal{O}(N, 4/5T) = 10N$ .

The choice of the frame overlap is an important decision which influences both speed and precision. Looking at the properties of sine and cosine functions, we found  $3/4T$  to be the best choice. The reason is the well known fact that the sine wave (the first base function) is the cosine wave (the second base function) shifted by a quarter of a period, i.e.,  $\cos \varphi = \sin(\varphi + \pi/2)$ . Exploring this fact, significant amount of computation can be saved for frame shift equal to  $T/4 \sim \pi/2$ .

Dot product  $a_i^1 = F_i \cdot S$  can be written as

$$a_i = \sum_{j=1}^T F_i(j)S(j) = D_i^s \left(1, \frac{T}{4}\right) + D_i^s \left(\frac{T}{4}, \frac{T}{2}\right) + D_i^s \left(\frac{T}{2}, 3\frac{T}{4}\right) + D_i^s \left(3\frac{T}{4}, T\right),$$

$$b_i = \sum_{j=1}^T F_i(j)C(j) = D_i^c \left(1, \frac{T}{4}\right) + D_i^c \left(\frac{T}{4}, \frac{T}{2}\right) + D_i^c \left(\frac{T}{2}, 3\frac{T}{4}\right) + D_i^c \left(3\frac{T}{4}, T\right),$$

where  $D_i^c$  are dot products of a part of the frame with a part of the cosine wave and similarly  $D_i^s$  are dot products of a part of a frame with a part of a sine wave. The two neighboring frames have  $3/4T$  overlap, therefore, six of the eight partial dot products differ only in the signs and it is not necessary to compute them twice. As a result, only  $2N$  (instead of  $8N$ ) multiplications are needed for data containing  $N$  samples.

### 3.5 Consistent Neighboring Frame (CNFr) Criterion

The principle of the criterion can be illustrated on the example of Gaussian depicted in Figure 1b. The frames of length  $T = 16$  and overlap  $3/4 T$  are shown as line segments below the curves. Table 1 shows the first two Fourier coefficients, phases and amplitudes for the frames of length 16. The frame 4 containing the maximum of the peak is a negative cosine frame ( $-b_i > |a_i|$ ) surrounded by the two sine frames ( $|b_i| < |a_i|$ ). Frames containing rising edge have negative sine coefficients and frames containing falling edge have positive sine coefficients as it could be expected considering the similarity to sine wave. Each cosine frame does not contain a peak. For example falling edge of Gaussian produces cosine frames (frames 7, 8 in Table 1) which do not contain any peak. CNFr criterion can be summarized as:

- A cosine frame  $F_i$  contains a stable minimum if  $b_i > k|a_i|$  &  $b_{i+1} > k|a_{i+1}|$  &  $a_i > 0$  &  $a_{i+1} < 0$  or  $b_i > k|a_i|$  &  $b_{i+1} < k|a_{i+1}|$  &  $b_{i-1} < k|a_{i-1}|$  &  $a_{i-1} > 0$  &  $a_{i+1} < 0$ .
- A cosine frame  $F_i$  contains a stable maximum if  $-b_i > k|a_i|$  &  $-b_{i+1} > k|a_{i+1}|$  &  $a_i < 0$  &  $a_{i+1} > 0$  or  $-b_i > k|a_i|$  &  $-b_{i+1} < k|a_{i+1}|$  &  $-b_{i-1} < k|a_{i-1}|$  &  $a_{i-1} < 0$  &  $a_{i+1} > 0$ .

The number  $k < 1$  (near 1) moderates the criterion on the frame to be considered as cosine. It reduces the chance to loose the candidate due to noise and other perturbations when the both coefficients have similar absolute values. The exact choice of  $k$  is not crucial; we used the value  $7/8$ .

