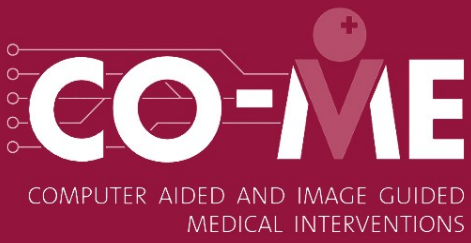
Matthias Harders
Gábor Székely (Eds.)

LNCS 4072

Biomedical Simulation

Third International Symposium, ISBMS 2006
Zurich, Switzerland, July 2006
Proceedings



Springer

Lecture Notes in Computer Science 4072

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

Matthias Harders Gábor Székely (Eds.)

Biomedical Simulation

Third International Symposium, ISBMS 2006
Zurich, Switzerland, July 10-11, 2006
Proceedings

Springer

Volume Editors

Matthias Harders
Gábor Székely
ETH Zurich, Computer Vision Laboratory
Sternwartstr. 7, 8092 Zurich, Switzerland
E-mail: {mharderszekely}s@vision.ee.ethz.ch

Library of Congress Control Number: Applied for

CR Subject Classification (1998): I.6, I.4, J.3, I.3

LNCS Sublibrary: SL 1 – Theoretical Computer Science and General Issues

ISSN 0302-9743
ISBN-10 3-540-36009-3 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-36009-4 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11790273 06/3142 5 4 3 2 1 0

Preface

This book contains the written contributions to the Third International Symposium on Biomedical Simulation (ISBMS), which was held in Zurich, Switzerland, on July 10-11, 2006.

The manuscripts are organized around three thematic sections which cover several of the major aspects of our rapidly growing field: anatomical modeling and tissue properties, simulation of biophysical processes, as well as systems and applications.

The symposium provided an international forum for researchers interested in using biomedical simulation technology for the improvement of patient care. It was held in the spirit and continuation of the symposia on Surgical Simulation and Soft Tissue Modeling (IS4TM) organized in 2003 by INRIA, and on Medical Simulation (ISMS) in 2004 by CIMIT.

We received 37 submissions from 13 different countries. Each was evaluated by three members of the scientific committee. Based on these reviews, 12 manuscripts were selected for oral presentation and 11 for poster presentation. All accepted manuscripts were allowed a written contribution of equal length and were presented at the meeting in single track. The geographical breakdown of the different institutions presenting their research was: Austria, Belgium, Canada, China, Denmark, France, Germany, Japan, Korea, Netherlands, Spain, and Switzerland.

The quality of the contributions indicates that the symposium is an important milestone in our rapidly growing research area at the confluence of several related disciplines. It was a great pleasure hosting this successful event, and we hope that all attendees enjoyed the intense and stimulating symposium.

July 2006

Matthias Harders
Gábor Székely

Organization

Steering Committee

Gábor Székely, Chair	ETH Zurich, Switzerland
Matthias Harders, Co-Chair	ETH Zurich, Switzerland
Nicholas Ayache	INRIA, France
Stéphane Cotin	CIMIT, USA
Steve Dawson	CIMIT, USA
Hervé Delingette	INRIA, France
Dimitris Metaxas	Rutgers University, USA

Program Committee

Fred Azar	Siemens Corporate Research, USA
Remis Balaniuk	Stanford University, USA
Cagatay Basdogan	Koc University, Turkey
Christos Davatzikos	University of Pennsylvania, USA
Jaydev Desai	Drexel University, USA
James Duncan	Yale University, USA
Anthony Gallagher	RCSI, Ireland
Dieter Hahnloser	University Hospital Zurich, Switzerland
Vincent Hayward	McGill, Canada
Karl-Heinz Höhne	University Hospital Hamburg-Eppendorf, Germany
Arun Holden	Leeds University, UK
David Holmes	Mayo Clinic, USA
Peter Hunter	Auckland University, New Zealand
Erwin Keeve	CAESAR, Germany
Jiri Kroc	University of West Bohemia, Czech Republic
Haymo Kurz	University Freiburg, Germany
Edoardo Mazza	ETH Zurich, Switzerland
Michael Miga	Vanderbilt University, USA
Paul Neumann	CIMIT, USA
Peter Niederer	ETH Zurich, Switzerland
Wiro Niessen	Erasmus MC, The Netherlands
Dinesh Pai	Rutgers University, USA
Richard Robb	Mayo Clinic, USA
Raimundo Sierra	Harvard Medical School, USA
Luc Soler	IRCAD, France
Dominik Szczerba	ETH Zurich, Switzerland
Frank Tendick	UC Berkeley, USA
Daniel Thalman	EPFL, Switzerland
Marc Thiriet	INRIA, France
Simon Warfield	Harvard Medical School, USA

Acknowledgments

The Third International Symposium on Biomedical Simulation would not have been possible without the help and support of many dedicated people.

First we would like to express our gratitude towards the members of the Program Committee for reviewing the submitted papers in a timely and professional manner, as well as for their general support of the concept and organization of the symposium. Special thanks go to Peter Hunter, Peter Leskovsky, and Ian Summers for their last-minute support. Furthermore we are thankful for the encouragement and support of the Steering Committee to continue the traditions laid by the previous meetings in Juan-Les-Pins and Cambridge, MA.

Moreover, our sincere thanks go to the members of the Computer Vision Lab of ETH Zurich for their involvement and support in the organization of the conference. In particular, we thank Axel Krauth and Barbara Widmer for managing the conference website, helping in preparing the camera-ready version of the proceedings, and handling of all local organizational matters.

Sponsoring Institutions

Swiss National Center of Competence in Research on Image Guided and Computer Aided Medical Interventions (NCCR Co-Me).

Table of Contents

Simulation of Biophysical Processes

Systems and Applications

Anatomical Modeling and Tissue Properties

Computer Prediction of Friction in Balloon Angioplasty and Stent Implantation

Denis Laroche[1], Sebastien Delorme[1], Todd Anderson[2], and Robert DiRaddo[1]

[1] Industrial Materials Institute, National Research Council Canada
75 de Mortagne, Boucherville, Quebec, Canada, J4B 6Y4
{denis.laroche, sebastien.delorme,
robert.diraddo}@cnrc-nrc.gc.ca
[2] University of Calgary, Dept. of Medicine
3330 Hospital Drive N.W., Alberta, Canada, T2N 4N1
todd.anderson@calgaryhealthregion.ca

Abstract. The success of balloon angioplasty and stent implantation depends on a balance between two conflicting objectives: maximization of artery lumen patency and minimization of mechanical damage. A finite element model for the patient-specific prediction of balloon angioplasty and stent implantation is proposed as a potential tool to assist clinicians. This paper describes the general methodology and the algorithm that computes device/artery interaction during stent deployment. The potential of the model is demonstrated with examples that include artery model reconstruction, device deployment, and prediction of friction on the arterial wall.

1 Introduction

Balloon angioplasty, the most practiced medical intervention worldwide, consists of dilating a stenosed artery with a polymeric balloon in order to restore blood flow to an acceptable level. Often, a stent is deployed and permanently implanted inside the artery to prevent elastic recoil of the artery. The most frequent complication of angioplasty, restenosis, is an excessive repair reaction of the arterial wall, related to its mechanical damage during the intervention. Restenosis has been shown to be related to two types of mechanical damage: 1) overstretch injury of the arterial wall and 2) denudation of the endothelium (the cell monolayer that lines the interior part of the arterial wall) due to friction with the balloon. The specific contribution of both types of injury to restenosis is still debated [1],[2]. Whether because of patients comeback after 6 months for target vessel revascularization or because of the use of expensive drug-eluting stents, it is generally recognized that restenosis increases by 25 to 30% the total cost of this intervention.

The success of angioplasty depends on a balance between two conflicting objectives: 1) maximizing the final deformation of the artery and 2) minimizing the mechanical damage to the arterial wall. Few research groups have attempted to simulate angioplasty with numerical or analytical models and predict its outcome. Angioplasty simulation, combined with current artery imaging technique such as intravascular

M. Harders and G. Székely (Eds.): ISBMS 2006, LNCS 4072, pp. 1 – 8, 2006.

ultrasound (IVUS), has the potential to become a clinical tool to assist in the selection of an appropriate intervention strategy for a specific patient. This could be done by virtually testing various strategies.

This work presents improvements to a finite element model for predicting the device/artery behavior during angioplasty [3],[4],[5]. These improvements consist of using pre-wrapped balloon models for saving computational time, and deploying both a balloon and stent inside an artery. Calculation of friction work is also proposed as a hypothesized predictor of endothelium denudation. To our knowledge, friction damage to the endothelium has never been considered in angioplasty simulations, possibly because of the difficulty of implementing a robust contact/slip algorithm for deforming bodies. An example of balloon angioplasty and stent implantation simulation on a coronary artery obtained from intravascular ultrasound imaging is presented to demonstrate the potential of the proposed model.

2 Methodology

A finite element modeling software developed for the analysis of large deformations of soft materials is used to solve angioplasty mechanics [3],[4],[5]. The model computes the device/artery interaction and large deformations that occur during device deployment into the diseased artery. It predicts the resulting artery lumen patency, including stress and strain distribution in the arterial wall, for a specific device and inflation pressure. It also predicts the amount of work done by the friction force on the arterial wall.

2.1 Multi-body Contact and Friction

Contact algorithms are usually time consuming and limiting factors in complex finite element applications. They have been traditionally used in explicit software and include collision detection [6],[7], non-penetration constraint and friction/slip capabilities [8],[9]. In this work a multi-body contact algorithm for implicit finite element computation is used [3]. Collisions between virtual nodes and surfaces moving with large displacement steps are detected with an implicit iterative approach that fully respects the non-penetration constraint. Once contact is detected, it is handled with an augmented Lagrange algorithm that computes slip and friction forces. The technique is stable for large displacement increments and is therefore directly applicable to finite deformation analysis.

It is assumed that the mechanical damage to the endothelium is related to the amount of friction exerted by the device during the intervention. In an effort to predict damage, the friction work per surface area is calculated. This value has the advantage of cumulating the amount of friction that occurs during the intervention. Equation 1 gives the friction work density Ω_f as a function of the history of the shear stress τ_f and relative velocity v between the two contacting surfaces.

$$\Omega_f = \int_{t=0}^{T} \vec{\tau}_f(t) \cdot \vec{v}(t) dt \tag{1}$$

The friction shear stress is a vector defined in Equation 2 where p is the contact pressure, μ is the friction coefficient, and v is the relative velocity between the two surfaces.

$$\vec{\tau}_f = p\mu \frac{\vec{v}}{\|\vec{v}\|} \tag{2}$$

2.2 Balloon Model

The angioplasty device model consists of a balloon wrapped around its longitudinal axis (Figure 1a). The balloon is modeled with triangular membrane elements and its mechanical properties are given by the Ogden hyperelastic constitutive equation [4]. The geometry of the wrapped balloon is constructed by mapping the nodes of the deployed balloon onto a wrapped configuration, as illustrated in Figure 1b. Points P_i on the deployed balloon of radius c are mapped onto points $\hat{P}_i$ of the folded balloon of inner radius a and outer radius b. Points P_i can be expressed in polar coordinates as (c,θ), and points $\hat{P}_i$ as $(\hat{r},\hat{\theta})$. A flap starts at point $\hat{P}_1$ and finishes at point $\hat{P}_2$. The angle β between P_1 and P_2 is calculated by Equation 3 where $\phi = 2\pi/n$.

$$\beta = \frac{\phi}{2}\left(\frac{b+a+2c}{b+a}\right) \tag{3}$$

The angle between $\hat{P}_1$ and $\hat{P}_2$ is given by:

$$\alpha = \frac{b+a}{2c}\beta \tag{4}$$

The angles and radii of points P_i are mapped onto points $\hat{P}_i$ using the following rule. If $\theta \in [0,\alpha]$ then

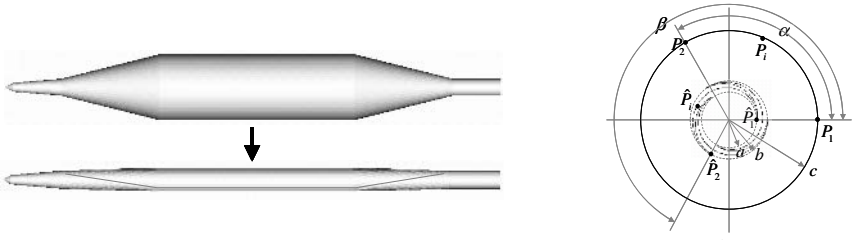
$$\hat{\theta} = \frac{\beta}{\phi}\theta \tag{5}$$

$$\hat{r}(\hat{\theta}) = a + \frac{b-a}{\beta}\hat{\theta} \tag{6}$$

If $\theta \in]\alpha,\beta]$ then

$$\hat{\theta} = \left(\frac{\beta-\phi}{\phi-\alpha}\right)(\phi-\theta)+\phi \tag{7}$$

$$\hat{r}(\hat{\theta}) = a + \frac{b-a}{\beta-\phi}(\hat{\theta}-\phi) \tag{8}$$



a) b)

Fig. 1. Balloon folding: a) balloon before and after folding; b) cross-section of balloon showing mapping of points from deployed configuration to wrapped configuration;

In this example an 8-mm long angioplasty balloon having a diameter of 2.7 mm was used. It was folded into a 3-ply configuration having an inner diameter of 0.7 mm and outer diameter if 1.0 mm. Balloon inflation occurs in two phases: 1) deployment and 2) stretching. Figure 2 illustrates three stages during the balloon deployment phase.

Fig. 2. Steps of balloon cross-section deployment

2.3 Stent Model

A three-dimensional model of a stent (Figure 3) was created using high resolution images of a crimped NIR Royal stent. The stent was meshed with 8-node hexahedral elements. The stainless steel behavior is modeled with a hyperelastic Mooney-Rivlin constitutive model. In order to overcome the high stress resulting from the use of an elastic model to simulate plastic deformation of stainless steel, a Young's Modulus of 6 GPa is used. The 6.2 mm long by 1.4 mm diameter stent is mounted over the balloon shown in Figure 1. Figure 3 shows the predicted deployment mechanics of the device. It can be seen that the extremities of the balloon inflate first. Even at 5 atm, the balloon is not completely deployed and a fold is still visible.

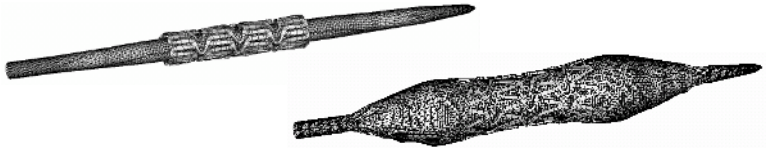


Fig. 3. Stent deployment

2.4 Artery Model

The artery model was reconstructed from intravascular ultrasound (IVUS) pullback images of a 39-mm long artery segment obtained during a percutaneous coronary

intervention (PCI). The 3D model of the artery was built using the Amira software (Mercury Computer Systems, Chelmsford, MA), from which an 11553-node mesh including tetrahedral elements was created [3]. Figure 4 shows a typical segmented image with identified lumen and media-adventitia borders.

The artery is modeled with incompressible solid elements and the Mooney-Rivlin hyperelastic constitutive model. The material constants (c_{10}=0.02, c_{01}=0.0) correspond to a homogeneous shear modulus of 0.04 MPa, as suggested in [10]. The use of an hyperelastic model instead of a more appropriate elasto-plastic has the major limitation of not allowing the prediction of the artery shape after the balloon is removed. However the mechanics of device deployment and its interaction with the arterial wall can be estimated. Tetrahedral elements rather than hexahedral elements are proposed for the artery, thus allowing the use of automatic mesh generation from medical images, and reducing the time required for pre-processing the data.

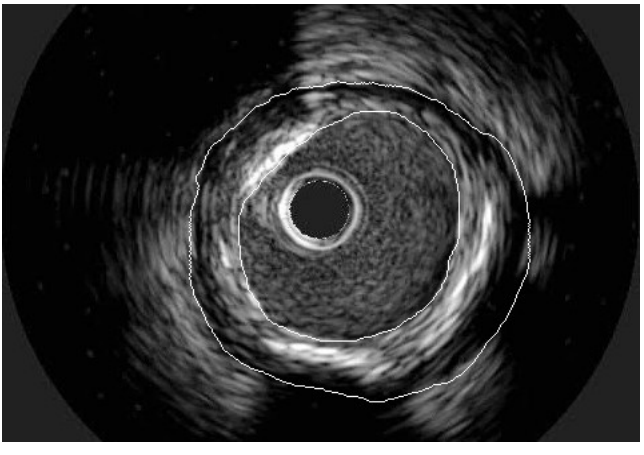


Fig. 4. Example of segmented IVUS image showing lumen and media-adventitia borders

3 Simulation Examples

Numerical simulations of balloon angioplasty and stent implantation in the mid-LAD coronary artery of a patient were performed at the most stenosed segment. In both cases, the extremities of the artery and the device were fixed and the device was slightly stretched axially.

3.1 Balloon Angioplasty

Figure 5 shows predicted balloon inflation steps at 1.4 atm and 5 atm of inflation pressure. It can be observed that contact between the balloon and arterial wall occurs during the deployment phase of the balloon inflation, involving friction slip. Because of boundary conditions applied on the balloon and artery, the balloon deployment is considerably affected by the artery shape and that the slip between the balloon and the artery is important.

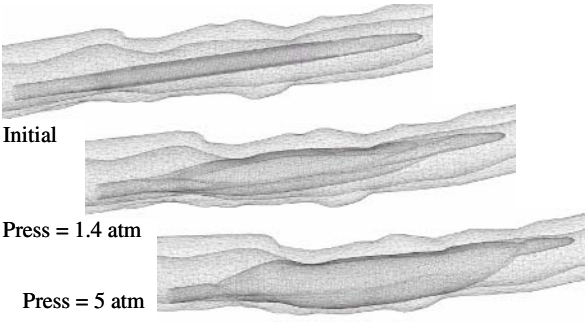


Fig. 5. Balloon angioplasty inside an artery, from 0 to 5 atm inflation pressure

3.2 Direct Stenting

The stent model shown in Figure 3 was deployed inside the same artery segment. Figure 6 shows the deployment at an inflation pressure of 2.8 atm and 10 atm. It can be seen that a much higher pressure is required to re-open the artery due to the presence of the stent.

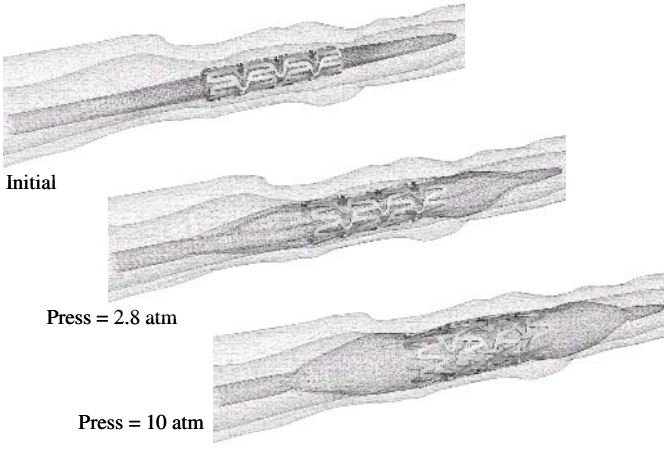


Fig. 6. Deployment of stent inside an artery

As it was the case with unconfined balloon/stent deployment (Figure 3), the extremities of the balloon are inflated first. This implies that the extremities of the stent might be the first to contact the artery. Compared with the predicted balloon angioplasty of the previous example, the balloon deploys into a more regular shape when a stent is present.

3.3 Prediction of Friction Work

The friction work density Ω_f was integrated over the artery inner surface to obtain the cumulative work of the friction force on the endothelium. Figure 7 shows its

evolution as a function of the inflation pressure for both the balloon angioplasty and the stent implantation.

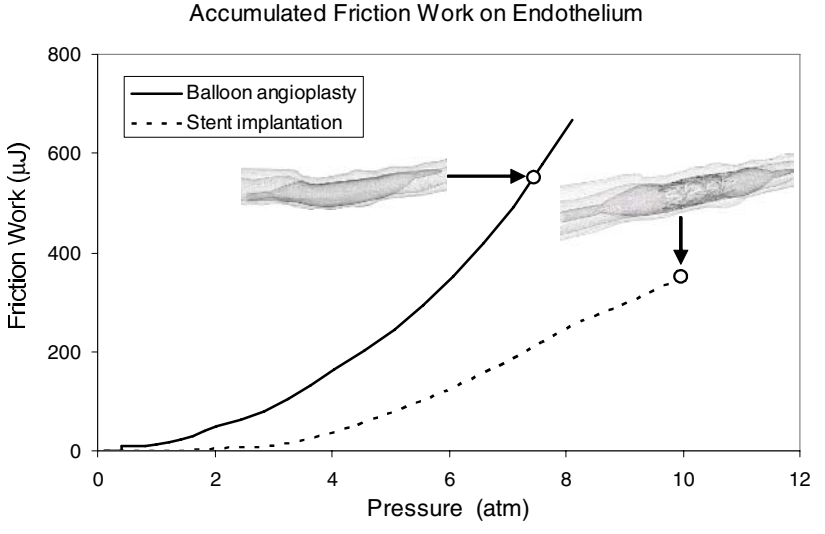


Fig. 7. Predicted cumulative friction work on the arterial wall for balloon angioplasty and stent implantation

Friction is expected to occur during the two phases of balloon inflation. First, during balloon deployment, the balloon flaps slip against the arterial wall. Second, when the balloon is fully deployed, it might start to stretch uniformly while the artery would stretch non-uniformly because of uneven thickness, explaining why friction work continues to increase after the balloon is fully deployed.

The predicted friction work on the endothelium is lower in the presence of the stent, possibly because the friction occurred mostly between the balloon and the stent. This could potentially indicate that stent deployment results in less endothelium damage than balloon angioplasty.

4 Conclusion

In this paper a finite element model for predicting balloon angioplasty and stent implantation from artery imaging was presented. It demonstrated that balloon angioplasty and stent implantation mechanics differ. Most notably, the presence of the stent constrained the balloon to a slower and more symmetrical deployment than when the balloon is deployed alone.

The proper modeling of the interaction between the balloon, the stent and the artery allowed predicting friction work on the endothelium. The presence of the stent resulted in a reduced friction work, suggesting that the stent might protect the artery against friction damage.

Future work will focus on constitutive models for the artery and the use of deformable catheters to further improve the accuracy of the simulation. Experimental work is underway to quantify the relationship between friction work and endothelial denudation. Also, predicted artery response after device removal will also be validated using post-intervention imaging data.

References

1. Clowes, A.W., Clowes, M.M., Fingerle, J., Reidy, M.A.: Kinetics of cellular proliferation after arterial injury. V. Role of acute distension in the induction of smooth muscle proliferation, Lab Invest, Vol. 60 (1989) 360-364
2. Fingerle, J., Au, Y.P., Clowes, A.W., Reidy, M.A.: Intimal lesion formation in rat carotid arteries after endothelial denudation in absence of medial injury, Arteriosclerosis, Vol. 10 (1990) 1082-1087
3. Laroche, D., Delorme, S., Anderson, T., Buithieu, J., DiRaddo, R.: Computer Prediction of Balloon Angioplasty from Artery Imaging, Medicine Meets Virtual Reality 14, J.D. Westwood et al. (Eds), Technology and Informatics, Vol .119 (2006) 293-298
4. Delorme, S., Laroche, D., DiRaddo, R., Buithieu, J.: Modeling polymer balloons for angioplasty: from fabrication to deployment, Proc Annual Technical Conference, ANTEC, SPE, Chicago, IL (2004)
5. Laroche, D., Delorme, S., Buithieu, J., DiRaddo, R.,: A three-dimentional finite element model of balloon angioplasty and stent implantation, Proc Comp Meth Biomech Biomed Eng, Vol. 5, Madrid (2004)
6. Hallquist, J.O., Goudreau, G.L., Benson, D.J.: Sliding interfaces with contact-impact in large-scale lagrangian computations, Comp Meth App Mech Eng, Vol. 51 (1985) 107-137
7. Zhong, Z.H.: Finite element procedures for contact-impact problems. Oxford University Press (1993)
8. Laursen, T.A., Simo, J.C.: A continuum-based finite element formulation for the implicit solution of multibody, large deformation frictional contact problems. Int J Num Meth Eng, Vol. 36 (1993) 3451-3485
9. Puso, M.A., Laursen, T.A.: A mortar segment-to-segment contact method for large deformation solid mechanics. Comp Meth Appl Mech Eng, Vol. 193 (2004) 601-629
10. Holzapfel, G.A., Stadler, M., Schulze-Bauer, C.A.J.: A layer-specific three-dimensional model for the simulation of balloon angioplasty using magnetic resonance imaging and mechanical testing, Ann. Biomed. Eng., Vol. 30 (2002) 753-767

Real Time Simulation of Organ Motions Induced by Breathing: First Evaluation on Patient Data

A. Hostettler[1], S.A. Nicolau[1], C. Forest[1], L. Soler[1], and Y. Remond[2]

[1] IRCAD-Hopital Civil, Virtual-surg, 1 Place de l'Hopital, 67091 Strasbourg Cedex
{alexandre.hostettler, luc.soler}@ircad.u-strasbg.fr
[2] Institut de Mécanique des Fluides et des Solides,
2 rue Boussingault 67000 Strasbourg

Abstract. In this paper we present a new method to predict in real time from a preoperative CT image the internal organ motions of a patient induced by his breathing. This method only needs the segmentation of the bones, viscera and lungs in the preoperative image and a tracking of the patient skin motion. Prediction of internal organ motions is very important for radiotherapy since it can allow to reduce the healthy tissue irradiation. Moreover, guiding system for punctures in interventional radiology would reduce significantly their guidance inaccuracy. In a first part, we analyse physically the breathing motion and show that it is possible to predict internal organ motions from the abdominal skin position. Then, we propose an original method to compute from the skin position a deformation field to the internal organs that takes mechanical properties of the breathing into account. Finally, we show on human data that our simulation model can provide a prediction of several organ positions (liver, kidneys, lungs) at 14 Hz with an accuracy within 7 mm.

1 Introduction

Preoperative 3D CT images are used in many medical contexts. For some of them, the breathing motion is an issue that is taken into account but not compensated. For instance, in radiotherapy the dosimetry computation is performed on a static preoperative CT image although the patient breathes freely during the irradiation. Then, the planned volume targeting is overestimated so as to totally irradiate the tumour. Consequently, healthy tissue irradiations are much more important than if the tumour position was perfectly known during the breathing. In interventional radiology, guiding systems for punctures provide a limited accuracy since the breathing movements are not compensated [10,16].

Respiratory gating techniques [11,17,2] are a first attempt to reduce the breathing influence. The patient is immobilized on the intervention table and his breathing is monitored to synchronize his lung volume to the one during the preoperative CT acquisition. These methods are however constraining: they lengthen the intervention, they are uncomfortable for the patient and the repositioning accuracy, of 1.5 mm on average, can sometimes exceed 5 mm [2].

Therefore, a predictive method to simulate the abdomen and thorax breathing motions would be a great improvement for the previous reported applications.

M. Harders and G. Székely (Eds.): ISBMS 2006, LNCS 4072, pp. 9–18, 2006.

Internal motion during a quiet breathing being between 10 and 25 mm [4], practitioners estimate that a prediction accuracy within 4 mm would bring a significant improvement to the current protocol in radiotherapy and interventional radiology.

Solutions to simulate organ mesh deformations and interactions are numerous [5,7,15,8], however they require the rheological parameters of the patient organs to provide predictive results, which is currently an unsolved issue. Consequently, these methods that were developed for realistic simulation of surgical interventions, are not adapted to intra-operative applications in the operating room.

Sarrut et al. propose to simulate an artificial 3D+t CT image during a breathing cycle from two CT images acquired at inhale and exhale breath-hold. Simulated 3D+t images are built by a vector field interpolation between both CT images [12]. Although this interesting method provides a simulated CT image in real time, it implies a supplementary irradiation of the patient since it requires at least two CT acquisitions whereas only one is done routinely. Moreover, it assumes that the breathing is perfectly reproducible and cyclical. Therefore, this method is limited to patients who are under general anesthesia and intubated.

In this paper we present a new method to provide in real time 3D CT or mesh images that simulate the patient breathing from a single pre-operative CT acquisition. In Sec. 2, we firstly describe qualitatively the breathing motion and show that an estimation of the internal organ position can be obtained from the skin position. In Sec. 3, we propose a method to compute from the skin position a deformation field applied to the internal organs. Finally, we provide in Sec. 4 a first evaluation of our method on human data showing that our simulation can be computed at 14 Hz with a prediction accuracy between 5 and 7 mm.

2 Estimation of the Organ Deformation from the Skin Position

Breathing movements are mainly caused by two sets of muscles: the intercostal muscles and the diaphragm. Those muscles move the thorax and the abdomen. Firstly, intercostal muscles induce the rib rotation around the vertebrae that makes the thorax volume change (cf. Fig. 1). This bone rotation is made possible since the sternum is linked to the ribs through deformable cartilage. Secondly, the diaphragm contraction dilates the lungs, causing the air insufflation. During a contraction, the diaphragm moves down and compresses the viscera, increasing the abdomen volume since most of the internal organs are incompressible and since viscera movements toward the feet is limited by the pelvis. Note that the abdominal muscles, the sterno cleido mastoidien, the scalene muscles, the trapeze muscles, the major dorsal muscles and the major pectoral muscle are involved as well in the breathing movements. However, they can be neglected in our context since they are essentially used during an important physical activity.

This qualitative description of the breathing motion highlights the link between skin and internal organ positions. Indeed, from the skin position, we can

understand how the viscera are compressed, and then the diaphragm position and the lung volume. Therefore, we think that it is possible to estimate the internal organ motion from skin position. Consequently, we consider in our model that the patient inspiration increases the lung volume (and the thorax volume) and pushes the viscera down, which increases the abdomen volume. We assume that the volume of the viscera [13] and that the tissue thickness between skin and organs remain constant (cf. Fig. 1). In addition, we suppose that the viscera have homogeneous mechanical properties.