### 3.6 Multi-scale SWD Algorithm

A good peak of a certain width has responses for several different SWD periods as can be seen in Figure 1. In the case of a sine wave, the localization precision was better than 0.04 of the sample for  $T \in \{8, 12, 16, 20, 24\}$ . The strongest

**Table 1.** Gaussian and frames of SWD,  $a$  is a sine coefficient,  $b$  is a cosine coefficient,  $\varphi$  is the phase relative to the frame

Frame	1	2	3	4	5	6	7	8	9
$a_i$	-0.01	-0.12	-0.46	-0.31	0.38	0.42	0.09	0.00	0.00
$b_i$	0.01	0.11	0.12	-0.38	-0.31	0.25	0.18	0.02	0.00
$\varphi_i$	2.17	2.38	2.88	-2.25	-0.70	0.54	1.11	1.34	1.45
Amplitude	0.01	0.16	0.47	0.49	0.49	0.49	0.21	0.02	0.00

response (the amplitude of the stable wave) was at  $T = 16$  (i.e., natural period of the signal). Similar results are for Gaussian (the precision was better than 0.03 for  $T \in \{8, 12, 16, 20, 24, 28, 32\}$ ) and for a rectangle (the peak was detected at  $T \in \{4, 8, 12, 16, 20, 24, 28, 32\}$ , and even zero localization error occurred at  $T \in \{12, 16, 20, 24, 28\}$ ).

Even though SWD algorithm does not require the precise knowledge of the peak width, its estimate must be provided. Such estimate may not be available in many practical situations or the peak width may vary in a wide range. The developed multiscale algorithm searches a hierarchy of peaks at several width levels similarly to the others multiscale detectors [9,7].

Considering excellent multiscale property of SWD, we can afford  $2^n$  step in scale which is more sparse than  $1.4^n$  used by Mikolajczyk [9], or Lowe[7]. The integer scale allows a more efficient computation compared to a non-integer.

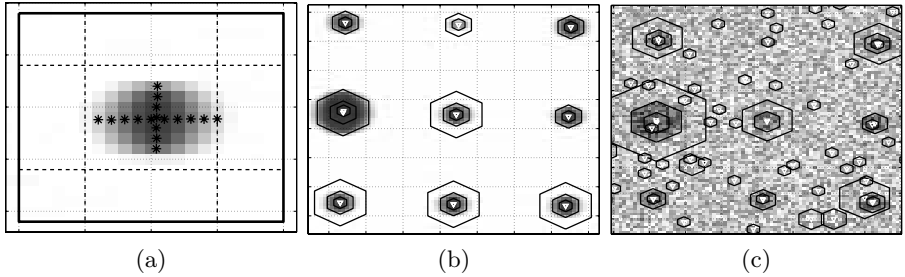
Similarly to [9,7] we can choose the best scale (or filter out weak peaks) according to the strength of the response measured as the amplitude  $A$ , see Equation (2).

The quality of the peak found can be measured as a variance of the locations found at different nearby scales. Good peaks are stable for more than an octave difference of the frame size. We evaluate the stability at approximately a half of an octave lower ( $T_L$ ) and higher ( $T_H$ ) than the original scale ( $T$ ) where the peak was detected. The word ‘approximately’ means that we always choose an integer period. For example, for the scale with  $T = 16$  we choose  $T_L = 12$  and  $T_H = 24$ . We estimate the multiscale stability as  $\min(|x_i - x_L|, |x_i - x_H|)$ , where  $x_i$  is the initial estimate at the period  $T$ ,  $x_L$  is estimated using the frame of length  $T_L$  centered around  $x_i$ , and  $x_H$  is estimated using the frame of length  $T_H$  centered around  $x_i$ .

### 3.7 Stable Wave Detector in 2D

Let us observe the example of an ideal blob in 2D in Figure 2a similarly as we observed ideal peaks in 1D. Looking at the intensity along the line (e.g., row or column of the image) passing a blob, there is a peak. The peak in intensity along the line in the image is a necessary condition for the blob presence. In addition, the peak provides two independent constraints – one for rows, second for columns. As we can see in Figure 2, the results of SWD-1D on the rows passing the blob lie on the line (with subpixel precision). Similarly, the results

of SWD-1D on a column passing the blob lie on another line. The two lines are almost perpendicular. The 2D location of the blob can be estimated as the intersection of these two lines.



**Fig. 2.** The image of blobs in a calibration pattern. a) *Black stars* in the middle of the dark patch show the results of SWD-1D along the rows and the columns of the image. b) Results of SWD-2D on the original image. c) Results of SWD-2D on the image with artificially added noise ( $S/N = 1$ ). The blob locations found by SWD-2D are depicted as *white triangles* and the diameter of *black hexagons* indicates the scale.

More thorough mathematical explanation of this approach can be derived from 2D Fourier Transformation (FT) and phase-based methods of stereo matching. The 2D FT is a vector function giving a phase and an amplitude to a vector  $[f_h, f_v]$  (horizontal and vertical frequency). Two independent frequency vectors are needed to obtain a 2D phase information. For better imagination, vector  $F_1 = [f, 0]$  corresponds to the wave parallel with axis  $x$ .  $F_2 = [0, f]$  is the wave parallel to axis  $y$ .  $F_3 = [f, f]$  is the wave parallel to the line  $y = 1 - x$ .

Let us look in detail on the integral of Fourier coefficient  $a$  for  $F_1$ . The base function  $S(x, y)$  depends on  $x$  only. The integral can be decomposed as

$$a = \int_{x,y} S(x, y) I(x, y) dx dy = \int_x s(x) i(x) dx .$$

$$i(x) = \int_y I(x, y) dy .$$

In the discrete case, it means the following. First, sum the intensity  $I(\text{row}, \text{column})$  over the row to get a function  $i(\text{column})$  and then transform function  $i(\text{column})$  by 1D FT to get the phase in the horizontal direction.

The phases corresponding to frequency vectors  $F_1, F_2$  can be used to localize the peak similarly as SWD-1D does. However this solution is not good. The application of CNFr criterion would be difficult. Intuitively, the other problem is that the integrals for Fourier coefficients contain a large neighborhood of the blob. The solid square in Figure 2a shows the optimal square window to detect the blob. The square areas in its corners bring just noise to the integrals.

Our SWD-2D algorithm combines the observation from Figure 2 with the mathematical derivation. In short, we detect the peaks in rows and fit the vertical

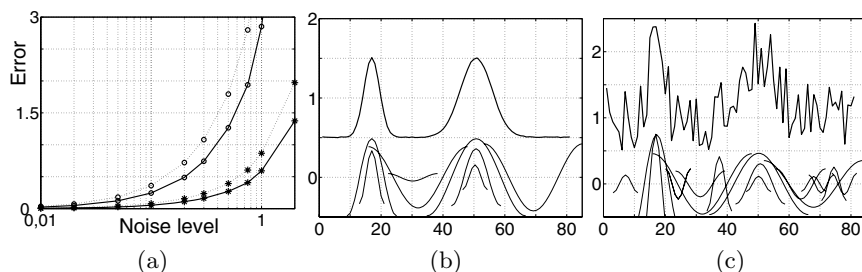


line  $v$  through them. Next, we detect the peaks in columns and fit the horizontal line  $h$  through them. The location of the peak is estimated as the intersection of lines  $h$  and  $v$ . The detailed algorithm is described in [2] due to lack of space.

## 4 Implementation and Experiments

The SWD-2D algorithm has been implemented in MATLAB and tested on synthetic and real data. The experiments are described in detail in [2].

Robustness to noise in 1D case was extensively tested. The results are summarized in Figure 3.



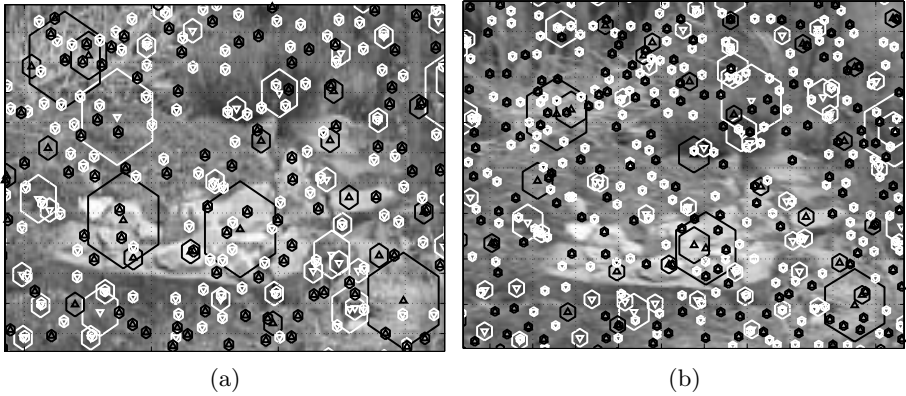
**Fig. 3.** Robustness of SWD-1D to uniform noise. a) Maximum (*circles*) and average (*stars*) localization error as the function of noise level added to original signal containing two Gaussian peaks (narrow peak – *solid line*, wide peak – *dotted line*). The results come from 100 repeats. b) Example of signal with noise level 0.01. c) Example of signal with noise level 1.