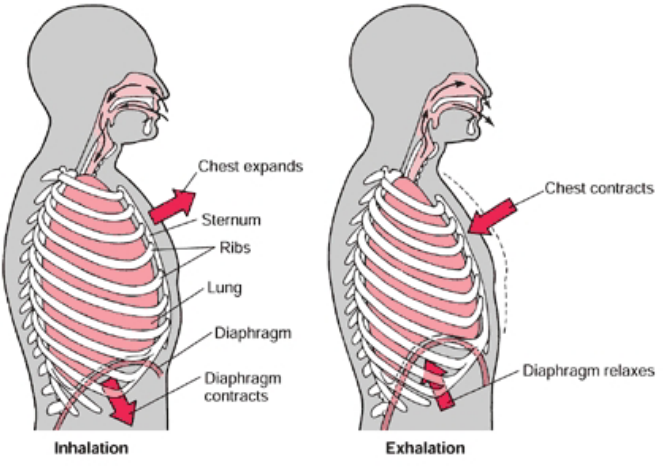


Fig. 1. Qualitative description of breathing. During the breathing, the bone rotation around the vertebrae makes the thorax volume change, and the diaphragm contraction compresses the viscera, causing the abdominal volume increase.

3 Simulation Methodology

3.1 Set Up

Our goal is to provide in real time a 3D deformation field to be applied on a preoperative CT or on the surfacic organ meshes previously extracted from the preoperative CT during a segmentation step. This deformation field is computed from the skin movements that are tracked in real time by an external system. In our set up, the patient is lying on the table (CT or radiotherapy table) and a tracking system provides in real time either a surface reconstruction of his abdominal and thoracic skin or feature points (for instance, some radio-opaque markers stuck on the patient skin before the preoperative CT acquisition). The registration between the CT frame and the tracking system frame is done using either the surfacic skin reconstruction or the feature points. We recall our method needs a segmentation of the 3D CT image into four different zones: skin, bones, lungs and viscera (liver, kidney, spleen, etc.): several software provide efficient tools to perform a quick and accurate organ segmentation [3,14]. In the following we call *internal envelope* the envelope of the lungs and the viscera.

In this section, firstly we explain how the skin, the bones and the internal envelope meshes are computed from the skin position. Secondly, we detail how we generate a deformation field from the internal envelope movement. In the following, we assume that the skin position is tracked using radio-opaque markers uniformly stuck on the patient skin.

3.2 Skin, Bones and Sternum

The displacement $\overrightarrow{D}$ of any point P of the surfacic skin mesh is computed from the displacements $\overrightarrow{V}_1, \overrightarrow{V}_2, \overrightarrow{V}_3, \overrightarrow{V}_4$ of the four closest markers M_1, M_2, M_3, M_4 with the following formula:

$$\overrightarrow{D} = k \cdot \frac{\sum_{i=1}^{4} 1/d_i \cdot \overrightarrow{V}_i}{\sum_{i=1}^{4} 1/d_i} \quad \text{with} \quad k = \frac{\sum_{i=1}^{4} 1/d_i \cdot \| \overrightarrow{V}_i \|}{\sum_{i=1}^{4} 1/d_i}$$

where d_i is the distance between M_i and P.

The rib movement is modelled by a rotation around the vertebra. We assume that the distance between sternum and skin remains constant and that the sternum is attached to the fourth rib pair. We choose the fourth pair since it minimizes the cartilage impact. Note that the simulation of the rib movement has no impact on the volume computation but only on the realism of rib and sternum position.

3.3 Movement of Internal Envelope

We call *internal organs* the union of lungs and viscera. To determine the movement of those organs we first study the movement of their envelope. This *internal envelope* is a convex balloon localized inside the trunk volume (cf. Fig 2). The movement of that balloon is computed in a way similar to the one that was used for the skin (cf. Fig 3): for each vertex, we look for the three closest skin markers and the resulting vertex displacement is computed from the displacements of these markers. Note that this method keeps a constant thickness between the internal envelope and skin.

The viscera go down along a curved axis that corresponds to the gravity centre of the viscera in each slice, filling the current volume of the computed internal envelope from the pelvis to the diaphragm. By hypothesis, viscera are incompressible, therefore a modification of the CT slice surface results in a change in the slice thickness. For the lungs, the slice thickness is uniformly changed to fill the remaining volume of the internal envelope (cf. Fig 2).

3.4 Generation of the Deformation Field

To describe the movement of each point in the internal envelope, we propose the following method to generate a deformation field. In each slice of the internal envelope, the perimeter is divided into segments of identical size, allowing to control the internal deformation of that slice in an homogeneous way (cf. Fig. 4). For each point P we want to displace, we look for the two nearest vertices of

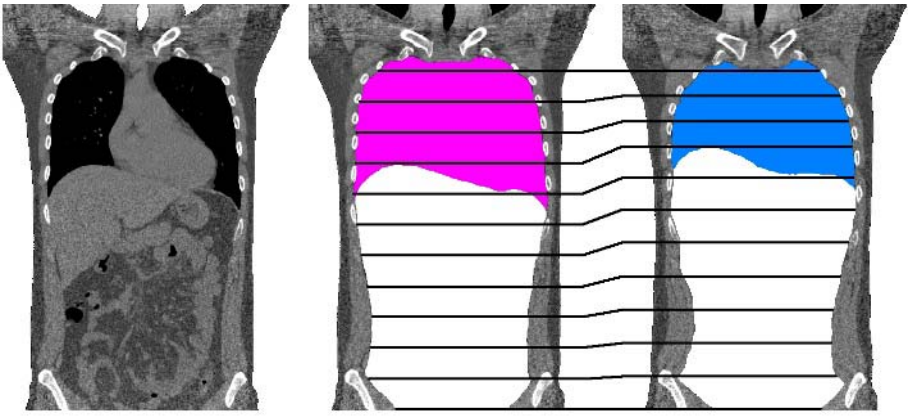


Fig. 2. Left: sagital slice of a patient showing the lungs and the viscera. The *internal envelope* is the convex balloon that contains the lung and the viscera. They are segmented in homogeneous grey and white on the middle image (inspiration phase). Since most of the internal organs are incompressible and since viscera movements toward the feet is limited by pelvis, we modify the thickness of each CT slice of the preoperative CT to keep its own volume constant (right image). The black lines represent the position of several corresponding CT slices from the inspired CT image on left, and the simulated expired image on right.

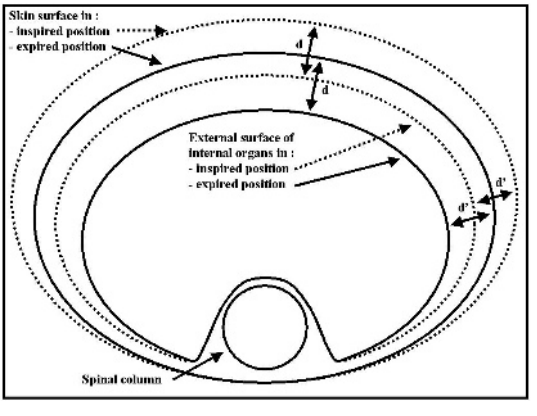


Fig. 3. The motion of the *internal envelope* is computed from the motion of the skin. Indeed, tissues contained between the skin and the internal envelope are almost incompressible, then the thickness between them remains constant.

the envelope B and C, and to the nearest vertex of the mean barycentric axis A. The displacement is then computed with the interpolation formula used for the skin (see Sec. 3.2). This allows to easily animate 3D meshes in real-time. Animating a CT image is trickier as we do not want to know the destination of

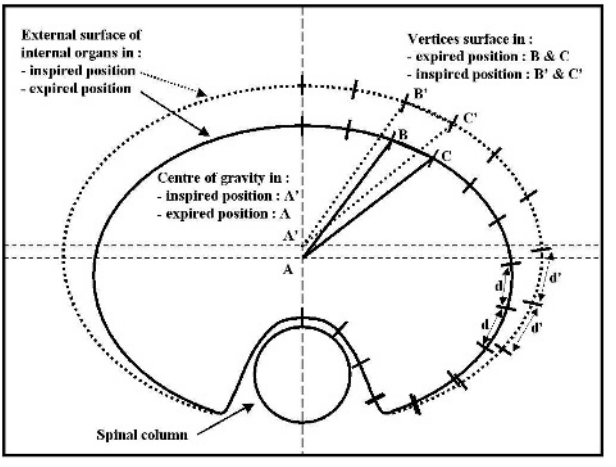


Fig. 4. Deformation field generation. In each slice of the internal envelope, the perimeter is divided into segments of identical size, and the gravity center is computed. Then, the displacement of a point is computed as a linear interpolation of the movement of the two nearest points of the perimeter (B and C) and the gravity center (A).

each pixel, but its provenance. In this case, the reverse application is computed and applied on each point of the 3D CT image.

4 First Evaluation on Human Data

In order to evaluate the computation time needed by our simulation model and its prediction accuracy, we use the data of a human volunteer. Thirty radio-opaque markers have been stuck on his skin (cf. figure 5) and two CT acquisitions were performed in inspiration and expiration positions. Then, the voluntary remained lying on the table and a stereoscopic video tracking of the markers (tracking accuracy < 0.5 mm) was performed during 100 breathing cycles and recorded (see [9] for the used tracking method). The segmentation step (lungs, bones, viscera and radio-opaque markers) was performed by our medical staff for both CT acquisitions.

4.1 A Real Time Simulation Method

To register the expired CT image in the tracking system frame, we used the radio-opaque markers segmented in the CT image and performed a 3D/3D registration [1] with the marker set, among all the one recorded by our tracking system, that minimized the RMS registration error (1.1 mm for this case). Then, we measured the time needed to compute and display the simulated mesh and predicted CT image for the marker positions recorded during the voluntareer's breathing. Our simulation reached a frame rate around 14 Hz on a standard PC (2 GHz, 1Go RAM) with no implemented code optimization.

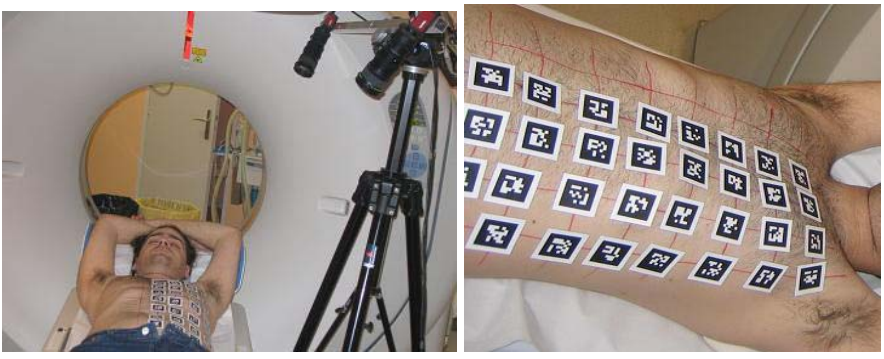


Fig. 5. Set up for the human data acquisition. Thirty markers have been stuck on the volunteer skin, then a stereoscopic video tracking of the markers is performed during 100 breathing cycles and recorded. ARTag markers were used [6].

4.2 An Accurate Prediction

To evaluate the prediction accuracy, we used the inspired CT as the preoperative CT data from which the simulation is computed. Then, we computed the skin marker excursions from the inspired position to the expired position (extracted from the second CT image) and finally computed the deformation field. Therefore, a comparison of the simulated positions of the skin and internal organs with their true positions in the expired CT image is possible.

Qualitative results are visible in Fig. 6. To evaluate quantitatively the discrepancy between each simulated S and true T organ, we computed for each organ the average distance between one mesh vertex and its closest neighbour in the other mesh:

$$\epsilon = \text{mean}_{i \in S}(\| \text{ mesh}_S(i) - \textit{closest point}(\text{mesh}_T(i)) \|)$$

We compute as well the same error for a distance under and above 3 mm:

$$\epsilon_1 = \text{mean}_{i \in S}(\| \text{ mesh}_S(i) - \textit{closest point}(\text{mesh}_T(i)) \| > 3mm)$$

$$\epsilon_2 = \text{mean}_{i \in S}(\| \text{ mesh}_S(i) - \textit{closest point}(\text{mesh}_T(i)) \| < 3mm)$$

This 3 mm threshold was chosen since we observed that the main prediction error of our model is along specific deformation axis. Without this threshold, this error is about 3 mm for the kidney for instance. This does not significantly point out the deformation error at its top and bottom, since it is averaged by the longitudinal points of the two meshes that are very close. Tab. 1 reports the prediction error for kidney, liver, lungs and skin. We add into bracket the percentage of vertex for which the distance is above 3 mm. Note that pancreas, stomach, and spleen are not reported since they could not be segmented in the CT images. Indeed, to limit the irradiation undergone by the voluntary, the CT

Table 1. Results of the system evaluation on three patients

	Left kidney	Right kidney	Liver	Lungs	Skin
Total error ϵ in mm	3.4	3.7	3.7	3.3	1.0
error ϵ_1 in mm (>3mm)	5.5 (51 %)	6.0 (51 %)	6.3 (48 %)	6.5 (39 %)	2.4 (32 %)
error ϵ_2 in mm (<3mm)	1.4	1.5	1.3	1.2	0.4

image was done with a low mA value (under 30 mA whereas a standard image is done with 70 mA) and without contrast agent. Therefore, the CT image was of insufficient quality to allow an accurate segmentation of these organs.

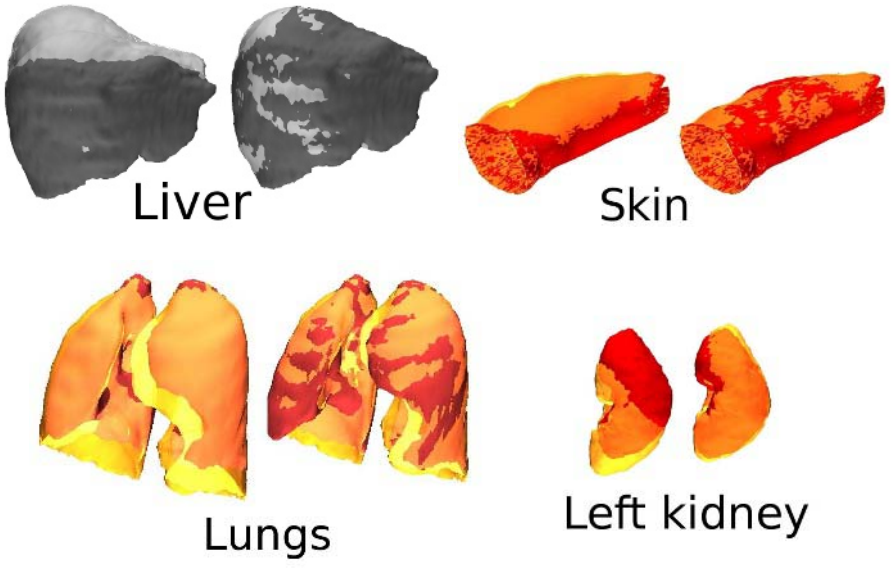


Fig. 6. Qualitative evaluation on organ meshes of our predicted simulation. For each organ: Left: in red: segmented meshes extracted from the expired CT acquisition. In Yellow: segmented meshes extracted from the inspired CT acquisition. Right: in red: segmented meshes extracted from the expired CT acquisition. In Yellow: segmented meshes simulated from the inspired CT acquisition.

5 Conclusion

In this paper we provide a new model to simulate and predict the movement of the internal organs due to breathing motion. Firstly, we showed that it is possible to estimate the displacement of any internal organ from a real time tracking of patient skin. Secondly, we proposed an original method to compute a deformation field from skin position that takes the physical properties of the internal organs into account (incompressibility of viscera, rib rotation, lung deformability). Finally, we evaluated our simulation model with CT-data of a volunteer and

showed that our method is not only able to predict in real-time (14 Hz) internal organ motions, but also to provide a prediction accuracy within 7 mm. Since the observed errors have been measured on the organ boundaries, we are confident that our simulation model can provide a position prediction of a pathology inside an organ with an error under 4 mm.

Our current system uses data from a stereoscopic tracking system that require to stick several radio-opaque markers on the patient skin before the preoperative CT acquisition and during the intervention. We are aware that this is uncomfortable for both the patient and the practitioner. We are currently developing a structured light reconstruction system that will allow to avoid markers on skin and to use its full surfacic information.

In the coming months, we plan to provide a larger evaluation on a database of 10 voluntareers. Then, we will make a careful validation of the prediction accuracy on specific features of some organs (hepatic vessels for instance) using a tracked US probe as a control.

References

1. K. S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3d point sets. *IEEE Transactions on Pattern analysis and machine intelligence*, 9(5):698–700, september 1987.
2. J.M. Balter, K.L. Lam, C.J. McGinn, T.S. Lawrence, and R.K. Ten Haken. Improvement of CT-based treatment-planning models of abdominals targets using static exhale imaging. *Int. J. Radiation Oncology Biol. Phys.*, 41(4):939–943, 1998.
3. L. Bornemann, J. Kuhnigk, V. Dicken, S. Zidowitz, and D. Wormanns. OncoTREAT: A software assistant for oncological therapy monitoring. In *Computer Assisted Radiology and Surgery*, pages 429–434, 2005.
4. M. Clifford, F. Banovac, E. Levy, and K. Cleary. Assessment of hepatic motion secondary to respiration for computer assisted interventions. *Computer Aided Surgery*, 7:291–299, 2002.
5. S. Cotin, H. Delingette, and N. Ayache. A hybrid elastic model allowing real-time cutting, deformations and force-feedback for surgery training and simulation. *The Visual Computer*, 16(8):437–452, 2000.
6. M. Fiala. Artag, an improved marker system based on artoolkit. NRC/ERB-1111 NRC 47166, National Research Council Canada, July 2004.
7. Delingette H. Efficient linear elastic models of soft tissues for real-time surgery simulation. In *MMVR 7 (Medicine Meets Virtual Reality)*, pages 139–151, 1999.
8. Schwartz J.-M. Modelling liver tissue properties using a non-linear visco-elastic model for surgery simulation. *Medical Image Analysis*, 9(2):103–112, 2005.
9. S. Nicolau, L. Goffin, and L Soler. A low cost and accurate guidance system for laparoscopic surgery: Validation on an abdominal phantom. In *Proceedings of ACM Symposium on Virtual Reality Software and Technology (VRST05)*, page Accepted, Monterey, 2005.
10. S. Nicolau, X. Pennec, L. Soler, and N. Ayache. A complete augmented reality guidance system for liver punctures: First clinical evaluation. In *Proceedings MICCAI05*, page submitted, 2005.

11. V. Remouchamps, F. Vicini, M. Sharpe, L. Kestin, A. Martinez, and J. Wong. Significant reductions in heart and lung doses using deep inspiration breath hold with active breathing control and intensity-modulated radiation therapy for patients treated with locoregional breast irradiation. *Int. J. Radiation Oncology Biol. Phys.*, 55:392–406, 2003.
12. D. Sarrut, V. Boldea, S. Miguet, and C. Ginestet. Simulation of 4d ct images from deformable registration between inhale and exhale breath-hold ct scans. *Medical physics*, 33(3):605–617, 2006.
13. T. Secomb. A theoretical model for the elastic properties of very soft tissues. *Biorheology*, 38(4):305–317, 2001.
14. L. Soler, S. Nicolau, J. Schmid, X. Pennec, C. Koehl, N. Ayache, and J. Marescaux. Virtual reality and augmented reality in digestive surgery. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR'04)*, page ??, November 2004.
15. Kühnapfel U. Endoscopic surgery training using virtual reality and deformable tissue simulation. *Computer and Graphics*, 24(5):671–682, 2000.
16. F. Wacker, S. Vogt, A. Khamene, J. Jesberger, S. Nour, D. Elghort, F. Sauer, J. Duerk, and J. Lewin. An augmented reality system for mr image-guided needle biopsy: Initial results in a swine model. *Radiology*, 238(2):497–504, February 2006.
17. J. Wong, M. Sharpe, D. Jaffray, V. Kini, J. Robertson, J. Stromberg, and A. Martinez. The use of active breathing control (abc) to reduce margin for breathing motion. *Int. J. Radiation Oncology Biol. Phys.*, 44(4):911–919, 1999.

Efficient 3D Finite Element Modeling of a Muscle-Activated Tongue

Florian Vogt[1], John E. Lloyd[1], Stéphanie Buchaillard[2], Pascal Perrier[2], Matthieu Chabanas[2], Yohan Payan[3], and Sidney S. Fels[1]

[1] Dept. of ECE, University of British Columbia, Vancouver, Canada
[2] Institut de la Communication Parlée - INPG, 38031 Grenoble, France
[3] Laboratoire TIMC-GMCAO, Faculté de Médecine, 38700 La Tronche, France

Abstract. We describe our investigation of a fast 3D finite element method (FEM) for biomedical simulation of a muscle-activated human tongue. Our method uses a linear stiffness-warping scheme to achieve simulation speeds which are within a factor 10 of real-time rates at the expense of a small loss in accuracy. Muscle activations are produced by an arrangement of forces acting along selected edges of the FEM geometry. The model's dynamics are integrated using an implicit Euler formulation, which can be solved using either the conjugate gradient method or a direct sparse solver. To assess the utility of this model, we compare its accuracy against slower, but less approximate, simulations of a reference tongue model prepared using the FEM simulation package ANSYS.

1 Introduction and Contributions

There is an increasing need for fast, accurate simulation of the human oral pharyngeal anatomy to support research activities in medicine, dentistry, and linguistics. Applications include: the study of physiological activities (such as speech production, breathing, and swallowing), surgical and dental training, and outcome prediction for clinical procedures. Dynamic anatomical models also allow inverse modeling, in which simulation is used to estimate physiological quantities (such as muscle activation levels) which are difficult to measure directly on live subjects.

The utility of a dynamical model is highly dependent on the speed with which it can be simulated. Interactive rates are essential for haptic applications and important for researchers to interactively test the system's dynamics. Likewise, for medical practitioners, real-time interaction expands the range of possibilities for exploring "what-if" scenarios for training and pre-operative planning.

The tongue is the central organ of the oral pharyngeal anatomy and is complicated to model due to its deformable nature and intrinsic muscular activation. In this paper, we show that it is possible to rapidly calculate the dynamics of a finite element model of a muscle activated tongue (Figure 1), with reasonable accuracy, using a stiffness-warping technique such as that described in [1]. We

M. Harders and G. Székely (Eds.): ISBMS 2006, LNCS 4072, pp. 19–28, 2006.

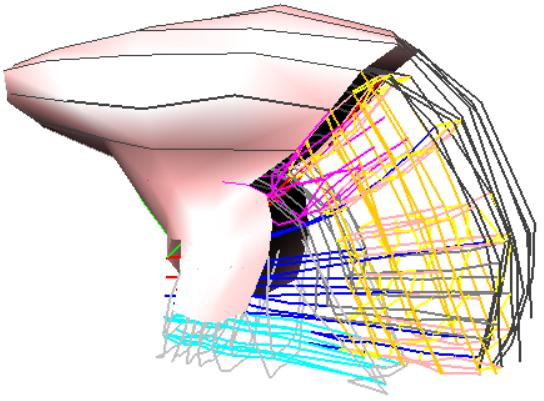


Fig. 1. Tongue model, showing its surface mesh and (in cutaway) the FEM edges corresponding to muscle fibres

build on the work of [2], who developed an FEM tongue model based on hexahedral elements and simulated it in ANSYS, which provided accurate solutions, but at very large computational cost. We test the accuracy of our faster approach by comparing it to models computed using ANSYS. The contributions of our work include:

- Combining stiffness-warping with muscle forces acting along FEM edges to create a fast model of muscle-activated tissue;
- Demonstrating that this type of model can be integrated using an implicit integrator that can be solved with a conjugate gradient solver;
- Testing the accuracy of our approach against a reported reference tongue model.

We anticipate that the techniques described here can be applied to other types of muscle-activated tissue.

This work is part of a larger project ArtiSynth [3], which is an open-source, Java-based, biomechanical simulation platform focused on the vocal tract and upper airway. One of the aims of ArtiSynth is to encourage collaboration and incremental development among scientific and medical researchers by making source code and model data easily available to the community. Source code and demos for the tongue model described in this paper can be obtained from `www.artisynth.org`.

2 Related Work

The tongue has been modeled in a wide variety of ways by various researchers. Parametric models of the tongue's shape have been developed using statistical methods [4,5] and spline descriptions [6,7]. A physiological model is described in

[8]. Dynamic models have been constructed using both mass-spring systems [9] and finite element methods [10,11,12]. A recent survey [13] describes existing methods in detail.

In addition to being efficient to simulate (ideally at interactive rates) and being validated against real measurements, an effective tongue model must provide:

- Emulation of both tissue and muscle fibre;
- Large deformations, particularly at the tip;
- Incompressible (hyperelastic) and non-linear tissue deformation.

Finite element methods provide a good solution for emulating both tissue and muscle fibre, and have a long tradition in Engineering [14,15]. FEM models also provide greater stability and accuracy than mass-spring models; however, current FEM solutions [10,11,16,17] do not compute in real-time. Recent developments in the fields of physical-based animation [1,18] and surgical simulation [19] provide finite element algorithms which can run in real-time, albeit with less (or even unknown) accuracy, to provide plausible results even for large deformations.

Modeling the tongue's muscle activation is particularly important for simulating physiological activities such as speech and swallowing. Most current approaches to muscle tissue modeling apply Hill's non-linear spring model [20] to either mass-spring systems [9], finite elements [21,18], or Cosserat models [22]. For consistency with the reference tongue model we use a muscle model based on [12].

3 Adaptation of the Reference Tongue Model Geometry

For this work, we use the 3D FEM tongue geometry developed in [12], which is shown in Figure 1 and was developed from medical image data. This geometry contains 946 nodes, connected to form 740 hexahedral elements. These hexahedra were further subdivided into 3700 tetrahedra (using the optimal number of five tetrahedra per hexahedron) as our present implementation of the stiffness-warping algorithm requires tetrahedral geometry.

4 Dynamic Modeling of the Tissue and Muscle Activation

To create a fast deformable model of the FEM tongue, we have adapted the stiffness-warping FEM approach developed by Mueller et al. [1].

Following Mueller, we use a lumped-mass FEM in which the positions of the vertices, collectively denoted by $\mathbf{x}$, obey the dynamic equation

$$\mathbf{M}\,\ddot{\mathbf{x}} + \mathbf{C}\,\dot{\mathbf{x}} + \mathbf{K}\,\mathbf{x} + \mathbf{f}_0 = \mathbf{f}_{\text{ext}} + \mathbf{f}_m \tag{1}$$

where $\mathbf{M}$ is the (diagonal) mass matrix, $\mathbf{K}$ is the stiffness matrix (based on linear Cauchy strain), $\mathbf{C}$ is the damping matrix, $\mathbf{f}_0$ is the rest-position stiffness forces, $\mathbf{f}_{\text{ext}}$ is the external forces, and $\mathbf{f}_m$ is the muscle forces (described below). This is a

linear FEM system, *except* that $\mathbf{K}$ and $\mathbf{f}_0$ are modified at each time step to factor out the effect of elemental rotations, effectively re-referencing each element to its rest pose at the end of the previous time step. This technique, known as *stiffness-warping*, allows the model to accommodate large rotational deformations with minimal volumetric distortion and a computational cost which is similar to that for linear FEMs.

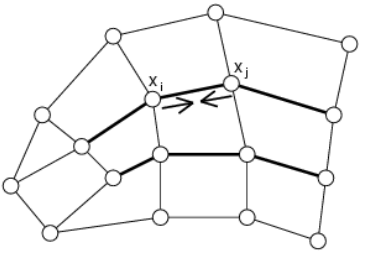


Fig. 2. Muscle forces are created by applying forces along lines between specific pairs of FEM nodes (indicated by dark lines)

To model the tongue's internal muscle activation, we designate specific collections of finite elements as muscles, each associated with an activation level (specified in Newtons). Certain FEM edges within each muscle are selected to act as "fibres", along which a uniform contractile force is exerted (Figure 2). If i and j are the nodes associated with such an edge, with positions $\mathbf{x}_i$ and $\mathbf{x}_j$, then the muscle forces acting on nodes i and j are given by $\mathbf{f}_{ij}$ and $-\mathbf{f}_{ij}$, where

$$\mathbf{f}_{ij} = \alpha_{ij}\mathbf{u}_{ij}, \quad \mathbf{u}_{ij} \equiv \frac{\mathbf{x}_j - \mathbf{x}_i}{l_{ij}}, \quad l_{ij} \equiv \|\mathbf{x}_j - \mathbf{x}_i\|, \tag{2}$$

and α_{ij} is the product of the muscle activation level and a weighting factor based on the masses of the nodes in the muscle. Note that this is *not* a Hill model [20], since the force does not depend on the edge length, and there is no damping (experiments showed that damping was adequately accounted for by the damping matrix $\mathbf{C}$ in (1)). The sum of all $\mathbf{f}_{ij}$ acting on each node produces the force term $\mathbf{f}_m$ in (1). In all, our model contains 11 muscles and a total of 1485 fibres. More details on the motivation for the muscle model and the selection of the muscles and fibres are given in [17].

Since we use a lumped-mass model, boundary conditions are effected either by applying external constraint forces to $\mathbf{f}_{\text{ext}}$ for the appropriate nodes (Lagrange multiplier approach), or by fixing certain nodes, with externally assigned positions and velocities, and then removing them from the dynamic calculation (1). The latter method is used to connect the tongue to the jaw and hyoid bone, using 117 surface nodes associated with the muscle attachment points to these structures. In the results described below, these attachment nodes are fixed rigidly. However, we are currently integrating our tongue model with the jaw/laryngeal model described in [23], and in this setting, the dynamics of the attachment nodes

will be integrated with rigid jaw and hyoid structures, using the forces generated by the surrounding finite elements. We are also implementing a scheme to handle collisions with the teeth and soft palate, which will be implemented using soft external constraint forces applied to $\mathbf{f}_{\text{ext}}$ for the colliding nodes.