The robustness to noise of WSD-2D was tested on the real image of the camera calibration pattern consisting of black circular dots. The detail of the image is depicted in Figure 2b. The original real image was degraded by the addition of the increasing amount of uniform noise (Figure 2c). The localization proved to be stable to noise level about 1. The achieved precision is below 0.05 of a pixel for original image and below 0.15 of a pixel up to noise level 0.3.

Let us show detected blobs on a real image of a rock garden. The purpose of the experiment is to display visually where blobs are detected. The second aim is to demonstrate the potential of the proposed method for stereo matching. The rock garden is captured from two different view points. The reader can see that many blobs in the first image have a corresponding partner in the second image.

## 5 Conclusions and Future Work

We have proposed a new blob detector which seems to have several favorable properties for practical applications. It is precise and fast. We believe that it is a right way to our application target – implementing it in the batter-powered hand-held instrument for on-line 3D positioning of geophysical measurements.



**Fig. 4.** Illustration of SWD on the rock garden image. The blob locations found by SWD-2D are depicted as *triangles* and the diameter of *hexagons* indicates the scale. *White* color is used for minima, *black* for maxima. (a) The detail of the first image; (b) The detail of the second image, taken from different place.

The semantic-less interest point-like detectors have had an enormous attention in the computer vision community in last few years. Our thorough comparison to them needs our further attention. We need to move from the MATLAB implementation to C language to be able to perform computation time tests and comparisons with other algorithms and implementations. We also would like to extend our method to cope with color images.

**Acknowledgments.** The authors were supported by The Czech Ministry of Education under Project 1M0567.

## References

1. R. Deriche and G. Giraudon. A computational approach for corner and vertex detection. *IJCV*, 10(2):101–124, 1993.
2. J. Dupač and V. Hlaváč. Stable wave detector for precise and fast detection of blobs in the image. Research Report CTU–CMP–2006–03, Center for Machine Perception, K13133 FEE Czech Technical University, Prague, Czech Republic, April 2006.
3. F. Fraundorfer and H. Bischof. Evaluation of local detectors on non-planar scenes. In *Proc. of 28th Workshop of the Austrian Association for Pattern Recognition (AGM/AAPR)*, pages 125–132, Osterreichische Computer Gesellschaft 3-85403-179-3, Hagenberg, 2004.
4. C. Harris and M. Stephens. A combined corner and edge detector. In *Proc. of 4th Alvey Vision Conference*, pages 147–151, March 1988.
5. T. Kadir, A. Zisserman, and M. Brady. An affine invariant salient region detector. In *Proceedings of the 8th European Conference on Computer Vision*, volume 1 of *LNCS 3021*, pages 228–241, Prague, May 2004. Springer-Verlag.