5 Dynamic Integration

For greater stability, we integrate (1) using an implicit Euler method. The formulation is similar to that described in [1], except that we must also account for $\partial \mathbf{f}_m / \partial \mathbf{x}$, since $\mathbf{f}_m$ is formed from the $\mathbf{f}_{ij}$, which in turn depend on the direction vectors $\mathbf{u}_{ij}$ and hence $\mathbf{x}$. Letting h denote the integrator step size, $\mathbf{x}^i$ and $\dot{\mathbf{x}}^i$ the node positions and velocities at step i, and $\mathbf{F}_m \equiv \partial \mathbf{f}_m / \partial \mathbf{x}$, the implicit formulation is

$$(\mathbf{M} + h\mathbf{C} + h^2\mathbf{K} - h^2\mathbf{F}_m)\,\dot{\mathbf{x}}^{i+1} = \mathbf{M}\,\dot{\mathbf{x}}^i - h(\mathbf{K}\,\mathbf{x}^i + \mathbf{f}_0 - \mathbf{f}_m - \mathbf{f}_{\text{ext}}) \qquad (3)$$

where at each step we solve for $\dot{\mathbf{x}}^{i+1}$. The computation of this solution is made easier by the fact that the matrix on the right side of (3) is symmetric positive definite (SPD):

Theorem 1. *The matrix on the right side of (3) is SPD.*

Proof: Without the $-h^2\mathbf{F}_m$ term, we have the original matrix in [1], which is SPD. We will show that $\mathbf{F}_m$ is symmetric negative semi-definite (SNSD), so that scaling it by $-h^2$ yields a symmetric positive semi-definite (SPSD) matrix, which when added it to the original SPD matrix yields another SPD matrix. Since $\mathbf{f}_m$ is the sum of force contributions for each fibre i-j, $\mathbf{F}_m$ is the sum of Jacobian matrices $\mathbf{F}_{ij}$ for each fibre, which are formed from $\partial \mathbf{f}_{ij} / \partial \mathbf{x}_j$ and $\partial \mathbf{f}_{ij} / \partial \mathbf{x}_i$. Referring to (2), it is fairly easy to show that

$$\partial \mathbf{f}_{ij} / \partial \mathbf{x}_j = \frac{\alpha_{ij}}{l_{ij}}(\mathbf{I} - \mathbf{u}_{ij}\mathbf{u}_{ij}^T) \quad \text{and} \quad \partial \mathbf{f}_{ij} / \partial \mathbf{x}_i = -\partial \mathbf{f}_{ij} / \partial \mathbf{x}_j$$

so that, ignoring zero rows and columns, $\mathbf{F}_{ij}$ has the form

$$\mathbf{F}_{ij} = \begin{pmatrix} \partial \mathbf{f}_{ij}/\partial \mathbf{x}_i & \partial \mathbf{f}_{ij}/\partial \mathbf{x}_j \\ -\partial \mathbf{f}_{ij}/\partial \mathbf{x}_j & -\partial \mathbf{f}_{ij}/\partial \mathbf{x}_i \end{pmatrix} = -\begin{pmatrix} -1 \\ 1 \end{pmatrix}^T (\partial \mathbf{f}_{ij}/\partial \mathbf{x}_j) \begin{pmatrix} -1 \\ 1 \end{pmatrix}.$$

Then, since it is possible to show that $(\mathbf{I} - \mathbf{u}_{ij}\mathbf{u}_{ij}^T)$, and hence $\partial \mathbf{f}_{ij} / \partial \mathbf{x}_j$, is SPSD, $\mathbf{F}_{ij}$ is the negative of a SPSD matrix and is hence SNSD. Finally, since $\mathbf{F}_m$ is formed by summing $\mathbf{F}_{ij}$, it too must be SNSD. □

6 Implementation

Our tongue model is implemented in Java, using the modeling and numeric library support of ArtiSynth [3], which provides a framework for creating and interconnecting various kinds of dynamic and parametric models to form a complete integrated biomechanical system.

Since execution speed is an important issue, it should be mentioned that the dynamic native code compilers (e.g., Hotspot) provided by current Java implementations usually produce execution speeds that approach compiled C/C++ code [24].

To solve the system (3), we used either the Pardiso sparse solver [25], or a conjugate gradient (CG) method; further details are given in Section 8.

7 Results: Accuracy

In this section we compare the accuracy of our stiffness-warping FEM implementation (denoted as WRP) with two methods implemented using the industry standard FEM package ANSYS: (1) a linear small-deformation model (LSD), and (2) a hyperelastic Mooney-Rivlin solid model (HYP). All models used the same tetrahedral meshing described in Section 3.

The tissue elasticity parameters were obtained from the experimental work reported by [17]. For the WRP and LSD models, we used a Young's modulus of $E = 6912$ and a Poisson's ratio of $\nu = 0.49$. For HYP, we set $C_1 = 1152$, $C_2 = 540$, and $\nu = 0.49$. All models used a Raleigh damping (i.e., $\mathbf{C} = \alpha\mathbf{M} + \beta\mathbf{K}$) with $\alpha = 6.22$ and $\beta = 0.11$.

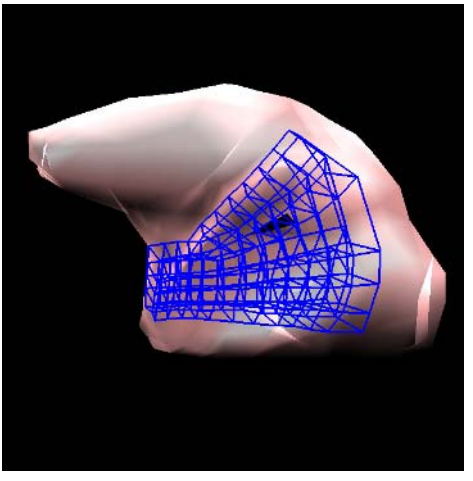
These models were used to simulate a set of five tasks in which a constant excitation was applied to one or more tongue muscles for 1.2 seconds and observed at a rate of 10 ms. The tasks are named and described in Table 1. The HYP and LSD models were computed using a variable rate ANSYS integrator, while our WRP model was computed using the single step implicit integration scheme of (3) with a fixed time step of 10ms.

To assess model accuracy, the deformations resulting from WRP and LSD were compared against those of the HYP (which was considered to be the most accurate and so was used as a reference). Specifically, the deformations $\mathbf{u}_i$ of a set of ten nodes lying on the tongue's mid-sagittal plane were compared against the reference deformations $\mathbf{u}_{ri}$ resulting from the HYP model. The deformation error e_i at each sample point was then computed simply as

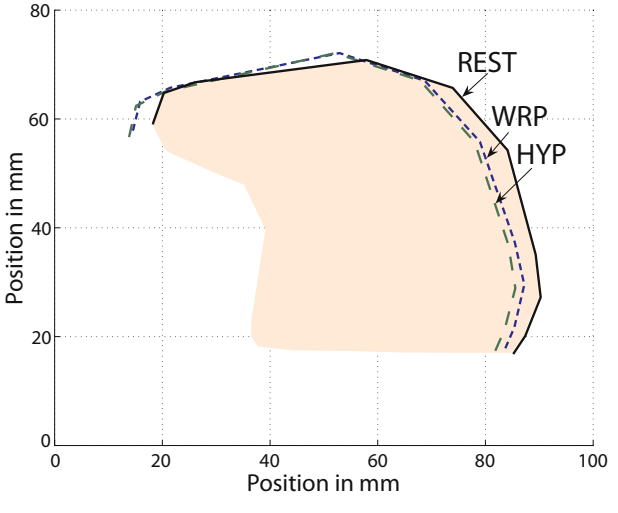
$$e_i = \|\mathbf{u}_{ri} - \mathbf{u}_i\|. \tag{4}$$

Table 1. Muscle activation tasks and end-task deformation errors of these tasks (compared to HYP), resulting from the methods LSD and WRP

Task	Muscle activations	LSD (mm)		WRP (mm)	
		max	mean	max	mean
A	posterior genioglossus (2.0N)	2.3	1.0	1.3	1.0
B	anterior genioglossus (0.5N)	3.2	1.6	1.0	0.9
C	hyoglosse (2.0N)	3.3	1.8	1.1	0.9
D	transversalis (2.0N)	1.2	0.6	0.6	0.4
E	inferior longitudinalis (0.5N)	1.4	1.1	0.9	0.8



(a)



(b)

Fig. 3. (a) Tongue at rest before activation, showing the fibres of the posterior genioglossus muscle; (b) Nodes in the tongue's mid-sagittal plane before activation (REST) and after activation of Task A, as modeled by both WRP and HYP

The mean and maximum of e_i were used to gauge the overall deformation error. Table 1 shows these values for both the WRP and LSD at the end points of each of the tasks. Figure 3(a) shows the unactivated tongue model, while 3(b) shows the mid-sagittal plane nodes before activation, and after activation for Task A as modeled by both HYP and WRP.

Figure 3(b) suggests that the deformations produced by our stiffness-warping model do in fact adhere quite closely to those produced by the hyper-elastic reference model. This is supported more quantitatively by Table 1, where the mean error for WRP is alway within 1 mm, and the maximum error is close to this as well, which is 2 to 3 times better than the results for LSD.

8 Results: Speed and Stability

Computation times for the results reported above were markedly faster for the WRP model as compared with the ANSYS LSD and HYP models, with the former requiring only about 10 CPU seconds per simulated second, while ANSYS required about 600 CPU seconds. All tests were run on a 2.8GHz Pentium IV single processor computer.

For the implicit integration step (3), we used the Pardiso spare solver [25]. We also used a conjugate gradient (CG) solver, although this was slower than Pardiso because we did not use a preconditioner and so about 300 iterations were required to achieve equivalent accuracy.

The per-step computation times were 100ms and 600ms for Pardiso and for unpreconditioned CG, respectively, with CG requiring about 2ms per iteration. About 2ms of both these times was required to update the $\mathbf{K}$ and $\mathbf{f}_0$ terms of (1), as required by the stiffness-warping algorithm, and so this overhead is small. Using the Pardiso solver with a 10ms step size resulted in a simulation speed of about 10 seconds to 1 real second.

With regard to stability, we found that our implicit solver was stable at time steps of up to 20ms, assuming that muscle activations were incorporated into the implicit scheme as described in Section 5. Without this, the stability limit is around 5ms.

9 Conclusion and Future Work

In this work we introduce a fast and stable finite element model and compare its performance to ANSYS FEM simulation using a previously published reference tongue model. The comparison required the adaptation of the hexahedral tessellation of the reference tongue model to a tetrahedral tessellation. Our method admits simulations speeds that are within a factor 10 of real-time, at the expense of a small loss in model accuracy.

We are considering several ways to improve the speed of our model. The first is to use a multi-threaded version of Pardiso, on a machine with several processor cores. Another possibility is to use a preconditioned CG method, which should greatly reduce the number of CG iterations required. Finally, we intend to investigate reduced coordinate approaches, along the lines of [26,27].

Continuing this work in other directions, we plan to apply these methods to other muscle groups and model the interaction with other anatomical substructures of the vocal tract.

References

1. Mueller, M., Gross, M.: Interactive virtual materials. In: Proceedings Graphics Interface. (2004) 239–246
2. Gerard, J., Perrier, P., Payan, Y. In: 3D biomechanical tongue modelling to study speech production. Psychology Press: Sydney, Australia (In Press)
3. Fels, S., Vogt, F., van den Doel, K., Lloyd, J., Stavness, I., Vatikiotis-Bateson, E.: Artisynth: A biomechanical simulation platform for the vocal tract and upper airway. Technical Report TR-2006-10, Computer Science Dept., University of British Columbia (2006)
4. Badin, P., Bailly, G., Raybaudi, M., Segebarth, C.: A three-dimensional linear articulatory model based on mri data. In: Proceedings of the International Conference of Spoken Language (ICSLP). (1998) 14–20
5. Engwall, O.: A 3D tongue model based on MRI data. In: Proceedings of the International Conference of Spoken Language (ICSLP). (2000)
6. Stone, M., Lundberg, A.: Three-dimensional tongue surfaces from ultrasound images. In: SPIE Proc. (1996) 168–179
7. King, S.A., Parent, R.E.: A 3d parametric tongue model for animated speech. JVCA **12**(3) (2001) 107–115
8. Takemoto, H.: Morphological analysis of the human tongue musculartureforthree-dimensional modeling. J. Sp. Lang. Hear. Res. **44** (2001) 95–107
9. Dang, J., Honda, K.: Construction and control of a physiological articulatory model. JASA **115**(2) (2004) 853–870
10. Wilhelms-Tricarico, R.: Physiological modeling of speech production: methods for modeling soft-tissue articulators. JASA **97**(5) (1995) 3085–98
11. Payan, Y., Perrier, P.: Synthesis of v-v sequences with a 2d biomechanical tongue model controlled by the equilibrium point hypothesis. Speech Communications **22**(2) (1997) 185–205
12. Gerard, J., Wilhelms-Tricarico, R., Perrier, P., Payan, Y.: A 3d dynamical biomechanical tongue model to study speech motor control. Recent Research Developments in Biomechanics **1** (2003) 49–64
13. Hiiemae, K.M., Palmer, J.B.: Tongue movements in feeding and speech. Crit Rev Oral Biol Med **14** (2003) 430–449
14. Bathe, K.J.: Finite element procedures. Prentice Hall (1996)
15. Zienkiewicz, O., Taylor, R.: The finite element method. Oxford (2000)
16. Dang, J., Honda, K.: A physiological articulatory model for simulating speech production process. JASJ **22**(6) (2001) 415–425
17. Gerard, J., Ohayon, J., Luboz, V., Perrier, P., Payan, Y.: Indentation for estimating the human tongue soft tissues constitutive law: application to a 3d biomechanical model to study speech motor control and pathologies of the upper airways. LNCS **3078** (2004) 77–83
18. Teran, J., Sifakis, E., Blemker, S., Hing, N.T., V., L., C., Fedkiw, R.: Creating and simulating skeletal muscle from the visible human data set. In: IEEE TVCG (in press). (2005)
19. Cotin, S., Delingette, H., Ayache, A.: Real-time elastic deformations of soft tissues for surgery simulation. IEEE Trans. Vis. & CG. **5**(1) (1999) 62–73
20. Hill, A.: The heat of shortening and the dynamic constants of muscle. Proc Roy Soc B **126** (1938) 136–195
21. Gladilin, E., Zachow, S., Deuflhard, P., Hege., H.C.: Virtual fibers: A robust approach for muscle simulation. In: Proc MEDICON. (2001) 961–964

22. Pai, D.K., Sueda, S., Wei., Q.: Fast physically based musculoskeletal simulation. ACM Trans Graph (2005)
23. Stavness, I., Hannam, A.G., Lloyd, J.E., Fels, S.: An integrated dynamic jaw and laryngeal model constructed from ct data. 3rd International Symposium on Biomedical Simulation (Springer LNCS) (2006)
24. Nikishkov, G.: Java performance in finite element computations. In: Proc Appl Sim & Mod. (2003) 410
25. Schenk, O., Röllin, S., Hagemann, M.: Recent advances in sparse linear solver technology for semiconductor device simulation matrices. In: IEEE SISPAD. (2003) 103–108
26. James, D.L., Pai, D.K.: Artdefo: Accurate real time deformable objects. In: Proceedings of the International Conference on Computer Graphics and Interactive SIGGRAPH. (1999) 65–72
27. Barbic, J., James, D.L.: Real-time subspace integration for st.venant-kirchhoff deformable models. ACM Trans on Graphics **24** (2005) 982 – 990

A 3-D Computational Model for Multicellular Tissue Growth

Lenny Tang and Belgacem Ben Youssef

School of Interactive Arts and Technology
Simon Fraser University
Surrey, British Columbia, V3T 5X3, Canada
{lctang, byoussef}@sfu.ca

Abstract. We report the development of a computational model for the growth of multicellular tissues using a discrete approach based on cellular automata to study the tissue growth rates and population dynamics of two different populations of migrating and proliferating mammalian cells. Cell migration is modeled using a discrete-time Markov chain approach and each population of cells has its own division and motion characteristics that are based on experimental data. A large number of parameters allow for a detailed study of the population dynamics. This permits the exploration of the relative influence of various system parameters on the proliferation rate and some other aspects of cell behavior such as average speed of locomotion.

1 Introduction

A primary goal of tissue engineering is to create three-dimensional tissues with the proper structure and function. Natural tissues are multicellular and have a specific three-dimensional architecture. This structure is supported by the extracellular matrix (ECM). The ECM often has the form of a three-dimensional network of cross-linked protein strands. In addition to determining the mechanical properties of a tissue, the ECM plays many important roles in tissue development. Biochemical and biophysical signals from the ECM modulate fundamental cellular activities, including adhesion, migration, proliferation, differentiation, and programmed cell death [1].

Scaffold properties, cell activities like adhesion or migration, and external stimuli that modulate cellular functions are among the many factors that affect the growth rate of tissues. Hence, the development of bio-artificial tissue substitutes involves extensive and time-consuming experimentation. The availability of computational models with predictive abilities will greatly speed up progress in this area. This research focuses on the development of a model to simulate the growth of three-dimensional tissues consisting of more than one cell type. Specifically, this model can be used to study how the overall tissue growth rate is affected by:

- cell migration speed;
- the initial density of the seed cells; and
- their spatial distribution.

M. Harders and G. Székely (Eds.): ISBMS 2006, LNCS 4072, pp. 29–39, 2006.

The rest of the paper is organized as follows: Section 2 describes some of the earlier work. Section 3 introduces the concepts of cellular automata. Section 4 discusses the modelling of the biological system. In Section 5, we discuss the sequential simulation results. We conclude the paper in Section 6.

2 Previous Work

Earlier studies by other investigators developed cellular automata models for 2-D problems involving aggregation and self-organization of the cellular slime mold *Dictyostelium Discoideum* or interactions between extracellular matrix and fibroblasts ([2], [3], and [4]). In addition, some of the previous research involved the development of computational models that are deterministic or stochastic in nature ([5] and [6], respectively).

Ben Youssef developed a computational model for tissue growth in three-dimensional scaffolds using a single type of mammalian cells [7]. Our work presented here builds on this 3-D model by extending its functionality to incorporate a variable number of cell types. The objective is to build theoretical models and computer simulation tools for tissue engineering [8].

3 Cellular Automata

Cellular automata (CA) were originally introduced by John von Neumann and Stan Ulam as a possible idealization of biological systems with a particular purpose of modeling biological self-reproduction [9]. This approach has been used since then to study a wide variety of physical, chemical, biological, and other complex natural systems.

The use of cellular automata in modeling various systems including biological ones has some advantages that include providing a computationally proficient technique for analyzing the collective properties of a network of interconnected cells as well as an alternative approach involving discrete coordinates and variables to represent the complex dynamic system where the model behaviour is completely specified by a simple operating mechanism in terms of local relations. This may be sufficient to support a whole hierarchy of structures and phenomena.

4 Model Development

The growth of tissue is a complex biological process. The migration and proliferation of mammalian cells is characterized by the following three subprocesses: cell division, cell motion, and cell collision. Due to space limitations, we will omit discussing the steps involved in modeling these subprocesses. The reader is referred to [7] and [10] for related descriptions.

4.1 States of Cellular Automata

The cellular space is comprised of N_t=$N_x \times N_y \times N_z$ computational sites. The automata in this space are governed by the von Neumann's neighborhood in three dimensions.

That is, the 6 neighbors of a cell c are to its east, north, west, south, and directly above and below it. Each occupied computational site in the model must describe the current state of a given cell. According to our modeling steps above, the requisite state information is as follows:

1. The type of cell.
2. The direction of cell motion.
3. The speed of the cell.
4. The remaining time until a change of direction.
5. The remaining time until the next cell division.

Hence, the state of a cell can be defined as a single eight-digit number in the following way:

$$\Upsilon = \{jklmnpqr | j, k, l, m, n, p, q, \text{and } r \in \mathbb{N}\},$$

where *j* identifies the cell type. The direction of motion is designated by *k*. When *k* is 0, the cell is in a stationary state. When the value ranges from $1 - 6$, it denotes one of the six directions in which the cell is currently moving. The digits lm identify the time until the next change in direction, called the *persistence counter*. The *cell phase counter* is denoted by $npqr$. The cell phase counter is the time remaining until the cell is ready to divide.

4.2 Mixed Cell Cultures and Tissue Architecture

Most tissues consist of several types of cells that organize themselves in very specific spatial patterns [11]. This three-dimensional architecture is what endows tissues with the special functions of organs. While previous studies have focused on a single type of cells, our model allows for simulating multiple cell types, with each cell type having its own migratory and proliferation characteristics. The model allows for the organization of cell populations into specific spatial patterns. Tissue engineers are attempting to achieve this goal by using microlithographic techniques to create surfaces with heterogeneous characteristics and solids with specific pore structure. When micro-patterned surfaces are used, different cells will migrate at different rates on different parts of the surface. Thus, we can guide cells of certain type to cluster in specified areas only. These areas may be surrounded by cells of another type.

5 Simulation Results and Discussion

5.1 Initial Conditions and Model Parameters

The simulations of the proliferation of multiple cell types are obtained using a $100 \times 100 \times 100$ cellular array where two cell populations are used. Cells of population 1 are faster moving cells while cells of population 2 are slower moving cells. We define R as the ratio of the initially seeded number of cells from these two cell populations. This is given by

$$R = \frac{\text{initial number of cells from population 1}}{\text{initial number of cells from population 2}}.$$

That is, when $R = 9$, there are 9 cells from population 1 for every cell from population 2.

We consider two seeding topologies: a uniform topology and a "wound healing" topology. In the former, the cells are randomly seeded in the cubic space. In the latter, an empty cylinder is surrounded by seeded cells in the remainder of the cellular space. The empty cylinder has a diameter of 50 and a height of 100. In addition, for each topology, we consider two seeding distributions:

1. *Mixed distribution*: The different cell types are seeded together according to the used cell distribution.
2. *Segmented distribution*: Each cell type is seeded in a separate area of the cellular space.

The migration speeds of cells in populations 1 and 2 are equal to 10 μm per hour and 1 μm per hour, respectively. Throughout these simulations, a confluence parameter of 100% and an average waiting time of 2 hours for the six directions of motion are utilized. Each cell is modeled as a cubic computational element whose sides are equal to 10 μm. We also use different division distributions for these two cell populations. The division times are given in Table 1.

Table 1. Division Time Distribution for the Two Cell Populations

	Cell Populations	
Division Times	Cell Population 1	Cell Population 2
12 - 18 hrs	64%	4%
18 - 24 hrs	32%	32%
24 - 30 hrs	4%	64%

The flexibility of the model permits the exploration of the influence of several system parameters on the growth rate and some other aspects of cell behavior such as the average cell speed. For instance, to compute the overall tissue growth rate, we use the following formula:

$$\frac{dk(t)}{dt} = \frac{\sum_{i=1}^{n} N_{c_i}(t) - N_{c_i}(t - \Delta t)}{\Delta t \times N_t},$$

where $k(t)$ is the cell volume fraction at time t as given in [12], Δt is the time step in days, N_t is the size of the cellular space (=$N_x \times N_y \times N_z$), $N_{c_i}(t)$ is the number of occupied computational sites by cell type i at time t, and n is the number of cell types ($n \geq 1$).

5.2 Uniform Topology

Effect of Seeding Density. Figure 1 shows the effect of the seeding density on the tissue growth rate. Here, cells from both populations are seeded in a mixed pattern using

a uniform distribution. The seeding density of population 2 is held constant at 0.1% while the seeding density of population 1 is varied from 0.1% to 10%. As expected, the time required to reach confluence decreases with the increase of the initial seeding density. We see that when the initial seeding density of population 1 is greater than 1%, the impact of cell division can be readily observed by the plateaus formed on the corresponding growth rate curves. The division time distributions of population 1 and population 2 specify that respectively 64% and 4% of cells will divide between 12-18 hours. This is depicted by the first plateau on the indicated curves for seeding densities of 5% and 10%. Additional plateaus are observed at later times reflecting the succeeding waves of cell divisions during the proliferation process. The subsequent waves of cell divisions include not only seeded cells giving birth to new cells but also daughter cells going through their own mitotic cycles. For small seeding densities, the growth rate curves show a more gradual increase that indicates a lower, though steady, overall impact on the tissue growth rate.

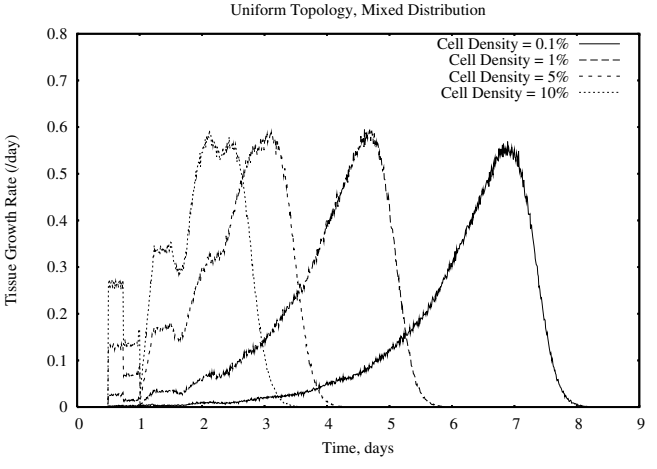
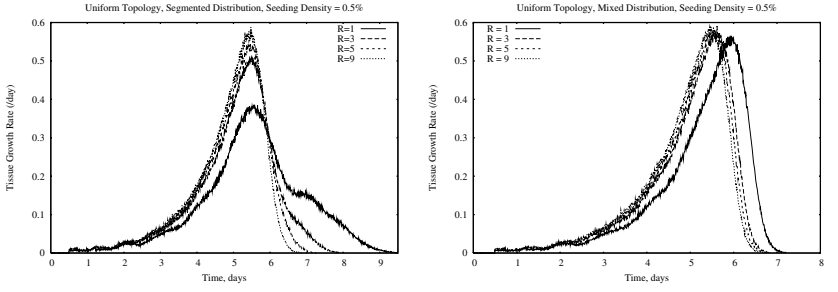


Fig. 1. Effects of varying the seeding density on the overall tissue growth rate. The seeding density of population 2 (slow-moving cells) is held constant at 0.1% while the seeding densities of fast-moving cells are varied.

Effect of Cell Heterogeneity. Figure 2(a) shows the tissue growth rate versus time for different values of the ratio R. In these simulation runs, R is equal to 1, 3, 5, and 9 while the total initial seeding density is maintained at 0.5%. For instance, with $R = 9$ the faster moving population of cells covers 90% of the cellular space while the slower moving population covers the remaining 10%. We observe that as R increases, the time taken to reach confluence decreases. This is because for larger values of R, the population of faster moving cells dominates the proliferation. This is clearly depicted in this figure where a higher tissue growth rate is observed as R is increased. Faster moving cells spread out in the cellular space preventing the formation of cell colonies; thus allowing for confluence to be reached sooner.

The effects of varying R on the tissue growth rate in a mixed distribution was also studied and is shown in Fig. 2(b). In these simulations, the total seeding density is 0.5%. We observe similar results to the ones obtained when using a segmented distribution. However, when $R = 1$, the mixed distribution yields a higher tissue growth rate (compare Fig. 2(a) with Fig. 2(b)). This may be attributed to the fact that contact inhibition has less of an effect in the mixed distribution where faster cells have more nearby empty spaces to move into, which in turn frees up sites for the slower-moving cells.



(a) (b)

Fig. 2. Overall tissue growth rate versus time for various values of ratio R in the case of two cell-seeding distributions

Effect of Cell-Seeding Distributions. Figure 3 show a comparison between the tissue growth rates obtained by using the segmented and mixed distributions of the uniform cell-seeding topology. Here, $R = 1$ and a seeding density of 0.1% are used. Cells from population 2 migrate at a fixed speed of 1 μm/hr. In both these figures, cells from population 1 have a migration speed of 10 μm/hr in part (a) and 50 μm/hr in part (b), respectively. We observe that the mixed-seeding distribution takes less time to reach confluence and yields higher tissue growth rates. However, as cell speeds increase, the difference between both seeding modes diminishes. Higher cell motility allows cells in the segmented distribution to move far away from their original seeded sites and disperse in the cellular space to minimize the impact of contact inhibition, thus creating a transient distribution that resembles the one induced by the mixed distribution. Moreover, with mixed seeding, the effects of increasing cell speed are less pronounced. For instance, the tissue growth rate curves are nearly identical for both population 1 speeds when a mixed distribution is used as in both cases confluence is reached in about 8 days and their growth rate curves are very similar. Increasing population 1 speeds is more beneficial in the case of a segmented distribution where an enhanced tissue growth rate and a reduction in the time to reach confluence are both observed.

5.3 Wound-Seeding Topology

Effect of Cell Speed. The growth rate curves in Fig. 4(a) and (b) demonstrate the effect of motility on volume coverage in the case of a wound-seeding mode using both

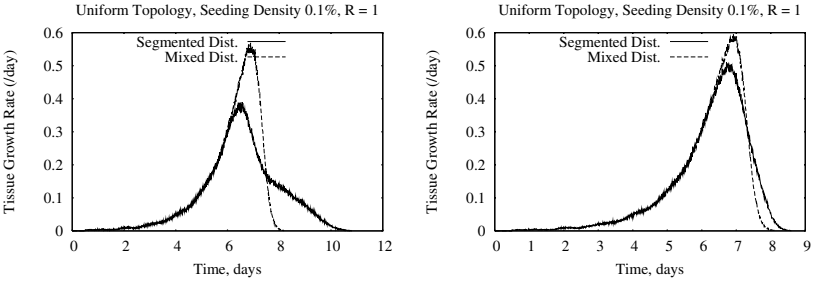


Fig. 3. Comparison of tissue growth rate for two cell-seeding distributions. Here, the migration speed of cells from population 2 is maintained at 1 μm/hr while cells from population 1 move at speeds of 10 μm/hr (part (a)) and 50 μm/hr (part (b)), respectively.

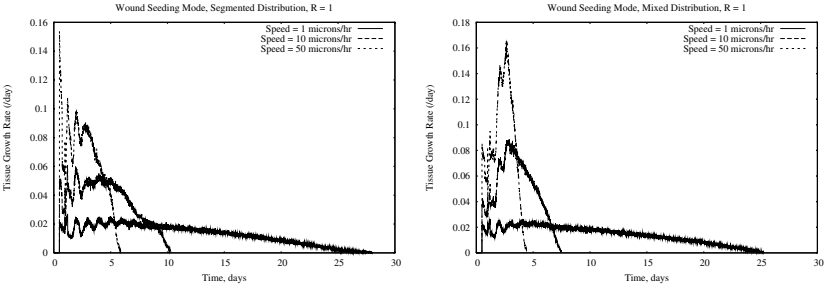


Fig. 4. Effect of varying the cell speed of population 1 on the overall tissue growth rate. Cells in population 2 move at a fixed speed of 1 μm/hr.

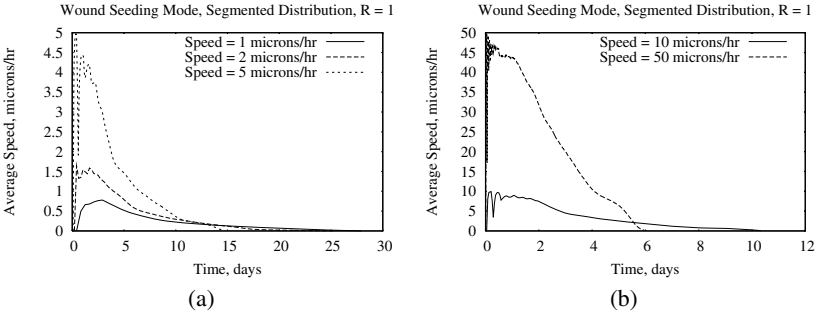


Fig. 5. The effect of cell motility (population 1) on the population-average speed of locomotion for cell speeds ranging from (a) 1 to 5 μm/hr and (b) 10 to 50 μm/hr

a segmented and mixed distribution with $R = 1$, respectively. We observe that as the motility of the cell increases, the tissue growth rate increases and confluence is attained faster; thus, healing the wound. Higher motility of cells decreases the impact of contact

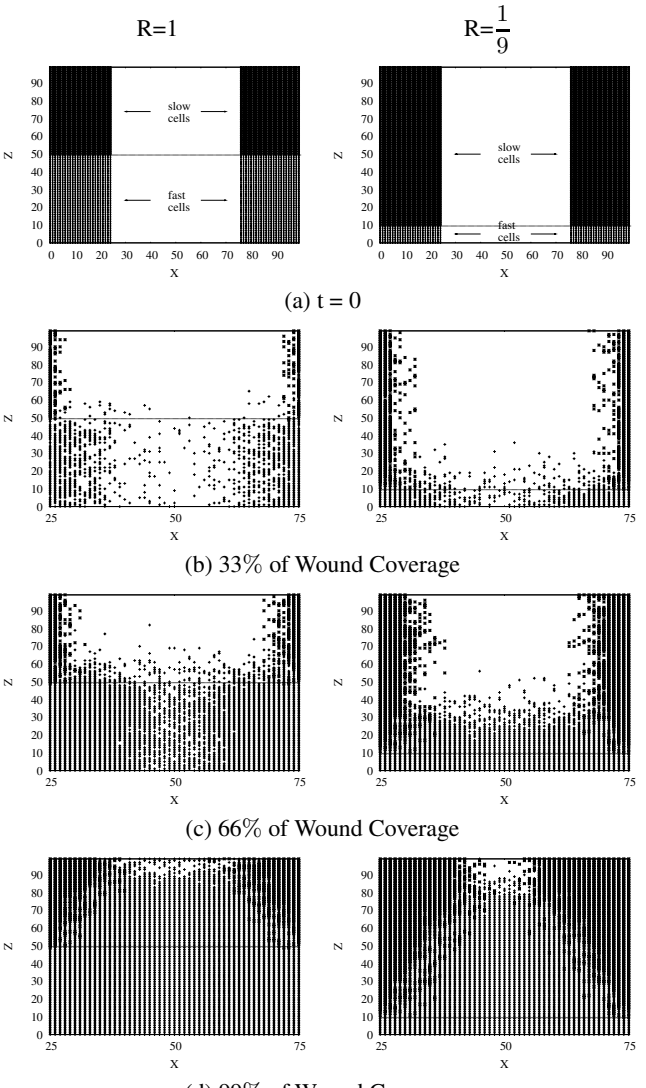
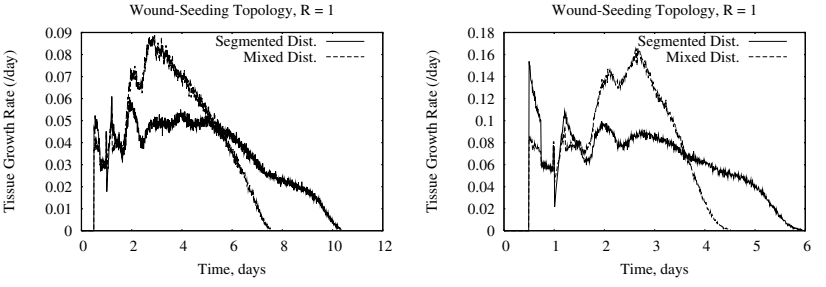


Fig. 6. Cell population profiles shown as 2-D cross sections of two simulation runs at t = 0 and at 33%, 66% and 99% of wound coverage for $R = 1$ and $R = \frac{1}{9}$, respectively. For illustration purposes, we included a horizontal line to distinguish between the two cell-seeding populations.

inhibition on the proliferation rate as it reduces the formation of cell colonies. Moreover, the tissue growth rate is the highest when cells in population 1 move at the fastest speed of 50 μm/hr. Highly motile cells cover the wound area faster, thus allowing them to reach far away, empty sites quickly and affording them additional opportunities for division. Figure 5 depicts the population-average speed in the denuded area (that is, the initially empty cylinder) for different speeds of population 1. At the beginning of the

simulations, cells move into the "wound" at their peak speeds. The overall cell speeds drop rapidly as the "wound" becomes congested with new daughter cells and collisions become more frequent. The average speed decreases with time and shows a drastic decrease as confluence is attained due to the formation of local clusters.



(a) (b)

Fig. 7. Comparison of tissue growth rate for two cell-seeding distributions. In both seeding distributions, the migration speed of cells from population 2 is maintained at 1 μm/hr while cells from population 1 move at speeds of 10 μm/hr (part (a)) and 50 μm/hr (part (b)), respectively.

Cell Population Profiles. Figure 6 shows the cell population profiles of two simulation runs using a segmented wound-seeding distribution and two different values of the ratio R ($R = 1$ and $\frac{1}{9}$, respectively). These profiles are exhibited as 2-D cross sections of the wound area and at different levels of wound coverage. We observe the following:

- At 33% of wound coverage, faster cells cover larger portions of the wound than slow cells as most of the growth appears in the lower partition of both seeding examples. There is also sporadic diffusion of fast cells into the upper partitions.
- At 66% of wound coverage, nearly most of the two lower partitions are covered while the upper partions remain mostly empty. In particular, when $R = \frac{1}{9}$, the lower partition is completely covered whereas the top segment is sparsely occupied by cells when $R = 1$.
- At 99% of wound coverage, the fast cells have penetrated to the top of the cellular space and have completely covered the denuded area at the bottom. Slow cells appear to mostly proliferate toward the center of the wound in a radial direction causing a cone-shaped funneling of the growth of fast cells.

Effect of Cell-Seeding Distributions. Figure 7 shows comparisons between the tissue growth rates obtained by using segmented and mixed distributions in case of wound-seeding mode and with $R = 1$. Cells from population 2 migrate at a fixed speed of 1 μm/hr. In this figure, cells from population 1 have a migration speed of 10 μm/hr in part (a) and 50 μm/hr in part (b), respectively. We observe that the mixed-seeding distribution takes less time to reach confluence and yields higher tissue growth rates. As cell speeds increase, the mixed seeding distribution continues to maintain its advantage resulting in $40 - 50\%$ reduction in the time to reach complete volume coverage, thus healing the wound faster.

6 Conclusion and Future Work

The reported research involved extending a three-dimensional model initially developed by the second author [7]. Our work incorporated the use of a variable number of mammalian cell populations. The simulation results included two populations of cells with different division and migration characteristics. Their effects on the tissue growth rate and population dynamics of cells were elucidated. The research will help speed up progress in the area of tissue engineering where the development of bio-artificial tissues involves extensive and time-consuming experimentation. It also extends the existing features of the model to incorporate others that more realistically simulate the process of tissue growth. Future work involves the parallelization of the model and its implementation on a Beowulf cluster using different decomposition techniques to simulate the growth of realizable tissue objects.

Acknowledgments

We gratefully acknowledge the financial support provided by the Natural Sciences and Engineering Research Council (NSERC) of Canada via a Discovery Grant under project number 31-611407.

References

1. Soll, D., Wessels, D.: Motion analysis of living cells: Techniques in modern biomedical microscopy. Wiley-Liss, New York, NY. (1998)
2. Zygourakis, K., Bizios, R., Markenscoff, P.: Proliferation of anchorage-dependent contact-inhibited cells: I. development of theoretical models based on cellular automata. Biotechnology and Bioengineering **38**(5) (1991) 459–470
3. Zygourakis, K., Markenscoff, P., Bizios, R.: Proliferation of anchorage-dependent contact-inhibited cells. II: Experimental results and validation of the theoretical models. Biotechnology and Bioengineering **38**(5) (1991) 471–479
4. Lee, Y., Markenscoff, P.A., McIntire, L.V., Zygourakis, K.: Characterization of endothelial cell locomotion using a markov chain model. Biochemistry and Cell Biology **73** (1995) 461–472
5. Cherry, R.S., Papoutaskis, E.T.: Modelling of contact-inhibited animal cell growth on flat surfaces and spheres. Biotechnology and Bioengineering **33** (January, 1989) 300–305
6. Lim, J.H.F., Davies, G.A.: A stochastic model to simulate the growth of anchorage dependent cells on flat surfaces. Biotechnology and Bioengineering **36** (September, 1990) 547–562
7. B. Ben Youssef: Simulation of cell population dynamics using 3-D cellular automata. In: Proceedings of the Sixth International Conference on Cellular Automata for Research and Industry (ACRI 2004). Volume 3305., Published by Springer Verlag in Lecture Notes of Computer Science (LNCS) (2004) 562–571
8. B. Ben Youssef, Markenscoff, P., Zygourakis, K.: Parallelization of a 3-D computational model for wound healing. WSEAS Transactions on Computers **3**(4) (2004) 993–998
9. Wolfram, S.: Cellular Automata and Complexity: Collected Papers. Addison-Wesley Publishing Co. (1994)

10. B. Ben Youssef: A three-dimensional stochastic model for tissue growth. In: Proceedings of the 16th IASTED International Conference on Modelling and Simulation (MS 2005). (2005) 136–142
11. Palsson, B.O., Bhatia, S.N.: Tissue engineering. Pearson Prentice Hall, Upper Saddle River, NJ. (2004)
12. Cheng, G., B. Ben Youssef, Markenscoff, P., Zygourakis, K.: Cell population dynamics modulate the rates of tissue growth processes. Biophysical Journal **90**(3) (2006) 713–724

Hierarchical Multi-resolution Finite Element Model for Soft Body Simulation

Matthieu Nesme[1,2], François Faure[1], and Yohan Payan[2]

[1] GRAVIR/IMAG-INRIA
[2] TIMC/IMAG
Grenoble, France

Abstract. The complexity of most surgical models has not allowed interactive simulations on standard computers. We propose a new framework to finely control the resolution of the models. This allows us to dynamically concentrate the computational force where it is most needed.
Given the segmented scan of an object to simulate, we first compute a bounding box and then recursively subdivide it where needed. The cells of this octree structure are labelled with mechanical properties based on material parameters and fill rate. An efficient physical simulation is then performed using hierarchical hexaedral finite elements. The object surface can be used for rendering and to apply boundary conditions.
Compared with traditional finite element approaches, our method dramatically simplifies the task of volume meshing in order to facilitate the using of patient specific models, and increases the propagation of the deformations.

1 Introduction

1.1 Context

Soft body simulation is a growing research domain, for communities such as Computer Aided Surgery, Virtual Reality or Computer Graphics. Computer aided surgery (CAS) aims at assisting surgeons for the realization of diagnostic and therapeutic gestures in a rational and quantitative way in order to increase safety and accuracy [1]. While the first designed systems focused on orthopaedics, researchers addressed more recently anatomical structures that cannot be considered as "rigid" as they are mainly composed of biological soft tissues. The corresponding CAS systems therefore need to take into account the displacements of the structures as well as their deformations. In most cases, authors propose to build biomechanical models of the anatomical structures and use these models to predict, in the most accurate way, the tissue deformations induced by the surgical gesture.

Virtual Reality (VR), in its interactions with the Medical community, has recently provided surgical simulation systems (Cotin et al., 1996). As for the flight simulators used to train pilots, the idea is that these surgical VR systems could be a great help in the learning and training processes, allowing the surgeon to acquire, for example, some difficult hand-eye coordinations, to repeat several times the most difficult gestures or to choose the best surgical procedure for a given pathological case. As for the CAS systems, deformable models have been included into the simulators, with constraints in term of robustness and computation times.

M. Harders and G. Székely (Eds.): ISBMS 2006, LNCS 4072, pp. 40–47, 2006.

Computer Graphics (CG) has developped methods for the visually plausible animation of complex physical objects such as clothes and hairs. Considerable speedups have been obtained for stiff flexible bodies using implicit time integration, which allows arbitrary time steps to be performed. Significant advances have also been made using hierarchical modeling and the control of levels of detail.

These three communities now converge towards the same needs, in terms of soft body modelling: accuracy, robustness and interactivity (*i.e.* fast computation times). Indeed, from one side the CAS community is now looking for models that could be peroperatively used, with possible real-time re-planning of the surgical gesture. From the other side, the VR and CG communities now focus on the accuracy of the deformations, in order to be as realistic as possible, in comparison with real data.

In this framework, some recent works, coming from these communities, try to provide mechanical models with innovative implementations that preserve a continuous modelling context (with, for most of the works, a numerical resolution through the Finite Element Method) while proposing improvements, in terms of robustness and computation times. In addition, models built in the CAS or VR contexts need to be adapted to each patient anatomy. This point is particularly challenging (and time consuming) when a patient-specific Finite Element mesh needs to be defined.

Next part tries to summarize all of these recent works (part 1.2), while a new modelling approach is introduced in part 2. An example of implementation is presented in part 3 before providing some results (part 4).

1.2 Related Work

In order to improve the computational efficiency of continuous biomechanical models, researchers have proposed new approaches concerning (1) the Finite Element discretization, (2) the dynamical integration and (3) the numerical resolution methods.

Because of the need for speed, the first interactive methods were based on precomputed matrix inversion [2]. To extend these methods to large deformations frameworks, a non-linear computation of the strains is used in [3,4]. Recently proposed methods favor a new approach based on the decomposition of the displacement of each element into a rigid motion and a pure deformation tractable linearly in the local frame [5,6,7,8]. These methods allow a large displacements and rotations framework.

In the animation community, implicit integration methods have become popular, thanks to the iterative solution based on conjugate gradient presented by [9]. Although these methods permit large time step, they become ”expensive” when a fast propagation of the deformation is suited, since they require a lot of iterations to solve the system accurately. On the contrary, explicit integrations do not use iterations but require small time steps to maintain stability. Therefore, if a fast propagation of the deformations is needed, both implicit and explicit integration schemes remain computationally expensive. To face this problem, hierarchical methods have been proposed, providing an improvement of the propagation of the deformations (see for example the hierarchical solvers proposed by [10,11]).

In order to adapt the numerical solution schemes to the adequate level of details, authors have proposed to adapt the Finite Element (FE) mesh according to the actual state of the model (in terms of displacement, strain or stress). They propose therefore a

multi-resolution FE approach ([12,3,13]). The idea is for example to define, for a given anatomical structure, different FE meshes, from a very coarse one to a full refined one. If boundary conditions induce small deformations inside the structure, the coarse mesh is sufficient for providing accurate FE discretization. On the contrary, a dense mesh is used where the deformation is high. [14] shows how to link hexahedral elements of different sizes but the octree hierarchy is not used in the dynamics computations.

2 Contribution

Our approach proposes to merge a multi-resolution description with a Hierarchical FE integration. It is supposed to provide a numerical scheme that can be used for any type of mechanical description, from a small deformation framework to hyperelasticity. The objective is only to gather some methods already proposed in the literature in order to improve the propagation of the deformations as well as the efficiency of the computation according to the mechanical and geometrical state of the soft body. The specificity of the method is that a global 3D mesh is defined from a classical octree division of a bounding box including the soft body. Therefore, no FE mesh is needed to specificaly model the 2D or 3D geometry of the body. Indeed, the FE computations are applied to the 3D mesh defined by the octree. This octree can be directly built from the volumic data resulting of the segmented patient scan, so it is very efficient when we have few time to built a patient specific mesh, for exemple when the patient comes in emergency. To improve the propagation of the deformations, a hierarchical basis is defined to interpolate the FE computations, from the global parent cell defined by the bounding box to each child cell of the octree. A real difficulty is to convincingly animate stiff materials in an interactive context, when only an approximate solution can be computed. Our hierarchical approach gives more realistic results in this case. Moreover, the multi-resolution scheme is used to decide, for a given state of the body, which levels of the octree should serve as basis for (1) FE computation and (2) 3D rendering: for example, only regions with high strains level should use a dense octree level for FE computation, while regions that are not displayed on the screen because they are not seen by the camera should use a coarse octree mesh level for the 3D rendering.

2.1 Octree Mesh and Multi-resolution

The first step consists in defining the complete 3D octree mesh as presented in figure 1-a. Starting from a cubic bounding box of the body, an iterative algorithm is used to divide each "parent" cube in order to generate eight "child" cubes. The cubes that do not contain any part of the body are removed from the octree mesh. The remaining cubes are again divided, so that each of them will generate 8 new cubes. A maximal level of division N_{max} is defined once, leading to the "maximal density" octree mesh (figure 1-a). Using this octree mesh architecture as a baseline, two intermediate resolutions will be defined at each time step of the global computation of the system, namely the octree resolution N_{FE} used for the FE mesh interpolation (figure 1-b) and the octree resolution N_{Rend} used for the rendering display (figure 1-c). These two resolutions can change

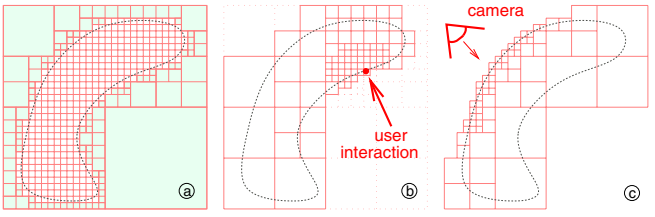


Fig. 1. An exemple of adaptativity. (a) leaves of the octree mesh = the finest level of details. (b) mechanical leaves = the finest mechanical level. (c) geometric leaves = the finest geometric level.

during the solving of the system, according to the changes in the boundary conditions as well as the location of the camera that looks at the scene. For example, one condition to define whether a given level of the octree mesh is suitable for a given point of the body consists in looking at the strain rate. If it is sufficiently low, this means that the current resolution is sufficient. On the contrary, if a high strain variation is observed, a denser mesh is preferable around this region of the body leading to the use of the child cells of the actual octree element. Once the N_{FE} level is reached, the corresponding octree 3D mesh is used for the FE computation. In order to limit the influence of cells that would contain a small amount of the body (cells located at the surface of the body), it is proposed to ponderate the rheology of these cells by their filling ratio as illustrated in figure 4.

2.2 Hierarchical FE Bases

A function decomposed in a hierarchical basis is modeled using a rough approximation based on a few broad-range sample points, along with a number of recursively narrower-range sample points encoding local detail added to the approximation. Each value of the function is thus the sum of shape functions with various radius of influence. This approach allows one to easily control the level of detail by simply inserting or dropping control values where desired [15]. Another nice feature of this approach is to considerably speed up the convergence of shape optimization, as shown in geometric modeling [16]. It has been successfully applied to finite element methods [17,18].

In the case of the octree mesh introduced above, the position stored for each vertex is relative from its parents position. Therefore, only vertices from the root cell (*i.e.* the cubic bounding box that includes the body) have real position in the 3D space. At start, all others child cells have a null relative position which only depends on their parents. Figure 2 illustrates the FE interpolation that will be provided with hierarchical linear functions.

3 An Implementation

This section proposes an example of the implementation of the previously presented hierarchical approach. This implementation uses the Cauchy's deformation tensor with a co-rotational handeling of large displacements [5,6,7,8] and a viscoelastic material.

3.1 Large Displacements

To handle rotational displacements, each detail (*i.e.* hierarchical value) has to be represented in a local frame that follows the rotation as showed in figure 3. In this way, a local frame is attached at each cell, and each node is defined in the local frame of a cell. The corresponding cell of a node is determined in taking the finnest cell that contains all the incident cells to the node as represented in the figure 3-right. To compute cell rotations, the eigenvectors are used as explain in [5] by decomposing the matrix formed with the egdes averages in the three directions.

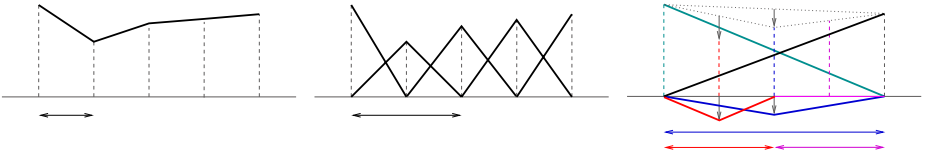


Fig. 2. Illustrations of the finite element (top-left), basis function (top-right) and hierarchical basis function (bottom) points of view with linear interpolation. In the basis function point of view, the solution corresponds to a combinaison of functions associated with the nodes. In the hierarchical point of view, the influence support of these functions varies.

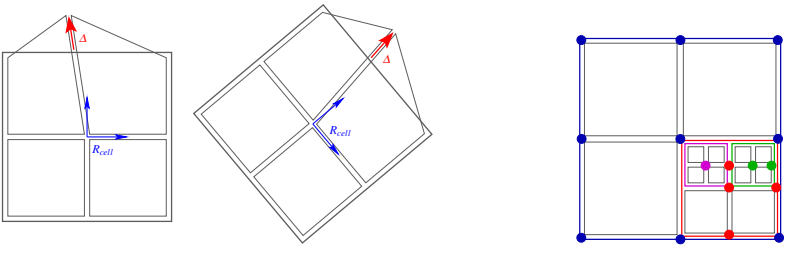


Fig. 3. By taking into account the local frame of the cell, the hierarchical values are invariant in rotation

3.2 Mechanics

The standard method used to simulate viscoelastic solids is considered, equations and notations can be found in [19].

Our approach induces differences with classical formulation concerning the displacement u which is not defined in global space coordinates, but is defined hierarchically. Only displacements of the vertices of the root cell are in space coordinates. The displacement of others vertices is relative from their parents. To build the mass matrix $\mathbf{M}$ and the stiffness matrix $\mathbf{K}$, not only finest elements are considered. Indeed, for each elements along the hierarchy we take into account all nodal functions that influence the considered element as exprimed in algorithms 1 and 2. $\mathbf{C}$ is the stress-strain matrix relating the material properties and the strain-displacement matrix $\mathbf{B}_i$ is obtained by differentiation of interpolation functions $\mathbf{h}_i$ with respect to natural coordinate and

premultiplying the result by the inverse of the Jacobian operator. $\mathbf{H}_i$ is a matrix of the interpolation functions $\mathbf{h}_i$.

Algorithm 1. BUILD MATRICES **K** AND **M**

for each *cell* **do**
  for each vertex i defined at level of *cell* **do**
    INTEGRATE($\mathbf{B}_i, \mathbf{B}_i, \mathbf{H}_i, \mathbf{H}_i, \mathbf{C}_{cell}, \mathbf{J}_{cell}$)
    for each vertex $j \neq i$ defined at level of *cell* **do**
      INTEGRATE($\mathbf{B}_i, \mathbf{B}_j, \mathbf{H}_i, \mathbf{H}_j, \mathbf{C}_{cell}, \mathbf{J}_{cell}$)
    end for
    for each *ancestor* of *cell* **do**
      for each vertex j defined at level of *ancestor* **do**
        take function h_j between range of *cell* in *ancestor* // *detail in section* **??**
        INTEGRATE($\mathbf{B}_i, \mathbf{B}_j, \mathbf{H}_i, \mathbf{H}_j, \mathbf{C}_{cell}, \mathbf{J}_{cell}$)
        INTEGRATE($\mathbf{B}_j, \mathbf{B}_i, \mathbf{H}_j, \mathbf{H}_i, \mathbf{C}_{cell}, \mathbf{J}_{cell}$)
      end for
    end for
  end for
end for
// *Note that some computations can be omitted in considering the symetric aspect of matrices* $\boldsymbol{K}$ *and* $\boldsymbol{M}$: $\boldsymbol{K}_{i,j} = \boldsymbol{K}_{j,i}^T$ *and* $\boldsymbol{M}_{i,j} = \boldsymbol{M}_{j,i}^T$

Algorithm 2. INTEGRATE($\mathbf{B}_i, \mathbf{B}_j, \mathbf{H}_i, \mathbf{H}_j, \mathbf{C}, \mathbf{J}$)

$$\mathbf{K}_{i,j} = \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} \mathbf{B}_i^T \mathbf{C} \mathbf{B}_j \, det\mathbf{J} \, dr \, ds \, dt$$

$$\mathbf{M}_{i,j} = \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} \mathbf{H}_i^T \mathbf{H}_j \, det\mathbf{J} \, dr \, ds \, dt$$

4 Results

Figure 4 shows an octree mesh for a liver model.

Table 1 compares the number of iterations necessary to converge to the equilibrium with a static solver using the nodal approach (*i.e.* the classical non-hierarchical one) against the hierarchical approach. Two examples are considered for several numbers of elements, the first one consists in a cubic fixed beam subject to gravity, while in the second a force is applied to a corner of the beam. As expected, the convergence is faster using the hierarchical approach. When the corner is pulled, the other end moves directly, whereas in the nodal model it is necessary to propagate the deformation along all elements.

This faster propagation is useful in case of real-time simulation when only few iterations can be performed on the implicit integration at each step. Using hierarchy, a small number of iterations (approximately ten) provides a much more accurate result, as illustrated in figure 5 that plots the convergence speed of the second example of table 1. In case of very soft materials, fast propagation can be unrealistic. In this case a classical nodal approach can be better suitable.

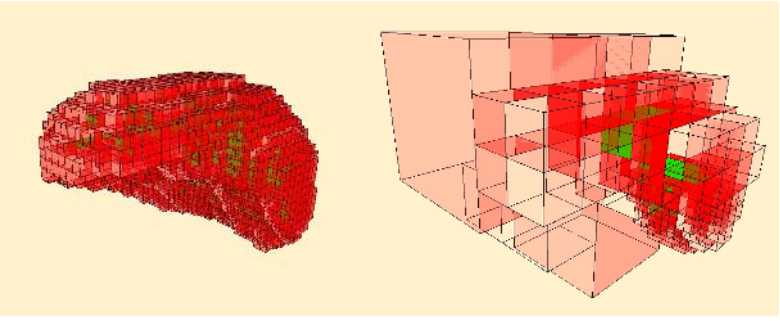


Fig. 4. An octree-mesh for a liver: densities of mechanical leaves for the finnest level of details and for a multiresolution mesh

Table 1. Number of CG iterations of the static solver until convergence, on two examples on a cubical fixed beam

Number of elements		1	8	64	512
example 1 (gravity)	nodal	1	13	55	146
	hierarchical	1	11	27	47
example 2 (boundary force)	nodal	8	50	87	198
	hierarchical	6	24	37	52

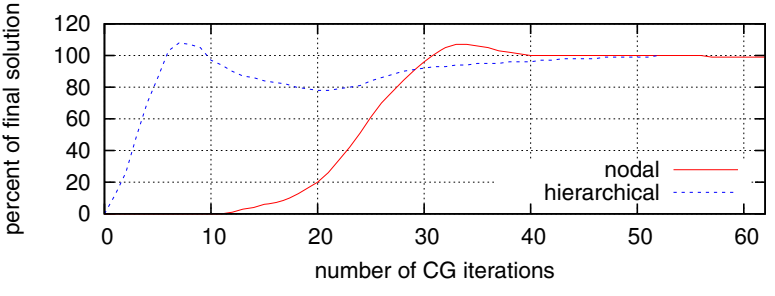


Fig. 5. Convergence speed in a static solver

5 Conclusion

We proposed in this paper a hierarchical multiresolution technique to animate soft bobies. This new approach based on a octree mesh permits to work on various geometrical representations of an object without needing to provide a volumetric mesh of this object. Using the hierarchical approach improves the propagation and permits to simulate more rigid materials. Despite it, we do not yet obtain better results in terms of computation time because an optimized structure is difficult to set up. To make this work usable, it will be necessary to integrate criteria of adaptivity (automatic definition of N_{FE} and N_{Rend} values), and to take into account effective boundary conditions.