6. E. Loupas and N. Sebe. Wavelet-based salient points: Applications to image retrieval using color and texture features. In *VISUAL '00: Proceedings of the 4th International Conference on Advances in Visual Information Systems*, pages 223–232, London, UK, 2000. Springer-Verlag.
7. D. G. Lowe. Object recognition from local scale-invariant features. In *Proc. of IEEE International Conference on Computer Vision (ICCV1999)*, pages 1150–1157. IEEE Computer Society, 1999.
8. J. Matas, O. Chum, Urban M., and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In Paul L. Rosin and David Marshall, editors, *Proc. of the British Machine Vision Conference*, volume 1, pages 384–393, London, UK, September 2002. BMVA.
9. K. Mikolajczyk and C. Schmid. Scale & affine invariant point detector. *International Journal of Computer Vision*, 60(1):63–86, 2004.
10. K Mikolajczyk, T Tuytelaars, C Schmid, A Zisserman, J Matas, F Schaffalitzky, T Kadir, and L van Gool. A comparison of affine region detectors. *International Journal of Computer Vision*, 65(7):43 – 72, November 2005.
11. H. P. Moravec. Towards automatic visual obstacle avoidance. In *Proc. of The 5th International Joint Conference on Artificial Intelligence, MIT, Cambridge, Massachusetts*, page 584. IJCAI, August 1977.
12. D. Nistér, O. Naroditsky, and J. R. Bergen. Visual odometry. In *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'04)*, pages 652–659. IEEE Computer Society, 2004.
13. T. Pajdla and V. Hlaváč. Zero phase representation of panoramic images for image based localization. In Franc Solina and Aleš Leonardis, editors, *Proc. of 8-th International Conference on Computer Analysis of Images and Patterns*, number 1689 in Lecture Notes in Computer Science, pages 550–557, Tržaška 25, Ljubljana, Slovenia, September 1999. Springer Verlag.
14. F. Schaffalitzky and A. Zisserman. Multi-view matching for unordered image sets, or ‘how do i organize my holiday snaps?’. In *Proceedings of the 7th European Conference on Computer Vision*, volume 1 of *LNCS 2350*, pages 414–431. Springer-Verlag, 2002.
15. C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. *International Journal of Computer Vision*, 37(2):151–172, 2000.
16. T. Tuytelaars and L. Van Gool. Content-based image retrieval based on local affinity invariant regions. In *International Conference on Visual Information Systems*, pages 493–500, 1999.
17. T. Tuytelaars and L. Van Gool. Matching widely separated views based on affine invariant regions. *International Journal on Computer Vision*, 59(1):61–85, 2004.

# Author Index

- Adam, Dan 1  
Admasu, Fitsum 384  
Aerts, Maarten 698  
Aihara, Kazuyuki 414  
Akgul, Yusuf Sinan 677  
Al-Zubi, Stephan 556  
Arnaud, Elise 253  
Axelsson, Maria 61
- Bay, Herbert 708  
Beder, Christian 657  
Bergtholdt, Martin 273  
Birbaumer, Niels 404  
Bischof, Horst 71, 122  
Blankertz, Benjamin 354  
Bleyer, Michael 465  
Borgefors, Gunilla 61  
Borsdorf, Anja 21  
Braun, Mikio L. 344  
Brenner, Claus 750  
Breun, Peter 394  
Brox, Thomas 475, 546, 688  
Buhmann, Joachim M. 344, 424, 708  
Bur, Alexandre 314  
Burkhardt, Hans 132, 182, 263,  
284, 294  
Buss, Martin 394
- Charron, Cyril 334  
Cireşan, Dan 172  
Clausen, Michael 142  
Cornelis, Kurt 192, 698  
Cornelis, Nico 192, 698  
Cremers, Daniel 142, 455, 475,  
546, 688
- Damian, Dana 172  
d'Angelo, Pablo 607  
Dang, Thao 627  
Delponte, Elisabetta 253  
Derichs, Christian 637  
Deselaers, Thomas 202  
Deutsch, Benjamin 536  
Dick, Thorsten 566  
Didas, Stephan 101
- Dornhege, Guido 354, 414  
Dornheim, Jana 162  
Dornheim, Lars 162  
Dovzhenko, Alexander 182  
Droske, Marc 525  
Dupač, Jan 760
- Elger, Christian E. 404  
Erez, Yael 1
- Fehr, Janis 263  
Fischer, Bernd 424, 708  
Franke, Uwe 475  
Fritsch, Jannik 212  
Fritz, Mario 232  
Fussenegger, Michael 122
- Garbe, Christoph 525  
Gebken, Christian 587  
Gelautz, Margrit 465  
Goldlücke, Bastian 729  
González, Jordi 485, 505  
Grau, Oliver 667  
Grest, Daniel 576  
Groß, Horst-Michael 607  
Grosse-Wentrup, Moritz 394
- Hähnel, Michael 324  
Härtel, Volker 739  
Hegerath, Andre 202  
Hellwich, Olaf 111  
Hertel, Ilka 162  
Herzog, Dennis 576  
Hill, N. Jeremy 404  
Hinterberger, Thilo 404  
Hlaváč, Václav 760  
Hoffmann, Christian 627  
Hornegger, Joachim 21  
Hügli, Heinz 314
- Jähne, Bernd 434  
Jehle, Markus 434
- Kappes, Jörg H. 273  
Kersting, Uwe G. 495, 546