References

1. Taylor, R., Lavallée, S., Burdea, G., Mosges, R.: Computer integrated surgery: Technology and clinical applications, Cambridge, MA: MIT Press (1996)
2. Cotin, S., Delingette, H., Clement, J.M., Tassetti, V., Marescaux, J., Ayache, N.: Volumetric deformable models for simulation of laparoscopic surgery. In: Computer Assisted Radiology. (1996)
3. Debunne, G., Desbrun, M., Cani, M.P., Barr, A.H.: Dynamic real-time deformations using space and time adaptive sampling. In: SIGGRAPH '01. (2001)
4. Picinbono, G., Delingette, H., Ayache, N.: Non-linear anisotropic elasticity for real-time surgery simulation. Graph. Models (2003)
5. Etzmuß, O., Keckeisen, M., Straßer, W.: A Fast Finite Element Solution for Cloth Modelling. Proc Pacific Graphics (2003)
6. Hauth, M., Straßer, W.: Corotational simulation of deformable solids. In: Proc WSCG. (2004)
7. Müller, M., Gross, M.: Interactive virtual materials. In: Proc Graphics Interface. (2004)
8. Nesme, M., Payan, Y., Faure, F.: Efficient, physically plausible finite elements. In: Eurographics (short papers). (2005) 77–80
9. Baraff, D., Witkin, A.: Large steps in cloth simulation. In: SIGGRAPH '98. (1998)
10. Terzopoulos, D., Fleischer, K.: Modeling inelastic deformation: viscolelasticity, plasticity, fracture. In: SIGGRAPH '88. (1988)
11. Wu, X., Tendick, F.: Multigrid integration for interactive deformable body simulation. In: ISMS. (2004) 92–104
12. Debunne, G., Desbrun, M., Barr, A.H., Cani, M.P.: Interactive multiresolution animation of deformable models. In: Eurographics Workshop on Computer Animation and Simulation. (1999)
13. Wu, X., Downes, M.S., Goktekin, T., Tendick, F.: Adaptive nonlinear finite elements for deformable body simulation using dynamic progressive meshe. In: EG 2001 Proceedings. (2001)
14. G.P.Nikishkov: Finite element algorithm with adaptive quadtree-octree mesh refinement. In: ANZIAM J. Volume 46(E). (2005) C15–C28
15. Stollnitz, E.J., DeRose, T.D., Salesin, D.H.: Wavelets for Computer Graphics: Theory and Applications. Morgan Kaufmann Publishers, Inc. (1996)
16. Gortler, S.J., Cohen, M.F.: Hierarchical and variational geometric modeling with wavelets. In: SI3D '95. (1995)
17. Grinspun, E., Krysl, P., Schröder, P.: Charms: a simple framework for adaptive simulation. In: SIGGRAPH '02. (2002)
18. Capell, S., Green, S., Curless, B., Duchamp, T., Popović, Z.: A multiresolution framework for dynamic deformations. In: SCA '02. (2002)
19. Bathe, K.J.: Finite Element Procedures in Engeneering Analysis. Prentice-Hall, Inc. (1982)

Simulation of the Retroglossal Fluid-Structure Interaction During Obstructive Sleep Apnea

Franz Chouly[1,2], Annemie Van Hirtum[2], Pierre-Yves Lagrée[3], Jean-Roch Paoli[4], Xavier Pelorson[2], and Yohan Payan[1]

[1] Laboratoire TIMC, UMR CNRS 5525, Université Joseph Fourier, 38706 La Tronche Cedex, France
{Franz.Chouly, Yohan.Payan}@imag.fr
[2] Institut de la Communication Parlée, INPG, UMR CNRS Q5009, 38031 Grenoble Cedex, France
{annemie, pelorson}@icp.inpg.fr
[3] Laboratoire de Modélisation en Mécanique, UMR CNRS 7607, B 162, Université Paris 6, 75252 Paris, France
pyl@ccr.jussieu.fr
[4] Centre Hospitalier Universitaire Purpan, Place du Docteur Baylac, TSA 40031, 31059 Toulouse Cedex 9, France
paoli.jr@chu-toulouse.fr

Abstract. A method for computing the interaction between the airflow and the soft tissue during an Obstructive Apnea is presented. It is based on simplifications of the full continuum formulation (Navier-Stokes and finite elasticity) to ensure computation time compatible with clinical applications. Linear elasticity combined with a precomputation method allows fast prediction of the tissue deformation, while an asymptotic formulation of the full Navier-Stokes equations (Reduced Navier-Stokes/Prandtl equations) has been chosen for the flow. The accuracy of the method has already been assessed experimentally. Then, simulations of the complete collapsus at the retroglossal level in the upper airway have been carried out, on geometries extracted from pre-operative radiographies of two apneic patients. Post-operative geometries have been also used to check qualitatively if the predictions from the simulations are in agreement with the effects of the surgery.

1 Introduction

The Obstructive Sleep Apnea Syndrome is now identified as a major health care topic, which affects a growing part of the population in the western countries [1]. At least, 2 % of the women and 4 % of the men suffer from this disorder [2]. Its main consequences are excessive daytime sleepiness and an increased risk of cardiovascular diseases [1]. Fundamentally, it is caused by the interaction between the respiratory airflow and the soft tissue in the pharynx [3]. Physical modelling of this complex phenomenon is of interest both for going deeper into the understanding of its mechanisms and for further improvements of the treatments. In particular, realistic numerical simulations of the interaction between the airflow

M. Harders and G. Székely (Eds.): ISBMS 2006, LNCS 4072, pp. 48–57, 2006.

and the pharyngeal structures during Obstructive Sleep Apnea, on geometries obtained from medical imaging, might be helpful for the clinician or the surgeon, who attempt at modifying the geometry or the mechanical properties of the upper airway. In this perspective, two prerequisites should be considered. The first is that the simulations should be validated experimentally, so that the clinician may trust their predictions. The second is that computational cost in a clinical context should be low. Indeed, for this problem, the best method would be to solve accurately the full unsteady Navier-Stokes equations in interaction with soft tissue governed by non-linear viscoelastic behaviour, in large deformations, using tridimensional data such as CT-scans or MRI. However, this complete formulation is very difficult to solve numerically (convergence problems on real in-vivo geometries) and requieres hours if not days of computations [4] [5]. A reasonable objective for clinical applications should be a simulation time of less than one hour, or even of a few minutes, so that many simulations, involving different modifications of the morphology or of the tissue properties, should be carried out one after the other. Therefore, the objective of this paper is to present numerical simulations that meet these two prerequisites, which has never been done before for Obstructive Sleep Apnea[1]. The method, as well as the framework for simulations, is first explained in section 2. In particular, the simplifications in the assumptions and in the numerical solving, in view of the second prerequisite, are given. For the first prerequisite, systematic comparison to in-vitro measurements has already been proceeded and is detailed in [8] [9] [10], so we focus here on simulations from in-vivo data. Therefore, in section 3, the simulation of an apnea is detailed, followed by confrontation between simulations from pre-operative and post-operative data, which will provide a preliminary qualitative in-vivo evaluation of the method.

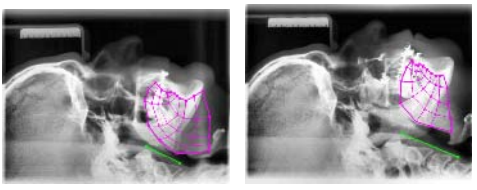
2 Material and Methods

The construction of a patient geometry from a radiography is first described, followed by the method of computation for simulations. Finally, the choice of values for physical parameters and the estimation of the reference configuration after the application of gravity are detailed.

2.1 Extraction of the Geometry from Sagittal Radiographies

The upper airway geometry of two apneic patients was obtained from sagittal radiographies. These two patients undergone maxillomandibular surgery: their maxilla and mandible were translated forward in the antero-posterior direction,

[1] Most of the simulations for biomechanical processes at the macroscopic level involving fluid-structure interaction are devoted to blood flows or to the pulmonary airway. To our knowledge, the only published numerical simulations for Obstructive Sleep Apnea are those of B. Shome and coll. [6], focused on the fluid flow (no interaction with the soft tissue), and of A. Malhotra and coll., in which computation time is not indicated and which have not been validated experimentally [7].



(PreOp) (PostOp)

Fig. 1. Segmentation of the pre-operative (*PreOp*) and the post-operative (*PostOp*) radiographies of apneic patient no. 1. The initial mesh and the borders of the tongue are displayed in purple, while the posterior pharyngeal wall is in green.

of approximatively 8 to 10 mm. Two acquisitions took place for each of them: one before and one after the surgical intervention, at approximatively one month of interval. The main interest of the radiographies is to dispose of the post-operative data, since in the normal surgical protocol, no MRI or CT-scans are taken after the intervention. Table 1 provides patients clinical information. In particular, the Apnea-Hypopnea Index (AHI) drops dramatically after the intervention, which shows the efficiency of the surgical gesture.

From each radiography, a bidimensional model of the tongue has been extracted. First, landmarks have been manually positioned in order to determine the border of the tongue. From these landmarks, a mesh has been generated. It is a structured, isoparametric mesh, as in [11], made of 216 quadrilateral linear elements. The posterior pharyngeal wall has been considered in a first approximation as a straight segment. The airway is naturally delimited by the base of the tongue and the posterior pharyngeal wall. An example of segmentation is depicted in figure 1. For the two patients considered in this paper, segmentations have been validated by a clinical expert.

2.2 Computation of the Fluid-Structure Interaction

As pointed out in the introduction, since our objective is a low computational cost, some simplifications in the formulation of the problem have been adopted. First of all, in agreement with the clinical data at our disposal, a bidimensional formulation has been chosen, both for the soft tissue and for the airflow. Moreover, the problem is considered as quasi-steady, since the duration of the closure lasts about a few seconds. This allows to solve the fluid-structure interaction using a segregative method: the equations that govern the fluid and the structure are solved alternatively [12]. For the soft tissue, small deformations and small displacements assumptions have been thought as suitable. This is different from a context such as speech production where large deformations are involved [13]. The material has been chosen as homogeneous, isotropic and linear (Hookean material). The resulting equations are solved using the finite element method [14]. Within this framework, the relationship between the nodal displacements $\{\mathbf{u}\}$ and the nodal forces $\{\mathbf{F}\}$ is linear:

$$[\mathbf{K}]\{\mathbf{u}\} = \{\mathbf{F}\}, \tag{1}$$

where [**K**] is the stiffness matrix [14]. Then, a method of precomputation, similar to the one detailed in [15], has been used. It consists in computing the invert of [**K**] before entering into the fluid-structure interaction loop. Ansys(TM) software was used at this effect. As the displacement at each node is simply obtained from a matrix multiplication at each step of the fluid-structure interaction loop, the precomputation of the invert of [**K**] saves a great amount of computation time. The airflow has been considered as :

- *incompressible* : the Mach number is of the order of 10^{-2} [8],
- *laminar* : the Reynolds number is of the order of 10^3 [8]. We neglected the turbulent effects due to shear instability in the jet that appears downstream the base of the tongue. In fact, the flow may be transitional, but, as shown in the experimental study [8], laminar regime remains a satisfying first approximation.
- *stationnary* : the Strouhal number is of the order of 10^{-3} [8]. Moreover, the displacement speed of the soft tissue is very slow in comparison to the characteristic time of the flow, so that unstationnary effects should be neglected in the fluid[2].

Working at Reynolds number of the order of 10^3 allows to use an asymptotical simplification of the incompressible Navier-Stokes equations:

$$\bar{u}\partial_{\bar{x}}\bar{u} + \bar{v}\partial_{\bar{y}}\bar{u} = -\partial_{\bar{x}}\bar{p} + \partial^2_{\bar{y}^2}\bar{u}, \tag{2}$$

$$-\partial_{\bar{y}}\bar{p} = 0, \tag{3}$$

$$\partial_{\bar{x}}\bar{u} + \partial_{\bar{y}}\bar{v} = 0, \tag{4}$$

where $(\bar{u}, \bar{v})$ are the longitudinal $(\bar{x})$ and transverse $(\bar{y})$ components of the fluid velocity, and $\bar{p}$ is the pressure [16]. All the variables are nondimensional: $\bar{x} = x(h_0 Re)^{-1}$, $\bar{y} = yh_0^{-1}$, $\bar{p} = P(\rho U_0^2)^{-1}$, $\bar{u} = uU_0^{-1}$, $\bar{v} = vReU_0^{-1}$. h_0 is the transversal dimension of the pharyngeal duct, U_0 is the mean longitudinal speed, and Re is the Reynolds number ($Re = U_0 h_0/\nu$, with ν the kinematic viscosity of the air). This simplification, called Reduced Navier-Stokes / Prandtl (RNSP) equations, allows to take into account the boundary layer formation and the separation of the fluid after the narrowing of the pharyngeal duct, at the base of the tongue. Furthermore, it allows fast and efficient numerical solving, based on a finite difference scheme [16]. For more details about the assumptions, the mathematical formulation and the numerical solving, one can refer to [9] [10]. Finally, the method has been extensively validated thanks to comparison with pressure and deformation measurements on an in-vitro setup. For the computation of the fluid flow, the above assumptions and the RNSP equations have been validated on a rigid setup [8]. For the fluid-structure interaction, a deformable setup has been used [9] [10]. As a result, the first prerequisite is satisfied. For the second prerequisite, the duration of the computations is typically of the order of 20 minutes (see 4.1), which can be considered as correct for clinical applications.

[2] In opposition to applications such as snoring or speech production in which it would not be the case.

Table 1. Information on each patient, after pre-operative ('PreOp') and post-operative ('PostOp') examination. 'AHI' is the Apnea-Hypopnea Index, which is the number of obstructive events per hour of sleep, measured during polysomnography. h_c^i is the constriction height in vertical position, awake, as it is measured from the sagittal radiographies. h_c^f is the constriction height in a sleep supine position, after the application of the gravity loads.

Patient	1		2	
Examination	PreOp	PostOp	PreOp	PostOp
AHI	80	8	49	0
h_c^i (mm)	5.9	14.1	6.6	16.8
h_c^f (mm)	1.5	10.5	1.8	12.9

2.3 Framework for the Simulations

For each bidimensional model, the plane strain assumption has been chosen. The depth[3] has been fixed to 30 mm. It corresponds to a mean value of the depth of the oropharyngeal airway. The Poisson's ratio has been fixed to 0.499. It stands for quasi-incompressibility, in agreement with the fact that the soft tissues are mostly made of water. The same value for the Young modulus has been used for all the simulations: 6 kPa. It corresponds to tissue in passive configuration [7]. The boundary conditions are the following: the tongue is attached to the mandibulae and to the hyoid bone, which have been considered as immobile, in agreement with remarks from the clinical expert. Then, before the simulations of the interaction with the airflow, the influence of the gravity has been simulated. Indeed, as the radiographies were taken in vertical position, during wakefulness, a new configuration corresponding to sleep in supine position has to be computed. The norm of the gravity field $\mathbf{g}$ has been fixed to 9.81 m.s^{-2}. The density ρ of the tissue has been fixed to 1000 kg.m^{-3}. The configuration obtained after the application of gravity loads has been considered as the reference configuration for all the computations described in the next section. Table 1 gives some information about the deformation induced by gravity, especially the minimal height of the duct, at the extremity of the base of the tongue. It has been called the constriction height h_c.

3 Results

Simulations have been carried out on the oropharynx models built from the sagittal radiographies, before and after surgery. Comparison of the tongue deformation in response to inspiratory airflow allows us to check if the predictions from the simulations are in agreement with the consequences of the surgical intervention. In order to generate an inspiratory pattern, the inlet pressure has been fixed to 0 Pa while the outlet pressure decreases from 0 Pa to a negative chosen value P_s Pa.

[3] Or the dimension in the direction orthogonal to the sagittal plane.

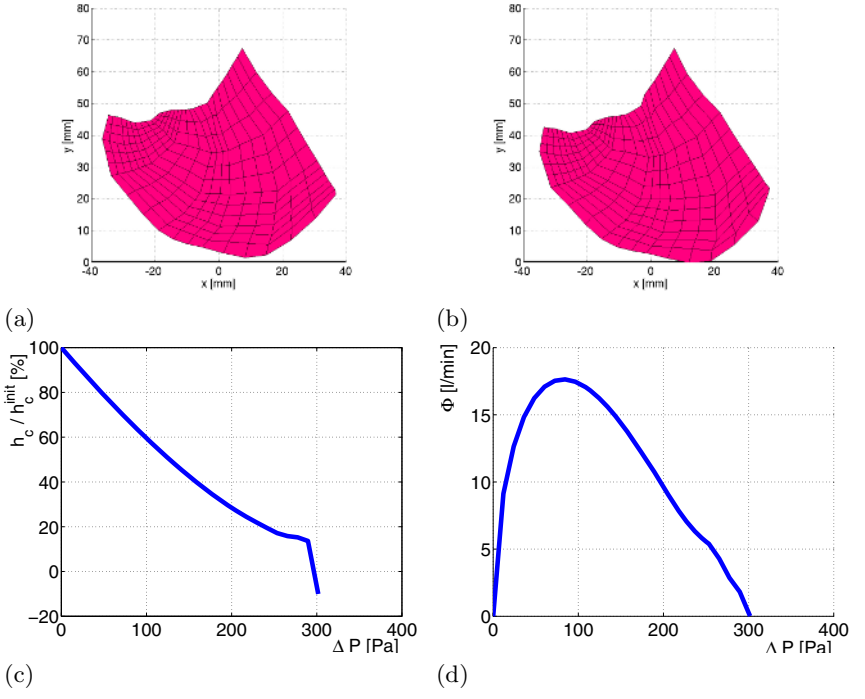


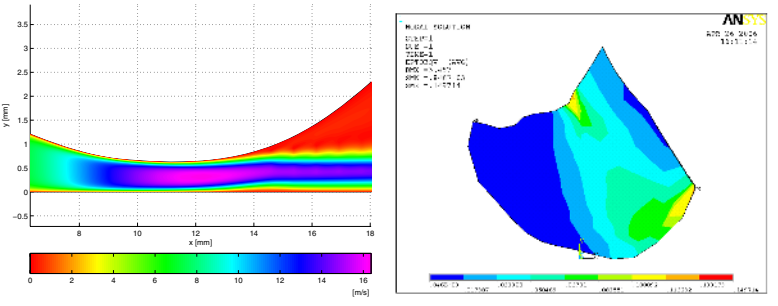
Fig. 2. Simulation of the upper airway obstruction from pre-operative data. (a) Initial configuration. (b) Final configuration. (c) Constriction height. (d) Airflow rate ϕ.

3.1 Simulation of an Obstructive Apnea

On figure 2 is depicted the simulation of a complete upper airway collapsus. The geometry is extracted from the pre-operative data of the patient no. 1. The constriction height h_c decreases from its initial value to 0 mm (figure (c)), at which collision with the posterior pharyngeal wall is detected (figure (b)). In parallel, the airflow rate ϕ increases with the pressure drop ΔP until approximatively 80 Pa, where the maximal value of 17.5 l/min is reached (figure (d)). After 80 Pa, the flow rate ϕ decreases as the pressure drop ΔP is increased. In this part of the curve, the relationship $\phi(\Delta P)$ is approximatively linear. This phenomenon of flow limitation is caused by the narrowing of the pharyngeal duct, which effect counterbalances the increasing pressure drop. It has been observed in clinical conditions during a hypopnea, which are caused by partial closure of the upper airway [17]. Finally, when the pressure drop ΔP reaches approximatively 300 Pa, the flow stops as the airway is closed. This complete closure of the airway in response to inspiratory airflow corresponds to an obstructive apnea.

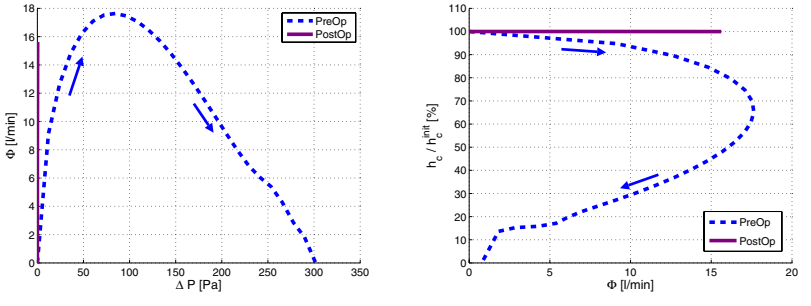
More details about the kinematics are given in figure 3. The figure 3 (a) depicts the flow velocity in the pharyngeal duct during the closure (when $\Delta P = 145$ Pa and $\phi = 14.8$ l/min). The phenomena already mentionned in 2.2 of flow separation and of jet formation can be observed. On the figure 3 (b), the Von

Mises strain distribution at the end of the simulation, after complete closure, has been plotted. The mean value is of the order of 5 %, which confirms the small deformations assumption. The peak values, of less than 20 %, are at the level of the mandible and of the hyoid bone (fixed nodes).



(a) (b)

Fig. 3. Simulation of the upper airway obstruction from pre-operative data. (a) Velocity of the fluid at the level of the constriction (isovalues). (b) Von Mises strain.



(a) (b)

Fig. 4. Comparison between pre-operative and post-operative simulations. (a) Airflow rate ϕ as a function of the pressure drop ΔP. (b) Constriction height h_c as a function of ϕ. The arrows in the pre-operative curves show the direction of increasing ΔP (from the initial configuration to the complete closure).

3.2 Comparison Between Pre-operative and Post-operative Data

In a second step, a simulation has been carried out on the geometry extracted from post-operative data of the same patient. The pre-operative and post-operative results are overlaid in figure 4. The airflow rate ϕ has first been plotted as a function of the pressure drop ΔP between the inlet and the outlet of the oropharynx (figure (a)). For a typical value of ϕ of 10 l/min, which is of the order of the rate recorded during sleep [18], it can be observed that:

- on the pre-operative simulation, the pressure drop ΔP associated to ϕ is approximatively 200 Pa.
- on the post-operative simulation, the same flow rate is obtained with ΔP close to 0 Pa (less than 1 Pa).

Indeed, the pre-operative case corresponds to a pharynx which is very narrow ($\simeq$ 1.5 mm) and thus very collapsible. Therefore, it is very resistive to the transit of the airflow. Conversely, in the post-operative case, the pharynx has been enlarged up to 10.5 mm so that it could be approximatively considered as a straight rigid duct, which offers few resistance to the airflow. Moreover, in the pre-operative case (figure (b)), as the flow rate ϕ is increased, the constriction height decreases down to a value of approximatively 65 % of the initial height, when ϕ reaches its maximum ($\simeq$ 17.5 l/min). Then, both the constriction height and the flow rate decrease, until complete obstruction. In the post-operative case, the initial constriction height is so high that no significant decrease can be observed.

For the patient 2, the behaviour of the model is the same as for patient 1. In the pre-operative case, the occlusive pressure[4] P_f is respectively of 295 Pa and of 260 Pa for patients 1 and 2. The maximal airflow ϕ_{max} is approximatively 18 l/min (17.6 l/min for patient 1 and 18.1 l/min for patient 2). For the two patients, no significant variation of the constriction height can be observed in the post-operative case.

4 Discussion

4.1 Clinical Interest of the Simulation Software

First, the two simulations from pre-operative radiographies predict an important predisposition to complete closure, while there is no more significant impact of the airflow on the upper airway deformation in the post-operative case, when the pharynx has been enlarged. This is in good agreement with clinical data (see table 1). This can be considered as a first qualitative in-vivo validation of the simulations. The second point is that the simulation time is of the order of 20 mn[5]. This makes the software compatible with clinical applications[6]. Moreover, with our method, it is not difficult to change the geometry from one patient to another. Indeed, a new geometry of the tongue corresponds to a new matrix $\mathbf{K}$ in the fluid-structure interaction loop (see section 2). This matrix can be easily obtained from segmentation on a radiography followed by a precomputation step. These points are important aiming at simulations for clinical studies or for examination of the effects of a given treatment (surgery or mandibular splint).

[4] The value of the outlet pressure P_s for which complete closure occurs.

[5] For a non-optimal version of the software, which actually is built in Matlab(TM) - software optimization should lead to computation times of a few minutes.

[6] The time constraints due to clinical practice are difficult to estimate with precision, and might be relativized as other tasks, such as imaging or polysomnography, are very time consuming. Nevertheless, the faster the simulations run, the more comfortable it is for the clinician.

4.2 Actual Limits and Perspectives

The assumptions stated for these simulations could appear as very restrictive and one could wonder whether they do not prevent from taking into account important aspects of the phenomenon. However, a long work of physical analysis and of experimentation has been done in order to proove they are correct as a first approximation [8] [9] [10]. Moreover, as regarding the accessible medical data, more general assumptions and a more sophisticated formulation would be superfluous for the moment. Indeed, postoperative CT-scans or MRI, necessary for validation of a 3D model, are not currently available. Concerning the soft tissue, before envisaging a more complex behaviour (hyperelastic or viscoelastic), the first task would be to determine accurately their Young modulus. A first estimation has been taken from [7], but it would be better to proceed to in-vivo measurements of the tongue rheology on apneic patients, with an instrument such as the one described in [19]. This will be the first short-term perspective of this work. The second short-term perspective will be to obtain a larger sample of sagittal radiographies, so as to confirm our first results. Concerning long-term perspectives, an important one would be to extend the formulation to the tridimensional case, using CT-scans or MRI, which will of course allow to take better into account the complex morphology of the upper airway. Other sites of obstruction, such as the velopharynx or the hypopharynx [20], should be also explored. The choice of the oropharynx has been suggested by the treatment, as maxillomandibular surgery affects mostly this site. Another long-term perspective would be to build models from less restrictive assumptions, and to test through comparison with in-vitro and in-vivo measurements the impact it has on accuracy and on computation time.

References

1. Malhotra, A., White, D.: Obstructive Sleep Apnoea. The Lancet **360** (2002) 237–245
2. Young, T., Palta, M., Dempsey, J., Skatrud, J., Weber, S., Badr, S.: The Occurence of Sleep-Disordered Breathing among Middle-Aged Adults. The New England Journal Of Medicine **328**(17) (1993) 1230–1235
3. Ayappa, I., Rapoport, D.: The Upper Airway in Sleep : Physiology of the Pharynx. Sleep Medicine Reviews **7**(1) (2003) 9–33
4. Li, Z., Kleinstreuer, C.: Blood Flow and Structure Interactions in a Stented Abdominal Aortic Aneurysm Model. Medical Engineering and Physics **27** (2005) 369–382
5. Tada, S., Tarbell, J.: A Computational Study of Flow in a Compliant Carotid Bifurcation-Stress Phase Angle Correlation with Shear Stress. Annals of Biomedical Engineering **33**(9) (2005) 1202–1212
6. Shome, B., Wang, L., Santare, M., Prasad, A., Szeri, A., Roberts, D.: Modeling of Airflow in the Pharynx with Application to Sleep Apnea. J Biom Eng **120** (1998) 416–422
7. Malhotra, A., Huang, Y., Fogel, R., Pillar, G., Edwards, J., Kikinis, R., Loring, S., White, D.: The Male Predisposition to Pharyngeal Collapse. Am J Respir Crit Care Med **166** (2002) 1388–1395

8. Hirtum, A.V., Pelorson, X., Lagrée, P.: In Vitro Validation of Some Flow Assumptions for the Prediction of the Pressure Distribution during Obstructive Sleep Apnoea. Medical & Biological Engineering & Computing **43** (2005) 162–171
9. Chouly, F.: Modélisation Physique des Voies Aériennes Supérieures pour le Syndrome d'Apnées Obstructives du Sommeil. PhD thesis, Grenoble, France (2005)
10. Chouly, F., Hirtum, A.V., Lagrée, P., Pelorson, X., Payan, Y.: An Attempt to Model Fluid-Structure Interaction during Obstructive Sleep Apnea Syndrome: Numerical Simulations and Validation. Journal of Fluids and Structures (2006) submitted.
11. Payan, Y., Perrier, P.: Synthesis of V-V Sequences with a 2D Biomechanical Tongue Model Controlled by the Equilibrium Point Hypothesis. Speech Communication **22** (1997) 185–205
12. Carpenter, P., Pedley, T., eds.: Chap. 2. Flows in Deformable Tubes and Channels. Theoretical Models and Biological Applications (M. Heil and O.E. Jensen). In: Flow in Collapsible Tubes and Past Other Highly Compliant Boundaries. Kluwer (2005)
13. Napadow, V., Chen, Q., Wedeen, V., Gilbert, R.: Intramural Mechanics of the Human Tongue in Association with Physiological Deformations. J Biomech **32** (1999) 1–12
14. Zienkiewicz, O., Taylor, R.: The Finite Element Method. Basic Formulation and Linear Problems. McGraw-Hill Book Company (1989)
15. Cotin, S., Delingette, H., Ayache, N.: Real-Time Elastic Deformations of Soft Tissues for Surgery Simulation. IEEE Transactions On Visualization And Computer Graphics **5**(1) (1999) 62–73
16. Lagrée, P., Lorthois, S.: The RNS/Prandtl Equations and their Link with Other Asymptotic Descriptions: Application to the Wall Shear Stress Scaling in a Constricted Pipe. International Journal of Engineering Science **43** (2005) 352–378
17. Gould, G., Whyte, K., Rhind, G., Airlie, M., Catterall, J., Shapiro, C., Douglas, N.: The Sleep Hypopnea Syndrome. Am Rev Respir Dis **137** (1988) 895–898
18. Trinder, J., Kay, A., Kleiman, J., Dunai, J.: Gender Differences in Airway Resistance during Sleep. J. Appl. Physiol. **83**(6) (1997) 1986–1997
19. Bruyns, C., Ottensmeyer, M.: Measurements of Soft-Tissue Mechanical Properties to Support Development of a Physically Based Virtual Animal Model. In: MICCAI 2002. (2002) 282–289
20. Rama, A., Tekwani, S., Kushida, C.: Sites of Obstruction in Obstructive Sleep Apnea. Chest **122**(4) (2002) 1139–1147

Image Based Musculoskeletal Modeling Allows Personalized Biomechanical Analysis of Gait

Lennart Scheys[1,2], Ilse Jonkers[2], Dirk Loeckx[1], Frederik Maes[1], Arthur Spaepen[2], and Paul Suetens[1]

[1] Medical Image Computing (ESAT/PSI), Faculties of Medicine and Engineering, University Hospital Gasthuisberg, Herestraat 49, B-3000 Leuven, Belgium
lennart.scheys@uz.kuleuven.ac.be
[2] Laboratory of Ergonomics and Occupational Biomechanics (Department of Kinesiology), FABER/K.U.Leuven, Tervuursevest 101, B-3000 Leuven, Belgium

Abstract. This paper describes a workflow, for building detailed, subject-specific musculoskeletal models from magnetic resonance (MR) images allowing enhanced biomechanical analysis of gait. . Bones are segmented semi-automatically using a hybrid approach while muscles attachments are retrieved automatically by atlas-based non-rigid registration followed by optional interactive correction using a user-friendly interface. Compared to previously proposed methods for MR based musculoskeletal modeling, integration of automated image processing procedures and problem-tailored visualization techniques result in a considerable reduction of the processing time, thus making MR-based musculoskeletal modeling practically feasible and more attractive.

1 Introduction

Gait analysis has proven to be a valuable tool for the clinical management of walking abnormalities of patients suffering from a wide variety of medical conditions. Especially in cases demonstrating a combination of motor coordination dysfunction and skeletal deformity, as present in cerebral palsy (CP), its added value has been documented [1].

Biomechanical analysis of muscle function during walking requires the use of musculoskeletal models describing the musculoskeletal geometry and moment generating capacity of the muscles [2]. More recently these new 'derived' parameters, closely related to musculoskeletal geometry (e.g. muscle length, muscle moment arms) and muscle-tendon dynamics (e.g. force generating capacity of muscles), have been introduced in the clinical decision making process [3]. As a consequence, accurate analysis of gait has an even greater need for models accommodated for inter-individual variability in musculoskeletal geometry [4], especially when analyzing movement in a pediatric population or in pathological conditions where bony deformations are present [5].

Subject specific musculoskeletal modelling refers to the need for the model to reflect the subject's characteristics in the definition of the bones, muscles and joints. Two solutions are possible. One option is rescaling and deforming a generic

M. Harders and G. Székely (Eds.): ISBMS 2006, LNCS 4072, pp. 58–66, 2006.

model to approximate the patient's musculoskeletal geometry [6]. Alternatively, subject-specific models can be built based on radiological images. Previous research [7, 8] has proven the potential to build accurate personalized models using magnetic resonance (MR) images. However, these methods are labor intensive and time consuming [7] since they rely on manual segmentation of the various muscles and bone structures of interest in the images, where parametrization of the muscle structures is the most time demanding step [9]. In this work a workflow is presented for the definition of image based patient-specific musculoskeletal models integrating automated image processing procedures and problem-tailored visualization techniques. As a result, this approach requires less user input and is less time consuming than manual segmentation, while delivering models with a higher level of patient-specific detail than achieved by deformed generic models.

2 Implementation

2.1 Model Structure

The biomechanical models suitable for gait analysis, are formatted for use in SIMM (Software for Interactive Musculoskeletal Modeling): a toolkit that facilitates the definition, visualization and analysis of three-dimensional musculoskeletal systems [10]. In SIMM, a musculoskeletal model consists of a set of body segments, containing one or more bones, interconnected by joints. The joint definitions consist of kinematic functions that define the relative positions of the body segments for all possible combinations of joint angles. Finally, the model includes muscle-tendon actuators, presented as a line model between origin and insertion on adjacent body segments. Furthermore additional wrapping points may be defined in case the muscle deviates from a straight line between origin and insertion or to prevent penetration of the muscle into the bones. The geometrical orientation of individual musculotendon actuators with respect to the joint, combined with the force generating capacity of the muscles, allows the calculation of the moment generating capacity around the joints.

The following sections will discuss the definition and parametrization of each of these anatomical structures based on image data. The Visualization Toolkit (VTK, www.vtk.org) is applied for visualization and creating a user-friendly GUI for 3D manipulation of the various model parts.

2.2 Image Acquisition and Post Processing

In this research, T1 weighted, axial spin echo MR images were used. Subjects were lying supine with extended knees. The field of view can cover the entire lower extremity, proximal from the anterior extremity of the iliac crest (Anterior Superior Iliac Spine, ASIS) until the distal end of the tibia. For the current application, T1 weighted images are chosen because of their feature of depicting intramuscular fat hyperintensely which significantly facilitates the distinction of individual muscle groups. To shorten data collection time, the slice thickness varied between series covering joints ($3mm$) and series covering the shaft of the

femur and tibia ($12.58mm$), where less detail is needed. Care was taken that the overlap between image series measured at least $9mm$. The entire acquisition takes about 11 minutes.

After acquisition the different series of the image volume are automatically registered rigidly two by two to obtain one large image volume covering the entire field of view. This is done automatically using a validated iterative approach which uses maximization of mutual information as a registration criterion [11]. By performing this step it is possible to operate on the entire volume of interest at once. Previously [7], image processing was performed series by series which required an additional computationally intensive two-by-two registration step for each separate segmentation result.

Occurring intensity inhomogeneities due to bias fields are removed from the medical images in a third and last step. The bias field is automatically estimated using a validated iterative expectation-maximization strategy [12].

2.3 Bone Segmentation

Various segmentation techniques have already been proposed in the literature. Several of these algorithms have been implemented and evaluated in previous research [9]. A hybrid approach, proposed by Zoroofi et al. [13], was chosen to define the bone structures. The segmentation procedure consists of several consecutive steps.

In a first stage a region of interest is selected which will be used in all subsequent steps. Next, a $3 * 3$ median filter is applied to smooth each image slice and reduce noise and remaining intensity bias. A suitable bone threshold is then automatically calculated for each filtered slice by fitting a Gaussian function to the appropriate peak in the slice's gray-value histogram. In a final step a series of threedimensional binary morphological operations are employed to remove remaining connections between the bone and the surrounding fatty tissue, as well as connections between adjacent bones. The respective operations are: (1) a morphological erosion to remove unwanted connections, (2) a threedimensional connectivity test to filter out all disconnected non-bone tissue and finally (3) a morphological dilatation and hole filling procedure to recover lost voxels representing the bone. The segmented image serves as input to create a tetrahedral mesh using the marching cubes algorithm [14].

Finally, the resulting mesh is smoothed, based on the Laplacian smoothing algorithm. This results in a polygonal mesh of the bone based on a more evenly distributed point cloud while preserving topology [15]. Examples of resulting bone objects can be seen in figure 2 and in more detail in 1.

2.4 Definition of Muscular Structures

Muscle definition requires identification of its attachment and insertion sites. Since line models are used to describe the muscle's line of action, attachment regions need to be identified using a centroid approach, as described by Koskinen et al. [16]. Furthermore, identification of additional wrapping points may be required as stated in section 2.1.

The muscle modeling procedure consists of two consecutive steps. In a first step the positions of all muscle insertions are initialized (figure 1(a)). This is done by non-rigid registration of the patient images with an atlas image wherein coordinates from all necessary muscle insertions are known. The registration procedure uses quadratic splines to model the deformation and mutual information as a similarity measure [17]. Initial insertion positions are defined by applying the resulting deformation to the original atlas coordinates.

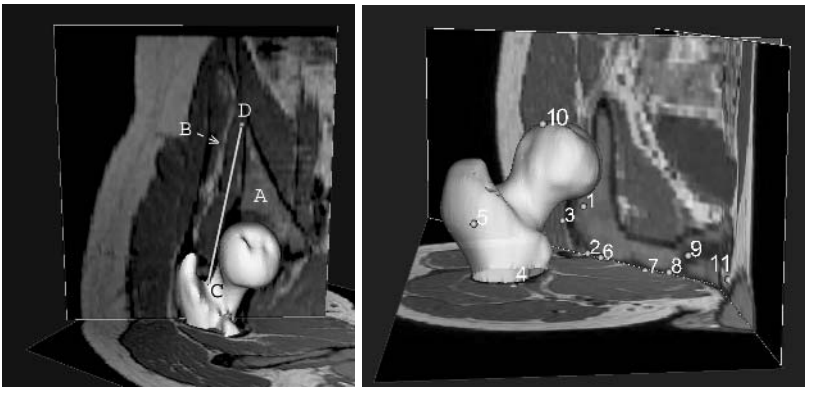
The second step allows to manually correct the suggested muscle insertions. To improve accuracy and accelerate the procedure, the medical image data and previously segmented bone structures are co-visualised in one 3D scene allowing for an easier interpretation of the anatomy in the 3D images. Furthermore, the user can switch between, or combine sagittal, axial or transversal slices to visualize the medical image data. A final feature allows the user to interactively place up to three separate image planes within the image volume with an arbitrary orientation, position and size. Accurately marking a muscle insertion is difficult without having a clear picture from the orientation of the muscle belly. In previous research manual segmentation of the muscle in consecutive image slices was required for this reason [7]. By positioning an image plane coinciding with the muscle belly along a significant part of its running length, a general overview of the anatomy is delivered therefore facilitating the identification of muscle insertions without the need for a manual segmentation step. An example is shown in figure 1(b).

2.5 Joint Kinematics

The positions of the different joint centers serve as the application points of the moments generated by the force-developing muscles and ligaments. These positions are closely related to the shape of the bones that define the joint under consideration. The previously segmented bone structures are used to derive the joint centers and the relative movement of the body segments during gait.

The coordinate systems for the tibia, femur and pelvis are chosen in accordance with the definitions used by Arnold in [7]. In a first step their origins are calculated in the scanned postion: 2(a). The origin of the pelvic coordinate system is situated at the midpoint between left and right ASIS. The origin of the femoral body segment is defined at the center of the femoral head which is determined by fitting a sphere to the femoral head, using the Iterative Closest Point algorithm (ICP) [18]. The center of the tibial coordinate system is located at the midpoint between the lateral and the medial epicondyles. This point is identified in two steps. In a first step the outer boundaries of both condyles are delineated after which an ellipse is fitted trough each of this contours. Afterwards, the centers of both ellipses are calculated and their midpoint is marked as the tibio-femoral joint center.

Kinematic descriptions, defining the relative positions of these coordinate system for all possible angles of the hip and knee joint, originate from those described in [19]. They are first transformed with a translation followed by a rotation so that the bone positions described by the adjusted kinematics



(a) (b)

Fig. 1. (a) Interactively placing an image slice within the image volume allows to visualize a significant part of the muscle belly of the gluteus minimus, delineated by the pelvic bone (A) and the intramuscular fat (B). This gives the user a clear picture of the muscle's main orientation, necessary for accurate parametrization of the muscle's insertions (C and D). (b)Muscle origins obtained by non-rigid registration with an atlas image: (1)long head of biceps femoris, (2)semitendinosus, (3)semimembranosus, (4)vastus medialis, and (5)lateralis, (6) adductor magnus inferior, (7)medialis and (8)superior, (9)adductor brevis, (10)rectus femoris and (11)adductor longus. Combining interactively positioned image slices, together with segmented bone models in a mutual threedimensional workspace, makes anatomical interpretation of the presented medical image data easier and allows better evaluation of the defined muscle insertions.

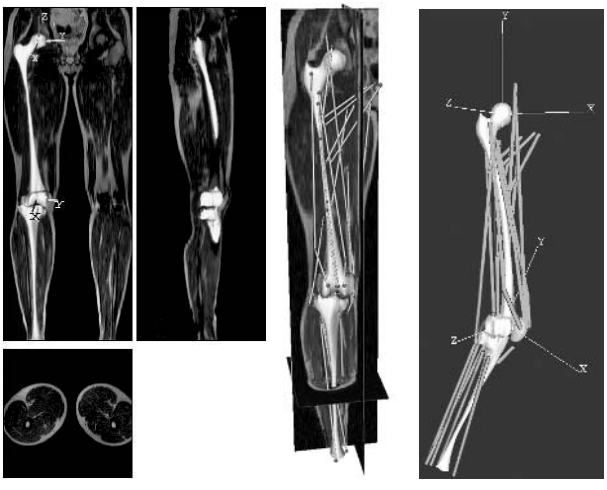
coincide with those calculated in the previous step. Subsequently, they are scaled to adapt to the specific geometric dimensions of the bone surfaces that define the joint.

3 Results

Figure 1(b) shows initial estimates for origins of 11 muscles situated around the hip joint, obtained by non-rigid registration. All insertions are lying on the outside layer of the pelvic an femoral bone at anatomically plausible positions when referenced to anatomy books [20]. This illustration especially focuses on the muscular anatomy around the hip joint an area of specific interest in children with CP which often suffer from significant musculoskeletal deformations around the hip joint.

Figure 2(b) shows how the proposed workflow was successfully applied to construct a highly detailed subject-specific musculoskeletal model, based on MR images (figure 2(a)) from a 25 year old non-pathologic male. Next to bone models for femur, tibia and patella, the resulting model contains the lines of action from 20 major muscles of the lower extremity: adductor longus, add brevis, add magnus superior, medialis and inferior, gracilis, semimembranosus, semitendinosus, biceps femoris (long head), sartorius, rectus femoris, vastus medialis, intermedius

and lateralis, gastrocnemius medialis and lateralis, soleus, tibialis anterior, tibialis posterior and the patellar ligament. The model was successfully imported in SIMM were the calculated subject-specific joint kinematics measured during gait analysis can be imposed on the model (figure 2(c)) for further biomechanical analysis.



(a) (b) (c)

Fig. 2. (a) Starting from T1 weighted SE MR images a subject-specific musculoskeletal model was build containing all necessary bone models and joint kinematics form the hip and knee joint. (b) Lines of action from 20 major muscles were integrated in the model. (c) Finally, the model was successfully imported in SIMM were it can be animated (notice the implied knee flexion) using the calculated biomechanics and used for biomechanical analysis.

4 Discussion

MR based musculoskeletal modeling has the potential to provide an accurate and efficient means of estimating muscle-tendon lengths and moment arms during gross motor activity such as gait [7]. In contrast with previous research, the approach proposed in this paper combines state-of-the-art problem-specific visualization techniques and medical image computing technology which makes MR-based musculoskeletal modeling practically feasible and more attractive.

Registration of individually segmented pieces of musculoskeletal structures was made unnecessary by merging all image series into one large image volume covering the entire lower extremity. Furthermore, it gives the user a more complete picture of the total field of view and the anatomy depicted therein compared to handling each series separately.

The specific technique which is proposed for segmenting bone structures was selected after considering the trade-off between speed, segmentation quality

and user-friendliness. In previous work [9], two other techniques were tested: iso-contouring and the level-set algorithm by Osher and Sethian [21]. While iso-contouring was slow and labor-intensive, using a level-set approach made it possible to automatically acquire qualitative bone models. However, careful tuning of several parameters is required to prevent the evolving segmentation front from passing through the bone-fat and bone-bone interfaces. The technique presented in this paper yields bone models of comparable quality within an equal timeframe of 11 minutes [13] without the need for parameter tuning.

Segmentation of the bone relies on the inside contour of cortical bone, which is easier to extract than the "outline" of cortical bone. The cortical layer is rather thin in the area of the joints so that the choice between using the inner or outer contour has little effect in those important regions. Along the shaft of the femur and the tibia, however, the cortical layer becomes substantial, resulting in a narrowed shaft. Apart from the femoral head, the femoral condyles and the tibial plateau, bone structures are only used for aiding anatomical interpretation thus requiring less precision in those areas.

Initializing muscle definition by non-rigid B-spline registration with an atlas image significantly enlightens and speeds up the process of muscle parametrization. It narrows the user's search space when looking for muscle insertions by projecting anatomical a priori knowledge on the medical image data in a subject-specific way. While previous research only used geometric characteristics of bone landmarks to calculate muscle insertions [22], this method accounts for the anatomical variability of muscle attachments [4] by also incorporating the patient's fat and muscle tissue when calculating subject specific muscle origins and insertions. Because this approach hasn't been tested yet on pathologic image data, future research has to point out if the registration algorithm is flexible enough to compensate for defective musculoskeletal geometry or if a more elaborate atlas has to be used.

Previous testing [23] revealed that even for an adult, non-pathological subject with leg height comparable to a generic model, simple rescaling of the generic model results in significant differences in the calculated muscle moment arms when compared to a subject-specific model, built using MR images. For some muscles, opposite muscle functions were predicted by both models in certain parts of the gait cycle. Larger discrepancies are to be expected in a pediatric population, therefore strengthening the cause and goal of this research.

Acknowledgements

This work was supported in part by FWO-grants G.0570.05 and G.0566.06.

References

1. Cook, R.E., Schneider, I., Hazlewood, M., Hillman, S., Robb, J.: Gait analysis alters decision-making in cerebral palsy. J Pediatr Orthop. **23**(3) (2003) 292–295
2. White, S.C., Yack, H.J., Winter, D.A.: A three-dimensional musculoskeletal model for gait analysis. Journal of biomechanics, (22(8-9)):885-893, 1989 **22**(8) (1989) 885–893

3. Arnold, A., Liu, M., Schwartz, M., Ounpuu, S., Delp, S.: The role of estimating muscle-tendon lengths and velocities of the hamstrings in the evaluation and treatment of crouch gait. Gait and Posture **23**(3) (2006) 273–281
4. Duda, G.N., Brand, D., Freitag, S., Lierse, W., Schneider, E.: Variability of femoral muscle attachments. J Biomech. **29**(9) (1996) 1185–1190
5. Beals, R.K.: Developmental changes in the femur and the acetabulum in spastic paraplegia and diplegia. Dev med child neurol **11**(3) (1969) 303–313
6. Arnold, A., Salinas, S., Delp, S.: Evaluation of a deformable musculoskeletal model: Application to planning muscle-tendon surgeries for crouch gait. An. Biomed. Eng. **29**(1) (2001) 1–11
7. Arnold, A., Salinas, S., Asakawa, D., Delp, S.: Accuracy of muscle moment arms estimated from mri-based musculoskeletal models of the lower extremity. Comput. Aided Surg. **5**(2) (2000) 108–119
8. Cohen, Z.A., McCarthy, D., Roglic, H., Henry, J., Rodkey, W., Steadman, J., Mow, V., Ateshian, G.: Computer-aided planning of patellofemoral joint OA surgery: Developing physical models from patient MRI. LNCS **1496** (1998) 108–119
9. Scheys, L., Jonkers, I., Schutyser, F., Pans, S., Spaepen, A., Suetens, P.: Image based methods to generate subject-specific musculoskeletal models for gait analysis using semi-automated segmentation. International Congress Series **1281** (2005) 62–67
10. Delp, S., et Loan, J.: A graphics-based software system to develop and analyze models of musculoskeletal structures. Comput. Biol. Med. **25**(1) (1995) 21–34
11. Maes, F., Collignon, A., Vandermeulen, D., Marchal, G., Suetens, P.: Multimodality image registration by maximization of mutual information. IEEE Trans. Medical Imaging **16**(2) (1997) 187–198
12. Van Leemput, K., Maes, F., Vandermeulen, D., Suetens, P.: Automated model-based bias field correction of MR images of the brain. IEEE Trans. Medical Imaging **18**(10) (1999) 885–896
13. Zoroofi, R., Nishii, T., Sato, Y., Sugano, N., Yoshikawa, H., Tamura, S.: Segmentation of avascular necrosis of the femoral head using 3-D MR images. Comput Med Imaging Graph. **25**(6) (2001) 511–521
14. Lorensen, W., Cline, H.: Marching cubes: A high resolution 3d surface construction algorithm. Computer Graphics - Proceedings of the SIGGRAPH **21**(4) (1987) 163–169
15. Vollmer, J., Mencl, R., Mller, H.: Improved laplacian smoothing of noisy surface meshes. Proc. 20th Conf. Eur. Assoc. for Computer Graphics **18**(3) (1999)
16. Koskinen, S., Kujala, U.: Patellofemoral relationships and distal insertion of the vastus medialis muscle: a magnetic resonance imaging study in nonsymptomatic subjects and patients with patellar dislocation. Arthroscopy **8**(4) (1992) 465–468
17. Loeckx, D., Maes, F., Vandermeulen, D., Suetens, P.: Nonrigid image registration using free-form deformations with a local rigidity constraint. LNCS **3216** (2004) 639–646
18. Besl, P., McKay, N.: A method for registration of 3-d shapes. IEEE PAMI **14**(2) (1992) 239–256
19. Delp, S., Loan, J., Hoy, M., Zajac, F., Topp, E., Rosen, J.: An interactive graphics-based model of the lower extremity to study orthopaedic surgical procedures. IEEE Trans. Medical Imaging **37**(8) (1990) 757–767
20. Cahill, D.R., Orland, M., Miller, G.: Atlas of cross-sectional anatomy. Lea and Febiger (1984)
21. Sethian, J.: Level set methods and fast marching methods. 2 edn. Cambridge University Press (1999)

22. Kepple, T., Arnold, A., Stanhope, S., Siegel, K.: Assessment of a method to estimate muscle attachments from surface landmarks. J Biomech. **27**(3) (1994) 365–371
23. Scheys, L., Jonkers, I., Spaepen, A., Suetens, P.: Individualized musculoskeletal modeling based on MR images: is it worth the effort? Gait and posture **22S** (2005) 20

Real-Time Simulation for Global Deformation of Soft Tissue Using Deformable Centerline and Medial Representation

Pengfei Huang, Lixu Gu, Xiaobo Li, Shaoting Zhang, Jianfeng Xu, Chen Lin, and Qi Fang

Image-Guided Surgery Laboratory, Shanghai Jiao Tong University, 800 Dongchuan Road, Shanghai, 200240, P.R. China
herofay@hotmail.com, gu-lx@cs.sjtu.edu.cn, lixiaobo@sjtu.edu.cn, kawaii_tony@hotmail.com, xujf@sjtu.edu.cn, chenlin_lc@hotmail.com, fangqiqi@sjtu.edu.cn

Abstract. Real-time simulation for global deformation of soft tissue, using Mass-Spring or traditional finite element method (FEM), is difficult because of the following reasons: (1) The linear elastic model is inappropriate for simulating large deformation for the probably unreasonable distortion; (2) The size of the 3D problem (the number of elements in the Mass-Spring or FEM mesh) is much larger than a 2D issue. In this paper, we propose a novel approach for these 2 problems: (1) deploying a deformable centerline, combined with medial representation method instead of the traditional linear elastic model, to simulate large motion and global deformation of a 3D tube like object; (2) applying a simplified algorithm which reconstructs the soft tissue surface by the medial representation method and reduces the complexity of a 3D problem.

1 Introduction

The simulation of human soft tissue has a long history in biomedical engineering and computer science. Nowadays, aiming to achieve virtual reality, the real-time requirement of simulation has been considered as the key part.

In the recent decades, simulations for local deformation of 3D objects can yield plausible results by using FEM or Mass-Spring [1, 2]. Local deformation means the deformation which is limited in a relatively small region of the object, like poking, nipping, pressing, and cutting. Besides small alternation of an object, we may also want to simulate the large scale deforming. However, as far as the global deformation is concerned, we can hardly find the solution by simply using FEM due to its expensive computational cost or Mass-Spring because of its inaccuracy, so that an efficient and accurate deformable modeling method for the simulation is still under further research.

Global deformation is commonly happened in medical domain and worth simulating for virtual surgery. We could take the human intestine as a good example for this case: while observing the patience's intestine, it is inevitable that the patience

M. Harders and G. Székely (Eds.): ISBMS 2006, LNCS 4072, pp. 67 – 74, 2006.

will move and so will his/her intestine move dynamically. In this situation, which is different from the other soft tissue's minor deformations like needle insertion, scalpel cutting or forceps nipping, the intestine will move globally. Zhuang proposed a non-linear FEM model to simulate the global deformation of object [3], but it is difficult to implement and expensive in computation.

Medial Representation was proposed to use Medial Structure to represent the deformation of an object [4, 5]. The traditional Medial Representation method costs too much computation so that, to improve the efficiency, we propose a Simplified Medial Representation Method (SMRM). Specifically, in this paper, we introduce a novel concept of Deformable Centerline and use it as the Medial Structure to model the global deformation based on SMRM.

Our work is distinguished from many other previous works by the following two aspects: 1) simulation on global deformation of tube like organ rather than the local deformation; 2) simulation based on the dynamic performance instead of static performance of the deformation.

In section 2, we would propose a brand new model of Deformable Centerline. Then in section 3, we would simplify the traditional Medial Representation method to simulate the global deformation in a real-time manner. At last, we would illustrate our experimental results and draw our conclusion.

2 Deformable Centerline Modeling Based on Mass-Spring

The notion of centerline was first proposed by Blum [4]. Intuitively, a centerline of a soft tissue is a central path which traverses that organ. However, some soft tissues may have complex structures, such as the human heart, so they have several branches connecting to the centerline or to each other through the organ and forming a skeleton [6]. In this paper, we use the term of centerline and our model will be principally based upon it. In the model, a fast and automatic centerline extraction algorithm was employed based on a Distance Mapping Method [7].

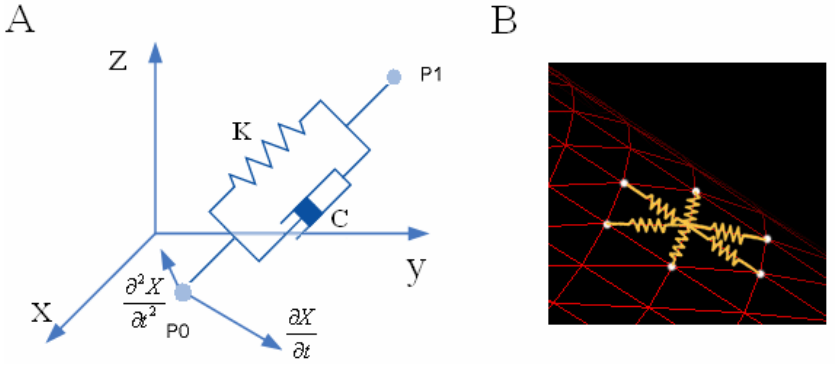


Fig. 1. (A) The spring element in 3D Mass-Spring system. (B) The Spring Mesh Structure.

2.1 Mass-Spring Model

The Mass-Spring system is a physical based technique that has been widely and efficiently used to model deformable objects [2]. A non-rigid object is modeled as a collection of point masses linked by springs in a mesh structure.

Mass-Spring is especially effective in simulating the dynamic behavior of the objects; however, it is innately unsuitable to simulate the static behavior due to its unstable vibration among the spring system and the shape distortions of the topology [8]. To reduce vibration and improve stability of the deformable model, damping factor and non-linear elasticity methods are used. In the dynamic Mass-Spring system, the equilibrium equation has the following form:

$$M\frac{\partial^2 X}{\partial t^2} + D\frac{\partial X}{\partial t} + KX = F(X) \tag{1}$$

where $\frac{\partial X}{\partial t}$ and $\frac{\partial^2 X}{\partial t^2}$ are the first and second derivatives of x with respect to time, M is the mass matrix, D the damping factor matrix, and K the stiffness matrix. F denotes the external forces. Equation (1) defines a coupled system of 3n ordinary differential equations for the n position vectors contained in X. To solve them, we could transfer equation (1) into a coupled system of linear equations based on Euler's first order method.

2.2 Centerline Modeling

After the extraction, the centerline is modeled to a Mass-Spring system, a set of points that are averagely distributed along its backbone and linked with springs by each two points in order of (Fig. 2 (A)). When a dynamic force is applied, a deformation bended reasonably and smoothly enough is expected (Fig. 2 (B)).

Mass-Spring system has the deficiency that the mass points and spring links are over discretized [7]. While the force is applying on one point, its propagation is not simultaneous. Therefore, an unexpected peak will occur, and the cause of this problem is the lack of upward forces on the neighbor points (Fig. 2 (C)).

To conquer the problems described above, we propose a novel method to model the deformation of the centerline.

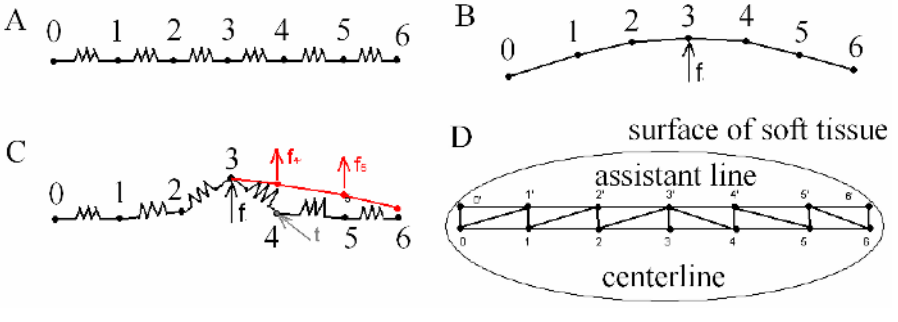


Fig. 2. (A) Mass-Spring Centerline. (B) Desirable Deformation Result. (C) Factual Deformation Result. (D) The Extended Mesh.

A dynamic offset line as an Assistant Line (AL) is constructed from the original centerline to refine the model composition and approximate the deformation.

To compensate the upward forces, tractive springs are designed and the extended mesh is established by the following steps: (1) Offset the centerline (P_0, P_1, …, P_n) in the direction of the applied force as the AL (P_0', P_1', …, P_n'); (2) Link the corresponding points of two lines, P_0 with P_0', P_1 with P_1', etc; (3) Given the force applying on P_k, link P_k' with P_{k+1}, P_{k+1}' with P_{k+2}, etc, and link P_k' with P_{k-1}, P_{k-1}', with P_{k-2}, etc, as the tractive springs.

Under the assistance of AL, the force applied on P3 will propagate orderly through P_2, P_1, and P_0 on the left side, P_4, P_5, and P_6 on the right side, yielding the desirable result described above (as shown in Fig. 3).

Moreover, in our model, it is assumed that the human soft tissue is attached to several fixed points, therefore they would return to the original position after motion or deformation via returning springs [9]. As a result, the deformable centerline always tends to return back to the original position as a stable status. To refine the model, angular spring is employed to simulate bending under the curvature force [10].