- Kertzsch, Ulrich 434  
 Keuchel, Jens 41  
 Keyzers, Daniel 202  
 Klappstein, Jens 475  
 Koch, Reinhard 576, 739  
 Kolev, Kalin 688  
 Köser, Kevin 739  
 Köthe, Ullrich 81  
 Kraiss, Karl-Friedrich 324, 566  
 Krajsek, Kai 91  
 Krauledat, Matthias 354  
 Krüger, Lars 607  
 Kuhl, Annika 607  
 Küttel, Daniel 41
- Labbani-Igbida, Ouiddad 334  
 Lal, Thomas Navin 404  
 Lange, Tilman 344  
 Laskov, Pavel 374  
 Leibe, Bastian 192  
 Li, Zhe 212  
 Linz, Christian 729
- Magnor, Marcus 729  
 Mathes, Tom 515  
 Meine, Hans 81  
 Mekuz, Nathan 364  
 Mester, Rudolf 91  
 Molkenstruck, Sven 718  
 Mouaddib, El Mustapha 334  
 Mozerov, Mikhail 485, 617  
 Müller, Klaus-Robert 354, 374, 414
- Ney, Hermann 202  
 Niemann, Heinrich 536, 637  
 Nolte, Guido 414
- Odone, Francesca 253  
 Ommer, Björn 11  
 Ouerhani, Nabil 314  
 Özden, Egemen 698
- Palme, Klaus 182  
 Pasternak, Taras 182  
 Perwass, Christian 647  
 Peter, Hansruedi 424  
 Piater, Justus H. 515  
 Pinz, Axel 122
- Pock, Thomas 71  
 Powell, Katie 495  
 Preim, Bernhard 162  
 Preusser, Tobias 525
- Rahtu, Esa 284  
 Raupach, Rainer 21  
 Reid, Ian 505  
 Reisert, Marco 132  
 Rhemann, Christoph 465  
 Rieck, Konrad 374  
 Rink, Karsten 152  
 Ripperda, Nora 750  
 Rius, Ignasi 485  
 Roca, Xavier 485  
 Ronneberger, Olaf 182  
 Rosenhahn, Bodo 495, 546  
 Roth, Peter M. 122  
 Roth, Volker 11  
 Rowe, Daniel 505  
 Ruhnau, Paul 444  
 Rumpf, Martin 525  
 Ruzsala, Simon 304
- Sagerer, Gerhard 212, 597  
 Schaefer, Gerald 304  
 Schar, Hanno 51  
 Schechner, Yoav Y. 1  
 Schiele, Bernt 232, 242  
 Schlesinger, Dmitrij 31  
 Schmidt, Frank R. 142  
 Schmidt, Thorsten 182  
 Schnörr, Christoph 273, 444  
 Schoenemann, Thomas 455  
 Schölkopf, Bernhard 404  
 Schröder, Michael 404  
 Schulz, Janina 182  
 Seemann, Edgar 242  
 Seidel, Hans-Peter 495  
 Setia, Lokesh 284, 294  
 Sommer, Gerald 222, 556,  
 587, 647  
 Stahl, Annette 444  
 Steffen, Richard 657  
 Steldinger, Peer 81  
 Stöbel, Dirk 597  
 Strauss, Gero 162  
 Sugiyama, Masashi 354  
 Svensson, Stina 61

- Telea, Alexandru 525  
Teynor, Alexandra 284  
Tolvanen, Antti 587  
Tomioka, Ryota 414  
Tönnies, Klaus 152, 384  
Tsotsos, John K. 364
- Utschick, Wolfgang 394
- Van Gool, Luc 192, 698  
Verri, Alessandro 253  
Villanueva, Juan Jose 505  
Vural, Ulas 677
- Wachsmuth, Sven 212  
Wahl, Friedrich M. 718  
Wedel, Andreas 475  
Weickert, Joachim 101  
Wenhardt, Stefan 536  
Wey, Peter 708  
Widman, Guido 404  
Winkelbach, Simon 718  
Wiratanaya, Andreas 324  
Wöhler, Christian 607
- Zang, Di 222  
Zheng, Hongwei 111  
Zieren, Jörg 566