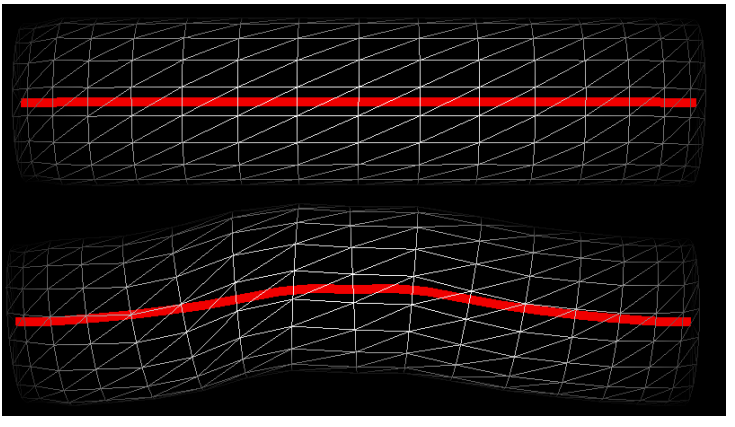


Fig. 3. Centerline Deformation Result Based on Mass-Spring with the Extended Mesh

3 Simplified Medial Representation Method

On the basis of Blum's medial axes and from Medial Representation, Prizer proposed the M-Rep method, which is excellent to represent the internal structure and uses medial atoms and a particular tuple $\{x, r, F(\vec{b}, \vec{n}), \theta\}$ to indicate the boundary of the deformable object [4, 5] (as shown in Fig. 4 (A)).

Traditionally, Medial Representation is used for registration and segmentation, but here we employ it for modeling the global deformation of human soft tissue which is similar to the bending of mitochondria and blood vessel. In the proposed simplified model, the Medial Representation method is adapted to a deformable model by deforming the centerline in advance and then accordingly alternating the surface mesh points to reconstruct the entire mesh rendering.

3.1 The Hubs and Spokes Structure

Due to the efficiency priority to the accuracy in simulation, the original Medial Representation method is simplified as hubs and spokes structure (as shown in Fig. 4 (B)) to sacrifice accuracy for speed. The medial atoms and tuples are refined as hubs and spokes, which imply the boundary information of the deformable object, to reduce the complexity of the method. The atoms are viewed as hubs, and spokes shoot out from them equally. In the simplified structure, one tuple is related to one spoke, containing less information, and spokes have various length, whereas in traditional Medial Representation such as M-Rep, each tuple is for each medial atom and radius is constant.

A novel method is proposed to calculate the static boundary C, and the tuple is reset to $\{x, r, F(V, \overline{AB}), \theta\}$ [9]. In the tuple, x is the position of the atom B; r denotes the radius of the spoke $\overline{BC}$; F represents the plane determined by vector V, called Orientation Vector of boundary orienting from A to C, and centerline segment $\overline{AB}$; the angle between a spoke $\overline{BC}$ and the centerline section $\overline{AB}$ is constant as θ.

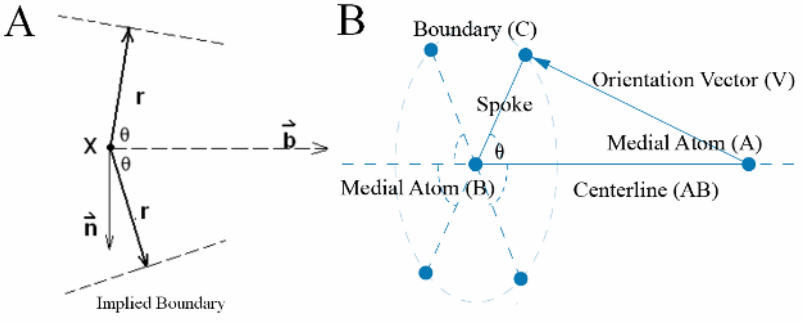


Fig. 4. (A) Medial Atom Tuple in Medial Representation. (B) Hubs and Spokes Structure.

3.2 Obtaining the New Boundary Location

As described in section 2, under the acted forces, the centerline deforms according to the dynamic law of the Mass-Spring system. It is presumed that only elasticity and curvature forces are applied on the centerline, while there is no constraint force. Therefore, the movements of the centerline are limited to translation and rotation around the atom or the combination of them, at the same time, there is no rotation around the central axis.

Above all, we propose the Orientation Vector mentioned in section 3.1. Orientation Vector, which is similar to the third element of Medial Representation, is used to establish the translation and rotation plane for the spoke. By translating the vector from the original point to the point after deformation, the centerline segment and the vector compose a plane, which contains the altered spoke.

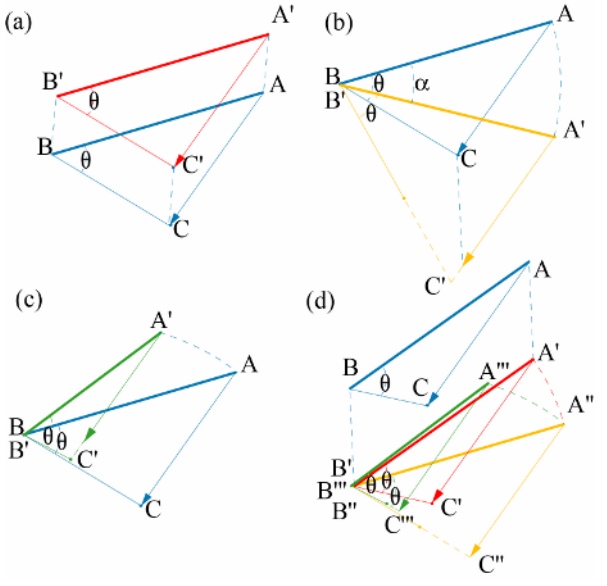


Fig. 5. (a) Translation. (b) Rotation within the Plane ABC. (c) Rotation vertical to the plane ABC. (d) Combination of Movements of (a), (b) and (c).

At this point, a lemma rule is introduced to validate the existence of the plane during dynamic deformation. Given the original atoms of *A*, *B*, the original boundary point *C*, and the deformed atoms of *A'*, *B'*, deformed boundary point *C'*, the lemma rule is described as the 4 cases below: 1) the centerline *AB* is translated to *A'B'* and the Orientation Vector *AC* is also translated to *A'C'* to determine the plane containing the moved spoke *B'C'*; 2) limited in the plane defined by centerline and spoke, rotating the centerline *AB* around the medial atom to *A'B'*(*A'B*), obviously, the plane is the same one; 3) limited in the vertical plane *ABA'* to the current plane *ABC*, the centerline *AB* rotates around the axis, which passes through *A* and *AB* is vertical to plane *ABA'*, to *A'B'*(*A'B*), then we translate the Orientation Vector *AC* from *A* to *A'* to determine the plane containing the spoke *B'C'*; 4) all the other movements can be defined as the 3 cases above – given *A'''B'''* as the destination, translating the centerline *AB* from *B* to *B'* as *A'B'*, then rotating *A'B'* in the plane *A'B'C'* around *B'* as *A''B''* until *A''B''* is at vertical intersection between plane *A'B'A''B''(A'A'' B'')* and *A''B''A'''B'''(A''A''' B''')*, and finally rotating *A''B''* in the plane *A''B''A'''B'''(A''A''' B''')* to *A'''B'''*. Fig. 5 shows the principles.

Under the lemma rule introduced above, the spoke BC as well as the boundary C can be uniquely determined after the deformation, and it is calculated by the following equation:

$$C = x + R_{V,\overline{AB}}(\theta, \overline{AB}) \cdot r_{\overline{BC}} / |\overline{AB}| \tag{2}$$

where x is the coordinate of medial atom B; $R_{V,\overline{AB}}(\theta, \overline{AB})$ is the operator to rotate $\overline{AB}$ by the angle of θ within the plane determined by V and $\overline{AB}$ (the procedure is

described in the lemma). $r_{\overline{BC}}$ is the length of $\overline{BC}$ and $|\overline{AB}|$ is the length of $\overline{AB}$ [9]. To determine the positions of other spokes as well as the boundaries, we simply rotate the spoke $\overline{BC}$ around the centerline axis $\overline{AB}$ and reset its length according to other spokes.

4 Experiments

Dynamic global deformation of elastic objects was implemented using Deformable Centerline and Medial Representation Method with a triangular mesh. On a 2.80 GHz Pentium IV PC with 1.0GMb DDR2 memory, and a GeForce 6800 graphics card, a centerline of 19 elements and a triangular surface mesh of 325 elements need about 0.02-0.04 seconds per time step. Fig. 6 shows the experimental results.

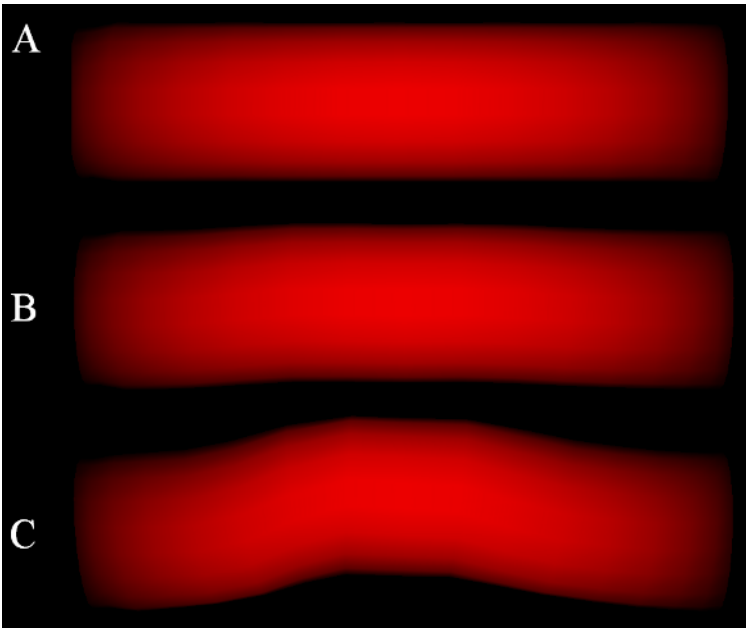


Fig. 6. (A) The Soft Tissue at Its Initial Status. (B) Bending Slightly. (C) Bending Intensively.

5 Conclusion

In this paper, a novel model is proposed to simulate the global deformation of soft tissue efficiently and effectively. This model combines the advantages of mass-spring model and medial representation model. On one hand, Mass-spring model is difficult to simulate large motion due to its inaccuracy of its linear mathematical formulation, but it is easy to solve. One the other hand, Medial representation method is ideal to represent the boundary information of an object however its computation cost is high.

As a whole, the proposed model uses deformable centerline of mass-spring system as the internal structure of soft tissue and applies simplified medial representation method to reconstructing and anchoring the boundary points.

In the future, we will endeavor to model more complicated soft tissue such as intestine, which has more rugae on the surface structure and a more roundabout and complex centerline structure, and some other organs which have long and narrow shapes. Aiming at a more accurate and robust level of virtual surgery, we will also exert ourselves to combine this model with the improved elastic model together and to refine the simulation.

Acknowledgement

The authors are grateful to Prof. Pizer for sharing his Medial Representation expertise, and Prof. Peters and Prof. Fenster for the valuable feedbacks. The research was partially supported by the Natural Science Foundation of China, Grant No. 70581171, and the Shanghai Municipal Research Fund, Grant No. 045118045.

References

1. M. Bro-Nielsen: Finite Element Modeling in Surgery Simulation. Proceedings of the IEEE: Special Issue on Virtual & Augmented Reality in Medicine, 86(3):524–530, March, 1998
2. S. Frisken-Gibson: Using Linked Volumes to Model Object Collisions, Deformation, Cutting, Carving, and Joining. IEEE Transactions on Visualization and Computer Graphics, VOL. 5, NO. 4, pp. 333-349, October-December 1999
3. Y. Zhuang et al.: Real-time and Physically Realistic Simulation of Global Deformation. ACM SIGGRAPH 99 Conference, ACM Press, pp. 270, 1999
4. Blum et al.: Shape Description Using Weighted Symmetric Axis Features. Pattern Recognition, 10: 167-180, 1978
5. S. Pizer et al.: Multiscale Medial Loci and Their Properties. IJCV Special UNC-MIDAG issue, 55(2/3): 155-1810, 2003
6. S. Pizer et al.: Segmentation of Single-Figure Objects by Deformable M-Reps. Medical Image Computing and Computer-Assisted Intervention, Vol. 2208 (2001) 862-871, 2001
7. M. Wan et al.: Automatic Centerline Extraction for Virtual Colonoscopy. IEEE Transactions on Medical Imaging, VOL. 21, NO. 12, December 2002
8. A. Duysak et al.: Fast Simulation of Deformable Objects. Eighth International Conference on Information Visualisation, Proceedings of Information Visualisation, pp. 422-427, 2004
9. S. Zhang et al. : Real-Time Simulation of Deformable Soft Tissue Based on Mass-Spring and Medial Representation. Computer Vision for Biomedical Image Applications, a workshop of ICCV2005, LNCS 3875, pp. 4110-428, 2005
10. L. Nedel et al.: Real Time Muscle Deformations Using Mass-Spring Systems. Proceedings of the Computer Graphics International, pp. 157-177, 1998

The Framework for Real-Time Simulation of Deformable Soft-Tissue Using a Hybrid Elastic Model

Shaoting Zhang[1], Lixu Gu[1,2], Weiming Liang[2], Pengfei Huang[1], Jan Boehm[3], and Jianfeng Xu[2]

[1] School of Software, Shanghai Jiao Tong University
kawaii.tony@gmail.com
http://www.se.sjtu.edu.cn/igst/
800 Dongchuan Road, Shanghai, P.R.China, 200240
[2] Computer Science, Shanghai Jiao Tong University
[3] Computer Science, Technische Universitat Berlin

Abstract. In this paper, we present a novel approach for real-time simulation of deformable soft-tissue. The framework includes segmentation of medical data, physical and mathematical modeling, performance optimization as well as collision detection. The physical and mathematical modeling which is the most significant phase in the approach is based on an improved elasticity theory which uses the skeleton structure of the deformable object to reflect volumetric information. We also refine the model to satisfy the real-time computation requirement and achieve a reasonable deformation effect on the global level as well as the local region by introducing the concepts of the angular spring, the return spring and the local deformation concept. A model based on real clinical data using a segmented left kidney and a collision detection demo are presented as an example in our case study.

1 Introduction

Today, most practical surgical training and education is done on living animals, cadavers and high-quality phantom models, all of which are expensive and might relate to ethical problems. Meanwhile, accompanied with the rapid development of medical image processing and soft-tissue modeling technology, virtual surgery appears to be a substitute of traditional training approaches. Thus Virtual Reality (VR) and real-time simulation are important in the biomedical imaging domain.

Numerous scientists have proposed various methods to model soft-tissue in the past years. Nicolas Ayache and INRIA presented a real-time Finite Element Method (FEM) [1] for virtual surgery. Sarah F. developed linked volumes (Mass Spring Systems) [2] to model object collisions, deformation, cutting, caving and joining. Stephen M. Pizer and Kaleem Siddiqi invented the Multi-Scale Medial Loci (M-Rep) [3] to represent the internal information of deformable objects and reconstruct the boundary by medial atoms. Sarah F. also developed 3D

M. Harders and G. Székely (Eds.): ISBMS 2006, LNCS 4072, pp. 75–83, 2006.

ChainMail [4] which is modeled by complicated calculations on a small number of elements and achieves a fast speed. However, existing methods are still not perfect. Although FEM is accurate enough, it is time-consuming and difficult to implement; Mass Spring Systems could meet with the real-time requirement when the number of mass points are limited [5], but the traditional Mass Spring System does not represent the inner structure appropriately, which might result in the unreasonable appearance under large scale force; M-Rep modeling of the global deformation is effective because of the skeleton structure, but the appearance of the surface is not good; ChainMail is fast enough but does not reflect the physics characteristic of soft tissue.

In our framework, we endeavor to deal with these problems mentioned above. We employ a proper segmentation method to obtain a regular surface mesh structure and establish a Mass Spring model based on the meshes. Then the skeleton of the soft-tissue is extracted by the Distance Mapping Method [6]. After that, the surface model and skeleton are correlated according to specific rules. Finally, the model is refined to reduce response time.

The following sections describe the details of each component in the framework. In section 2 we compare two sorts of surface mesh structures and chose the superior one. Section 3 models the soft-tissue using a hybrid approach. Section 3.3 introduces the local region deformation approach to decrease response time. Finally, we present the experiment results in section 5.

2 Surface Reconstruction

Medical image data supplied by the radiology department of Shanghai Oriental Hospital is used as source of the soft tissue model. In order to extract the organ rapidly and accurately, a Multi-stage Segmentation Approach [7] consisting of the Fast Marching method and Morphological Reconstruction is employed. In the first stage, a structuring element (e.g. a sphere with 1 pixel radius) is used to recursively erode the input 3D image until the Region Of Interests (ROI) is completely separated from the neighboring tissues. In the second stage, the Fast Marching method is employed to rapidly propagate the user-defined seed to a position close to the boundary. Then taking the output of the Fast Marching algorithm as the initial seed, Morphological Reconstruction is employed for refinement. In the final stage, recursive dilation is employed to recover the lost data from the first stage. The Multi-stage approach can make full use of the speed of the Fast Marching method as well as the accuracy of the Morphological Reconstruction algorithm. After the Multi-stage segmentation, Marching Cubes and Balloon Segmentation are employed separately on the segmented kidney to obtain two different surface meshes. Marching cubes is used for surface construction and can produce a triangle mesh by computing iso-surfaces from discrete data. Fig. 1(A) shows the complex mesh topology after the approach. Balloon Segmentation is a volumetric segmentation method based on dynamic deformable meshes [8]. An external image force is applied to an initial simplex mesh and makes the mesh expand or shrink towards the surface of the organ

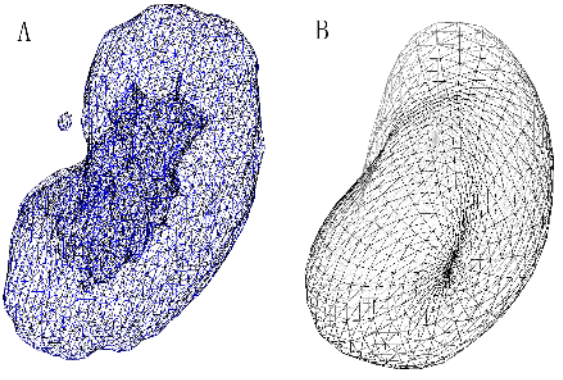


Fig. 1. Two sorts of meshes. (A) The mesh structure obtained by using marching cube. (B) The mesh structure acquired by employing balloon segmentation.

until the simplex meshes meets with the boundary. Fig. 1(B) displays the result of this method.

The former meshing algorithm generates extreme convoluted and irregular meshes with 2824 surface points and 5604 triangles, while the latter method delivers a relatively simple and regular mesh structure with just 994 surface points and 1984 triangles. Moreover, the former mesh topology is hollow and contains many fragments, which could result in the corruption of the mass-spring system under large large forces. Thus the latter mesh is preferred and appropriate for the following modeling process.

3 Soft-Tissue Modeling

We establish a surface model based on the Mass Spring system and employ a simplified M-Rep to represent the internal information. A hybrid elastic system is established by combining the two structures and later polished to reduce the calculation time.

3.1 The Surface Mass-Spring Model

The Mass Spring system models soft-tissue as a set of mass points linked with each other through springs. The topology of masses and springs is described in Fig. 2(A). Fig. 2(B) presents the construction of the fundamental unit of the system: the elastic spring and the damper. The former generates the elasticity force proportional to the change of the springs' length, and the latter provides a damping force proportional to the velocity of the mass points.

The system employs a differential equation to calculate the displacement of the surface points:

$$m_i \cdot \frac{\partial^2 \boldsymbol{x_i}}{\partial t^2} + d_i \cdot \frac{\partial \boldsymbol{x_i}}{\partial t} + \sum_{j \in \sigma(i)} k_{i,j} \cdot \delta l_{i,j} = F_i \tag{1}$$

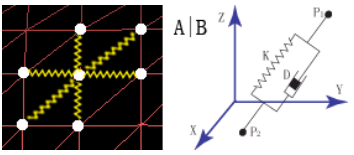


Fig. 2. (A) The topology of masses and springs. (B) The basic component of the Mass Spring system: masses (black dots), damper and spring.

Where m_i, d_i, and F_i are the mass, damping factor and external force of the ith mass respectively. x_i is the $3n$ vector displacement of the ith mass points. σ_i represents the mass points directly linked to the ith mass. $k_{i,j}$ and $\delta l_{i,j}$ denotes the elasticity factor and the length alternation of spring ij respectively. The explicit Euler method [9] is employed to solve formula 1 and computed the force on each node.

In the implementation, the Angular Spring [10] and the Return Spring [11] are introduced to refine and rationalize the deformation appearance. The former is employed to simulate the curvature force [10] which controls the degree of bending and twisting of soft-tissue. The latter is utilized to prevent soft-tissue from escaping from its original location.

3.2 The Skeleton Structure

S.M. Pizer introduced a deformable model (M-Rep) which uses medial atoms and a particular tuple $(\{x, r, F(\boldsymbol{b}, \boldsymbol{n}), \theta\})$ [3] to imply the boundary coordinates of soft-tissue. M-Rep is a sound approach to reflect the internal information because each medial atom represents an interior section [11] of the object.

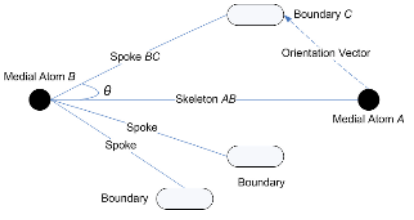


Fig. 3. The topology of the medial atoms and spokes

In the system the improved Distance Mapping Method [6] is employed to extract the skeleton from CT data, along which media atoms are selected evenly and automatically. Then we modify the topology of M-Rep to simulate the hub-and-spoke structure (Fig. 3) and alter the tuple to reduce the calculation time [11]. The tuple is altered to $\{x, r, F(\boldsymbol{V}, \boldsymbol{AB}), \theta\}$, where x is the coordinate of the medial atom B; r is the length of the spoke $\boldsymbol{BC}$; $F(\boldsymbol{V}, \boldsymbol{AB})$ is the plane

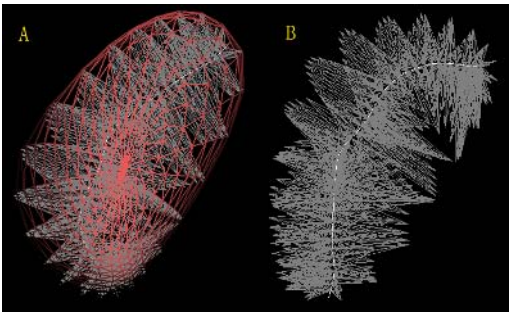


Fig. 4. (A) The hybrid elastic model consisting of surface mesh and internal topology. (B) The internal structure with spokes and the skeleton.

determined by vector $\boldsymbol{V}$ and $\boldsymbol{AB}$; $\boldsymbol{V}$ is the Orientation Vector of boundary C which links medial atom A with boundary C; θ is the angle between skeleton $\boldsymbol{AB}$ and spoke $\boldsymbol{BC}$. The coordinate of boundary C can be calculated by the following formula:

$$\underline{C} = x + R_{\boldsymbol{V},\boldsymbol{AB}}(\theta)\boldsymbol{AB} \cdot \frac{r_{\boldsymbol{BC}}}{|\boldsymbol{AB}|} \tag{2}$$

Where $\underline{C}$ and x are the coordinates of boundary C and medial atom B respectively; R denotes the operator to rotate its operand by the argument angle in the plane spanned by $\boldsymbol{V}$ and $\boldsymbol{AB}$; $|\boldsymbol{AB}|$ means the length of the vector $\boldsymbol{AB}$. Other spokes connecting with B can be calculated by rotating $\boldsymbol{BC}$ around $\boldsymbol{BA}$ and scale the r length. Iterating the process can obtain all boundaries.

3.3 The Hybrid Elastic Model

The surface Mass Spring system (Fig. 1(B)) and the skeleton topology (Fig. 4(B)) are combined in a hybrid elastic model. Firstly, each spoke will be linked with the nearest boundary points ([11]). However this topology is not robust enough due to the single connection between the centerline and each surface point. Thus each boundary point will be connected with the four nearest medial atoms by virtual springs (Fig. 4(A)). By doing this, the hybrid structure holds both the inner and the outer information which ensures a sound deformation effect on global level as well as local.

Finally, an iterative approach [11] is used to obtain the appropriate parameters of the elastic system, such as d, k, m of the inner structure and the outer mesh respectively. The values of the internal parameters should be much larger than the ones on the surface because the volume represented by the inner points is much larger than the one of the outside points.

4 Local Region Deformation

According to hospital statistics, most surgery operations are performed in a small area and the rests of the tissue is nearly stable [12], so we determined to restrict

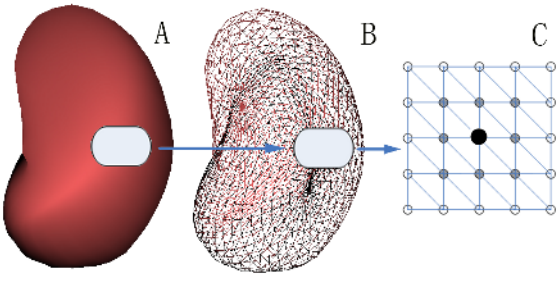


Fig. 5. (A) The local deformation region (the big grey spot) on a left kidney in the solid view. (B) (A) in the mesh view. (C) The topology of the local deformation region (The distance between the center black spot and the grey ones is 1, distance with the white ones is 2).

the deformation to a local level if the external force is relatively small in order to reduce the calculation time.

In the preprocessing stage of the application, the Dijkstra algorithm is employed to calculate the shortest path between all pairs of mass points. A table recording these distances is generated and kept. The lengths of all springs are assigned to be 1 in order to reduce the complexity and computation time. The deformation procedure is the following: if a small force is applied on the local deformation region (Fig. 5(A),(B)) and assuming that the propagation of the force is limited in the second layer, the application will search the distance table to find points whose distance from the black region is not greater than 2 (Fig. 5(C)) and these points will be transformed according to formula 1, while the displacement of other points, such as those marked 3 and 4 in the distance table, will be ignored and not calculated. The mechanism will be skipped when the external force is quite large because most mass points might be influenced under the global scale force.

5 Experiments

The framework is performed on a computer with an Intel Pentium IV-2.80 GHz CPU, 1G byte of ram, and a GeForce 6800 graphics card. The code is compiled with Visual C++ 7.0. There are 994 points on the surface, 48 points on the skeleton, and 256 strips in the surface mesh structure.

Fig. 6 compares the deformation results of the hybrid model with the surface mass-spring model. The hybrid elastic model receives a reasonable appearance on both local and global level (Fig. 6(A), (B), (C)). Fig. 6(A) and Fig. 6(B) also display the effect of collision detection. Fig. 6(D) reveal a unreasonable result of the surface mass-spring models.

Table. 1 displays the computation time of different models. Here local force means that we apply a relatively small force on a surface point, and global force means that a large forces on 10 points is applied. The update rate (including the

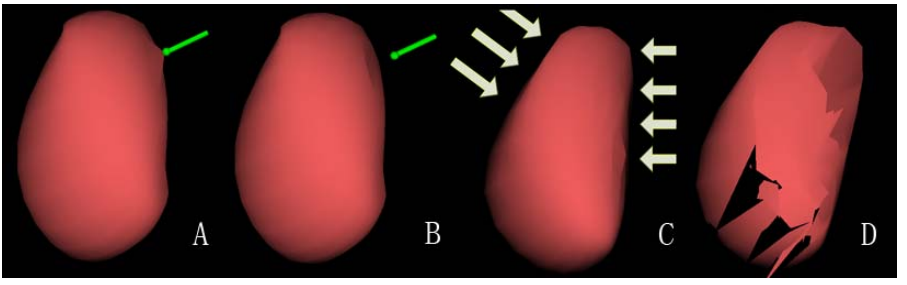


Fig. 6. Deformation effect of the segmented kidney: (A) Nip of the kidney and collision detection based on the hybrid model; (B) Release of the nip and local deformation; (C) Global deformation under large scale force; (D) Surface mass-spring model under large scale forces;

Table 1. Comparison of the calculation time (milliseconds per update time) among different elastic systems. (Surface MS = surface Mass Spring; Volume MS = Volume Mass Spring; Hybrid MS = the elastic system with internal structure; Hybrid LR = hybrid model with Local Region Deformation; LF = local force; GF = global force; 25, 50, 75, 100, 500 = the magnitude of the forces).

ms/update	Surface MS	Hybrid MS	Hybrid LR
LF(25)	0.531	0.766	0.750
LF(50)	0.547	0.829	0.765
LF(75)	0.546	0.812	0.782
LF(100)	0.563	0.859	0.797
GF(500)	1.891	11.187	7.938

rendering time) of the hybrid model with the local region optimization is about 150 times per second under local force and about 30 times under global force. Such rate is more efficient than the one without optimization and meets with the real-time requirement (15-20 times per second).

6 Conclusions

This paper introduced a convenient framework and methodology to model a generic soft-tissue and proposed a hybrid elastic system to simulate the deformation of soft-tissue effectively and efficiently on global level as well as in local regions. Our main contribution is combining the skeleton structure with the surface mesh, confining the deformation region when the external force is relatively small, and bringing in the concept of the return spring, all of which polish the appearance of soft tissue and decrease the response time.

The future work is to model more complicated organs such as the heart. In order to obtain the branches of the complex soft-tissue, the skeleton-extraction

algorithm needs to be improved. Moreover, the medial representation topology and the tuple mentioned in section 3.2 could be used to recalculate the positions of implied boundaries when the coordinates of the skeleton are changed. We endeavor to bring in this recalculation to better mix the advantages of the two models. Our recent work also includes the addition of force feed back equipment into the program.

Acknowledgements

This work was partially supported by the Natural Science Foundation of China, Grant No. 70581171, and the Shanghai Municipal Research Fund, Grant No. 045118045. The authors would like to thank Dr. Pizer (University of North Carolina) for patiently discussing the concept of M-Rep with members of our lab and Dr. Ayache (INRIA) for giving the impressive presentation about virtual surgery at ICCV2005. We also thank Guangxiang Jiang, a member in the Image Guided Surgery Therapy laboratory in SJTU, for developing the application to extract the skeleton of the kidney recorded in DICOM format.

References

1. Cotin, S., Delingette, H., and Ayache, N.: Real-time Elastic Deformations of Soft Tissues for Surgery Simulation. IEEE Transactions on Visualization and Computer Graphics (January-March 1999). Vol. 5, NO. 1, pp. 72-83
2. Frisken-Gibson, S.F.: Using Linked Volumes to Model Object Collisions, Deformation, Cutting, Carving, and Joining. IEEE Transactions on Visualization and Computer Graphics (October-December 1999). Vol. 5, NO. 4, pp. 333-349
3. Pizer, S.M., Siddiqi, K., and Szekely, G.: Multiscal Medial Loci and Their Properties. IJCV Special UNC-MIDAG issue. Vol. 55(2/3): 155-181
4. Sarah, F., Delingette, N., and Ayache, N.: 3D ChainMail: a Fast Algorithm for Deforming Volumetric Objects. Proc. 1997 Symposium on Interactive 3D Graphics. pp. 149-154.
5. Conti, F., Khatib, Q., and Baur, C.: Interactive Rendering of Deformable Objects Based on A Filling Sphere Modeling Approach. Proceedings of the 2003 IEEE International Conference on Robotics & Automation, Taipei, Taiwan (September 14-19, 2003)
6. Wan, M., and Liang, Z.: Automatic Centerline Extraction for Virtual Colonoscopy. IEEE Transactions on Medical Imaging (December 2002). Vol. 21, NO. 12
7. Gu, L., Xu, J., and Peters, T.: A Novel Multistage 3D Medical Image Segmentation: Methodology and Validation. IEEE Transaction Information Technology in Biomedicine. (in press)
8. Mitchell, B.R., Sahardi, T.A., and Vision, M.: Real-time Dynamic Deformable Meshes for Volumetric Segmentation and Visualization. In Proc BMVC 1997. Vol. 1, pp. 311-319
9. Chen, Y., Zhu, Q., and Kaufman, A.: Physically-based Animation of Volumetric Objects. Computer Animation 1998. pp. 154.
10. Nedel, L.P., Thalmann, D.: Real Time Muscle Deformations Using Mass-Spring Systems. Proceedings of the Computer Graphics International (1998). pp. 156-166

11. Zhang, S., Gu, L., and Huang, P.: Real-Time Simulation of Deformable Soft Tissue Based on Mass-Spring and Medial Representation. Computer Vision for Biomedical Image Applications, a workshop of ICCV2005. LNCS 3875, pp. 419-427
12. Chai, J.Y., and Sun, J.: Hybrid FEM for Deformation of Soft Tissues in Surgery Simulation. International Workshop on Medical Imaging and Augmented Reality (2001). pp. 298-303

A Flexible Framework for Highly-Modular Surgical Simulation Systems

S. Tuchschmid[1], M. Grassi[2], D. Bachofen[2], P. Früh[2], M. Thaler[2], G. Székely[1], and M.Harders[1]

[1] Computer Vision Lab, ETH Zurich, CH-8092 Zürich, Switzerland
{tuchschmid, szekely, mharders}@vision.ee.ethz.ch
[2] Inst. for Applied Information Technology, ZHW Winterthur, Switzerland
{bcd, frp, tha}@zhwin.ch

Abstract. We present a modular software framework which is currently used for high-fidelity surgical simulation of hysteroscopic interventions. Main design criteria was to meet various real-time requirements without losing maintainability or extensibility of the overall system. Moreover, communication and synchronization tools were developed for the multi-threaded environment. The efficiency and scalability of a convenient thread-based parallelization scheme is demonstrated for the distension fluid computation, as well as the collision detection algorithm. Performance measurements on a four processor system show an almost perfect scalability for larger problems.

1 Introduction

The main target of our current research is the development of a highly realistic reference simulator for procedural training of hysteroscopic interventions. The goal is to go beyond currently available systems (e.g. [9,12,10]) by focusing on the highest possible realism of the simulation. In order to provide the necessary fidelity, several specialized individual components have to be developed. This includes for instance modules for visualization, collision detection and response, soft tissue deformation, fluid simulation, or cutting. While these separate modules already comprise a high level of complexity, integration and control of all elements in a coherent system is also not straight-forward. Difficulties arise due to different requirements for update rates, varying model representations, or necessary synchronization of parallel threads. At the same time, we have to comply with several boundary conditions arising from a software engineering point of view. This is inevitable, since the project is following a long-term strategy (scheduled for twelve years), and thus will comprise the work of many generations of contributors. Therefore, the design of the modular framework has to fulfill several requirements:

Extensibility. Fast and simple integration of new results or components

Scalability. Utilization of additional computational resources in a scalable fashion, full support of multi-threading

M. Harders and G. Székely (Eds.): ISBMS 2006, LNCS 4072, pp. 84–92, 2006.

Real-time capability. Realistic simulator behavior requires compliance with various real-time constraints

Maintainability. System sustainability in a research environment with permanent change, and large number of developers

Flexibility. Modules and components exchangeable, supporting different clinical application domains and pathological situations

2 Related Work

A number of frameworks for surgical simulator systems have been proposed recently, the majority still being in a development stage. One of the first has been the SPRING system [11]. Several modules are provided, however, due to the limitation to a specific soft tissue deformation model it was not the optimal choice for our developments. Since several approaches to tissue modeling are going to be integrated into our application, a clear separation of the deformation model from the other modules was required.

Another recently presented system is GiPSi [3,7], an open source/open architecture framework for developing organ level surgical simulations. It already comprises several of the elements required in our system, but lacks components such as parallel computation or fluid handling. Moreover, the system was not available when the development of our simulator started in 2001.

Another related endeavor is the SOFA project [2]. It is a concerted activity of several renowned groups involved in the surgical simulation field. It targets at an extendible, open source framework for easy exchange of algorithmic blocks between research groups. However, it has not yet reached a fully functional state.

Apart from these activities directly focusing on surgical simulation, related work can also be found in the gaming industry. Similarities exists in some of the building blocks used in 3D game engines, which are available as commercial (e.g. [4,6]), as well as public domain solutions (e.g.[1,8]). Nevertheless, several fundamental elements are usually lacking, e.g. simulation of deformable objects or cutting.

3 Simulator Framework

In order to handle a highly modular real-time system, and achieve robustness and reliability without sacrificing performance, three major components were integrated. First, we rely on an object oriented design in order to meet the maintainability and extensibility requirements. Secondly, thread-based parallelization was integrated to ensure real-time capability as well as scalability. Finally, a synchronization toolset was developed to support stability and flexibility. Figure 1 gives an overview of the different simulator framework components.

3.1 Object Oriented Design

A domain model was defined in the object oriented design, which comprises instances of actual simulation objects and the simulation engines. The interaction

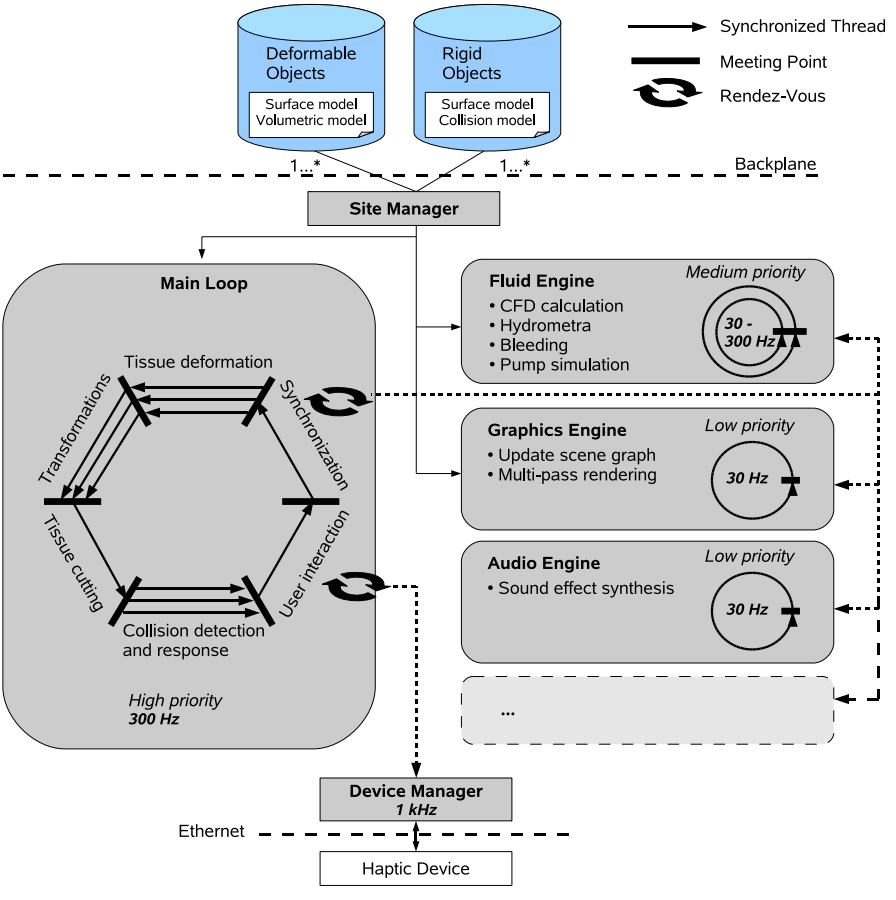
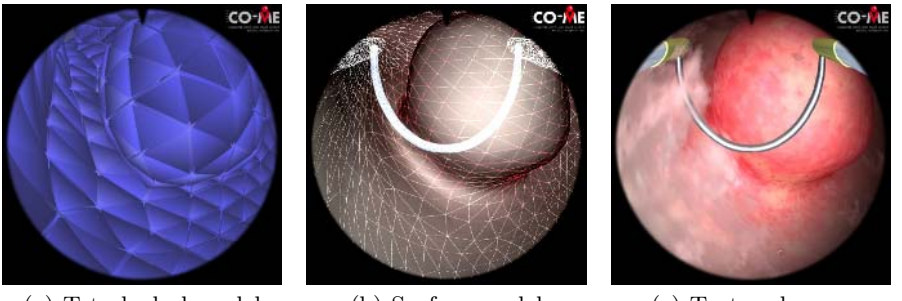


Fig. 1. Overview of the software framework

between all components is scheduled via a so-called *Site Manager*. The surgical scene consists of instances of *Deformable Objects* and *Rigid Objects*. Each of these is represented by separate physics and visualization models. The physics model of a deformable objects is based on a tetrahedral mesh, which is used for collision detection and response, as well as for calculating deformation. The visualization model is defined by a higher resolution triangular surface which is mapped to the lower resolution tetrahedral model and updated according to its changes (see Figure 2). This dual object representation has to be taken into account in all simulation engines, e.g. the cutting procedures require an update of both the surface and the volumetric model. For a rigid object a deformation model is not necessary. Nevertheless, to perform collision tests, the object geometry is approximated by collections of points. The visualization model is also based on triangular surface meshes. Each object instance is registered via the simulation backplane with the *Site Manager*. At least one deformable object for the uterine cavity and one rigid object representing the endoscopic tool have to be present.

In general, all modules involved in the mechanics of tool-tissue interaction build the core of the simulator, and are thus part of the *Main Loop*. It has the highest priority, since it is involved in the generation of appropriate haptic feedback. Six major elements can be identified. First, tissue deformation is computed for all deformable objects in parallel. Based on these deformations, the visualization models are updated in the same step. Thereafter, rigid objects and their corresponding collision meshes are updated in parallel. Next, if a cut is performed, deformation and rigid body update is halted. A hybrid approach [5] updates the corresponding surface and volume models, while assuring stability and continuity of the domain objects. If a complete piece of tissue is removed from a deformable object, then a new instance is created and registered with the *Site Manager*. After all objects have been updated, collisions are detected, and appropriate internal and external forces for soft-tissue deformation and force feedback are determined. The forces are communicated to the device manager, and user input is collected. The last element of the *Main Loop* performs synchronization with the lower priority engines.



(a) Tetrahedral model (b) Surface model (c) Textured scene

Fig. 2. Different model representations

The *Flow Engine* is responsible for the motion of the distension liquid which is determined according to simplified Navier-Stokes equations. The consumed processing power depends on both the grid resolution and the update rate, and can be tailored to the computing environment and the degree of realism needed. Besides providing fluid dynamics, the *Flow Engine* simulates the state of the hydrometra based on the pump valve settings and provides the motion of floating tissue and blood in the distension fluid.

As a further stand-alone unit within the framework, the *Vision Engine* is responsible for the handling and visualization of the surface data. The engine provides a hierarchical scene graph as an interface to the framework. The scene tree is fully dynamic, permitting the data structures to be modified at runtime. A realistic endoscopic camera model is integrated. which comprises lens distortion and camera focus. In a multipass rendering step, we use bump mapping for rough surfaces, environment mapping for shiny materials, and an extended illumination for spotlight effects.

User interaction is managed by the *Device Manager*. The user controls the simulation by 4-DOF movement of the hysteroscope, closing of the in-/outflow valves, setting of fluid pressure, changing camera parameters, and controlling the loop electrode. Interaction is possible either by an adapted hysteroscopic tool, a PHANToM haptic device, or mouse and keyboard. Forces calculated in the main loop can be sent to a haptic device to provide force-feedback. Moreover, all interaction data of a training session is recorded. This allows the replay and subsequent analysis of virtual interventions.

3.2 Synchronization

In order to simplify interaction between the different engines and to relieve the engine programmer from dealing with low-level synchronization issues, we developed specialized control mechanisms to prevent deadlocks, starvation or priority inversion. *Meeting Point*, *Rendezvous* and *Synchronized Thread* are concepts coming from the Parallel Programming area, which we adapted to our needs. The implementation of these tools is based on the POSIX library.

A *Meeting Point* is an extension of a barrier, where a coordinator thread releases the threads in the next group as soon as the last thread from the first group has joined at the Meeting Point. This allows to coordinate groups of worker threads and build arbitrary sequences of parallel blocks where each block has an arbitrary amount of parallel acting workers.

Rendezvous allows the exchange of information, like for instance the appearance data of all displayable objects, between independent threads. In the example mentioned, communication only has to take place when the data is needed by the vision engine, as these objects are rendered far less often than the computation of their behavior is performed. The synchronization decision is encapsulated by the service provider, thus the client sends requests to exchange information to a server without knowing the current state of the latter. The server can be either blocking or non-blocking, depending on the underlying real-time requirements. As soon as the server is ready, the necessary data exchange is performed.

The purpose of the *Synchronized Thread* is to encapsulate synchronization policies such that module developers do not need to know the underlying details. Immediately after being started the synchronized thread enters a loop and joins a Meeting Point. Once again released by the coordinator it calls its worker function. This way it is ensured that all descendants adhere to the protocol demanded by the system without dealing directly with synchronization issues. Since several of these threads can be grouped to run in parallel, we use them as the main scheme for parallelization.

While the synchronization tools are also successfully used to control the interaction between the different engines in the framework, the concepts are also applied to intra-modular parallelization. To achieve this, we first identify slow, parallelizable code sections. A domain decomposition is then carried out for an appropriate number of worker threads. A Meeting Point is added to coordinate the workers, while communication is taking place via Rendezvous. Two examples of applying these tools in our simulation framework are discussed below.

4 Results

4.1 Parallelized Fluid Computation

The flow of a liquid is described by the Navier-Stokes equations. In our system these equations are solved numerically on a uniform Cartesian grid to evaluate the time dependent behavior of the distension fluid. The local nature of the involved operators strongly motivates the tessellation of the spatial domain into partitions, which can be processed in a parallel way. In our case the straightforward partitioning is to cut the domain into horizontal layers (see Figure 3). The

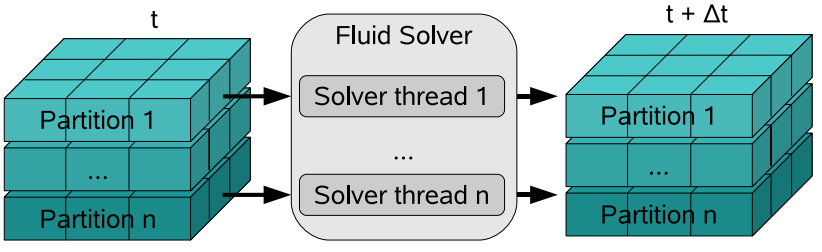


Fig. 3. Domain decomposition for the parallelized fluid solver

multi-threaded environment enables the threads to have convenient access to a shared memory area. We have to guarantee that the threads do not write to the same memory area at the same time. This is easily ensured since the general configuration during the velocity field calculation is such that the threads read from data arrays belonging to time-step t, and write data for their partition to a memory area corresponding to time-step $t+\Delta t$. Table 1 shows the corresponding pseudo-code.

The described tools are applied to achieve this thread-based parallelization. The fluid solver sets up a Rendezvous for synchronization with the Main Loop, creates a Meeting Point, and connects it with all partition solver threads. Once the fluid solver is entered, a call to the Meeting Point starts the worker threads. In a first step, an external force field is added describing the liquid in- and outflow. Next, the velocity field is diffused. The third step is the self-advection which can be interpreted as the velocity field moving itself. The last step assures mass conservation by balancing in- and outflow of the cell. As soon as all worker threads have finished the computation, a request for data exchange is placed, and the simulation time is advanced to $t + \Delta t$. Using three threads on a multiprocessor system we could achieve a speedup of 1.69 for a $22x22x22$ computation grid, and a speedup of 2.34 for a $40x40x40$ grid.

4.2 Parallelized Collision Detection

The collision detection module accounts for the interaction between deformable and rigid objects, and provides information for collision response. We use a

Table 1. Pseudo-code for the parallelized fluid solver

```
00  class FluidSolver implements Thread {
01    init() {
02      set fluid resolution
03      connect to RendezVous in MainLoop
04      create new MeetingPoint MP_PartitionSolver[NUM_THREADS]
05      for (int i=0; i < NUM_THREADS; i++) {
06         create new PartitionSolver and connect to MeetingPoint
07         set domain limits for PartitionSolver
08      }
09    }
10    run() {
11      start parallel threads for MP_PartitionSolver
12      request data exchange at RendezVous
13    }
14  }

20  class PartitionSolver implements SynchronizedThread {
21    run() {
22      addExternalForces
23      diffuseVelocityField
24      calculateSelfAdvection
25      assureMassConservation
26    }
27  }
```

highly optimized algorithm based on fast hashing of axis-aligned bounding boxes (AABB) [13]. In a first step, the algorithm classifies all object primitives, i.e. vertices and tetrahedra with respect to small AABBs, and maps the 3D cells to a 1D hash table. As a result, each hash table index contains a small number of object primitives that have to be checked against each other for intersection. The actual intersection test computes barycentric coordinates for all vertices with respect to a penetrated tetrahedron. This information can be employed to estimate the penetration depth for a pair of colliding tetrahedra which in turn can be used to calculate an appropriate collision response. This collision detection step is a major performance bottleneck of the fast *Main Loop*.

While the hashing of vertices and tetrahedra cannot be efficiently parallelized because of frequent write access to the common hash table, the more time intensive intersection test can. Similar to the flow computation, we split up the solution domain in equally large sets of vertices to optimize load balancing between the worker threads. The results of the parallelized collision detection are depicted in Table 2. Measurements were performed for different scenes, with 1, 2, or 4 enabled worker threads on a 4 x AMD Opteron 275 CPU 2.2 GHz system. As one can observe, the update rate increases with the number of threads. For larger datasets, the parallelization scales almost perfectly. However, for small problem sizes, speedup does not scale linearly with the number of threads.

Table 2. Parallelized collision detection for different scenes. (R update rate, μ load percentage of all 4 CPUs together, and Γ speedup.

	1 Myoma / 1 Uterus 3589 Tetrahedra			1 Polyp / 1 Uterus 5598 Tetrahedra			3 Polyp / 1 Uterus 10398 Tetrahedra		
# Threads	*R [Hz]*	*μ [%]*	*Γ*	*R [Hz]*	*μ [%]*	*Γ*	*R [Hz]*	*μ [%]*	*Γ*
1	300	25	-	170	25	-	40	25	-
2	440	39	1.5	270	43	1.6	80	46	2.0
4	630	51	2.1	370	61	2.2	150	81	3.8

4.3 General System Performance

Independent of targeted parallelization efforts, our modular framework already scales well when providing different numbers of CPUs. We measured update rates for different system configurations, where the hardware was either a single (with and without hyper threading) or a dual Pentium 4 CPU, 3GHz. Comparing main loop update rate and processor load, turning on the serial CFD module with a $64x64x28$ grid results in a drop for the single CPU from 155Hz/100% to 80Hz/100%, for the hyper threading version 180Hz/70% to 105Hz/100%, and for the dual system 180Hz/65% to 135Hz/95%. In a further test using the 4x CPU system, we also tested overall system performance. We used an organ model of 3198 tetrahedra and 1646 surface triangles, and a pathology of 206 tetrahedra and 1336 triangles. Grid resolution for flow computation was 32x32x18. The achieved update rates are 250 Hz for the Main Loop, 25 Hz for the Fluid Engine, and 30 Hz (clamped) for the Graphics Engine. With parallelized collision detection, the main loop rate increases to 330 Hz, and the fluid engine update rate to 33 Hz.

5 Conclusion

The proposed framework allows flexible real-time simulation in a multi-threaded, multi-processor environment. The object-oriented design of the framework has been proven to allow adapting and extending to many newly created internal and external modules. While the developed simulation modules are tailored to hysteroscopic interventions, the overall framework is not application specific and could be used for various other surgical simulation scenarios. A possible integration of some of the described parallelization mechanisms into the mentioned SOFA framework is currently being evaluated. The presented paradigms and tools could also be used to further parallelize lower level algorithms in order to speed up the simulation. This step is greatly simplified by the synchronization tools presented in this paper.

Movies of sample interventions, screenshots, and further descriptions of the different hysteroscopy simulator modules can be found on our project web page (http://www.hystsim.ethz.ch).

Acknowledgment

The authors would like to thank all developers of the hysteroscopy simulator project. This research has been supported by the NCCR Co-Me of the Swiss National Science Foundation.

References

1. OGRE 3D. http://www.ogre3d.org. (visited Mar 2006).
2. SOFA Simulation Open Framework Architecture. http://sofa-framework.org. (visited Mar 2006).
3. M.C. Cavusoglu, T.G. Goktekin, F. Tendick, and S. Sastry. GiPSi: An open source/open architecture software development framework for surgical simulation. In *Medicine Meets Virtual Reality*, pages 46–48, January 2004.
4. Epic Games. Unreal engine 3, http://www.epicgames.com. (visited Mar 2006).
5. M. Harders, D. Steinemann, M. Gross, and G. Szekely. A hybrid cutting approach for hysteroscopy simulation. In *Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI'05)*, pages 567–574, October 2005.
6. id Software. Quake 4 engine, http://www.idsoftware.com. (visited Mar 2006).
7. GiPSi General Interactive Physical Simulation Interface. http://gipsi.case.edu. (visited Mar 2006).
8. Irrlicht. http://irrlicht.sourceforge.net. (visited Mar 2006).
9. J.S. Levy. Virtual reality hysteroscopy. *J Am Assoc Gyn. Laparosc.*, 3:25–26, 1996.
10. Immersion Medical. AccuTouch system, http://www.immersion.com/medical. (visited Mar 2006).
11. K. Montgomery, C. Bruyns, J. Brown, S. Sorkin, F. Mazzela, G. Thonier, A. Tellier, B. Lerman, and A. Menon. Spring: A general framework for collaborative, realtime surgical simulation. In *Medicine Meets Virtual Reality*, pages 23–26, 2002.
12. K. Montgomery, L.R. Heinrichs, C. Bruyns, S. Wildermuth, C. Hasser, S. Ozenne, and D.Bailey. Surgical simulator for hysteroscopy: A case study of visualization in surgical training. In *IEEE Visualization*, pages 449–452, 2001.
13. M. Teschner, B. Heidelberger, M. Mueller, D. Pomeranets, and M. Gross. Optimized spatial hashing for collision detection of deformable objects. In *Proceedings of Vision, Modeling, Visualization VMV03*, pages 47–54, November 2003.

An Introduction to GPU Accelerated Surgical Simulation

Thomas Sangild Sørensen[1] and Jesper Mosegaard[2]

[1] Centre for Advanced Visualisation and Interaction, [2] Department of Computer Science
University of Aarhus, Denmark
sangild@cavi.dk, mosegard@daimi.au.dk

Abstract. Modern graphics processing units (GPUs) have recently become fully programmable. Thus a powerful and cost-efficient new computational platform for surgical simulations has emerged. A broad selection of publications has shown that scientific computations obtain a significant speedup if ported from the CPU to the GPU. To take advantage of the GPU however, one must understand the limitations inherent in its design and devise algorithms accordingly. We have observed that many researchers with experience in surgical simulation find this a significant hurdle to overcome. To facilitate the transition from CPU- to GPU-based simulations, we review the most important concepts and data structures required to realise two popular deformable models on the GPU: the finite element model and the spring-mass model.

1 Introduction

General-purpose computation using graphics hardware (or GPGPU) is a research area that has grown rapidly in recent years. By using the modern graphics card (i.e. the GPU) for computations, many computationally heavy algorithms have been accelerated significantly compared to conventional CPU-based algorithms. This includes most of the techniques currently being applied in surgical simulators. Unfortunately, the GPU is difficult to utilise efficiently. A substantial knowledge of its design, programming model, and limitations is necessary for optimal results. This paper is intended as an introductory article to GPGPU aimed specifically for researchers with experience in surgical simulation, who wish to attempt a GPU implementation of their algorithms. We review the literature introducing the most important concepts, and discuss the hardware limitations we must adhere for optimal results.

An overview of the many applications of GPGPU is best obtained by exploring the online resource [1] and the books in the GPU Gems series [2,3]. Moreover these surveys [4,5] and course material [6,7] highlight some commonly used algorithms. The survey by Strzodka et al [5] has a well-written introduction to scientific computation on the GPU. Several programming languages are available, ranging from a low level machine language [8] to high level C-like languages such as Cg [9] and GLSL [10]. Based on a general data abstraction model for parallel programming, streams, a compiler and run-time system is available [11]. A few getting started tutorials are available here [12]. This paper extends these references with a survey and

M. Harders and G. Székely (Eds.): ISBMS 2006, LNCS 4072, pp. 93 – 104, 2006.

discussion of GPU accelerated techniques aimed specifically towards surgical simulation.

2 GPGPU Concepts and Performance

The standard graphics pipeline in OpenGL and DirectX contains fixed functionality vertex and pixel shaders. A basic knowledge of this pipeline is assumed in the remaining paper, and only a brief review is provided below. More information can be found in e.g. [13]. The vertex shader transforms the geometry (triangles) received from the application from local object space coordinates to window coordinates through a series of transformations. The colour and texture coordinates are also computed for each vertex. The geometric primitive is subsequently *rasterized*, a term that describes the process in which the colour of each pixel in the primitive is computed. The pixel shader is responsible for computing these colours. Based on each pixel's spatial position in the geometric primitive, the pixel shader receives the per-pixel interpolated colour and texture coordinates as input. The fixed function pixel shader then computes the output colour as a function of the input colour and the texture colours. The texture colours are found from texture lookups using the per-pixel input texture coordinates.

In the past generations of GPUs the vertex and pixel shaders have gradually become fully programmable. In GPGPU we utilise this to write pixel shaders that no longer compute colours, but instead the scalars and/or vectors involved in a general computation. Each pixel can store a 4-tuple of floating point values in up to 32 bit precision per entry. We store the computed pixels directly in a non-visible GPU memory buffer as it is no longer meaningful to visualise these pixels directly. This buffer is actually a texture that can be used as input for subsequent iterations of our computations. Hence we have established a computational model in which we can both read from and write to GPU memory. A custom vertex- and a custom pixel shader program are uploaded to the GPU and applied in the subsequent processing of primitives in parallel. Due to this parallel nature of the GPU (the Radeon X1900XT has 48 pixel pipes, a Geforce 7900GTX has 24 pixel pipes), a high throughput can be obtained. A CPU-based physical simulation typically stores data in one-, two-, and three-dimensional arrays. On the GPU, data is stored instead in one-, two-, or three-dimensional textures. As the GPU works most efficiently on two-dimensional data structures, we transform both 1D and 3D textures to 2D textures in practise [14]. Naturally, some bookkeeping is necessary to handle this transformation. When a computation is invoked, the pixel shader receives an input texture coordinate that identifies the spatial position of the corresponding pixel in the input textures. If the algorithm requires access to neighbouring pixels, this is achieved by offsetting the input texture coordinate before looking up in the respective textures. These offsets can be either "global constants" or obtained through an additional texture lookup at the current pixel. As will be explained in section 3, the type of offset depends on the underlying spatial discretization of the computational domain; whether it is structured or unstructured.

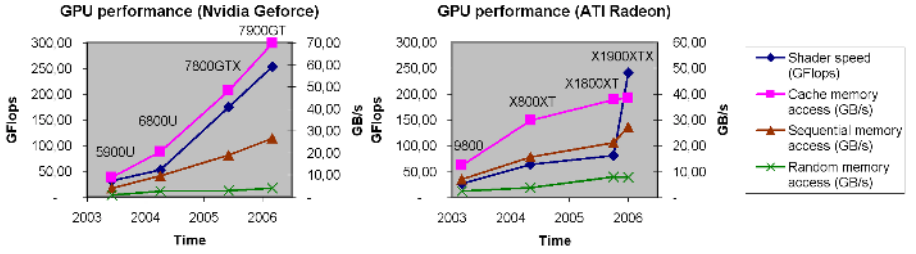


Fig. 1. *Observed* performance from the most recent generations of Nvidia and ATI GPUs. Data was obtained using GPUBench [15]. Blue diamonds represent the shader performance measured in GFlops. Cache, sequential, and random memory access measured in GB/s are depicted in the remaining graphs.

Using the benchmark test suite GPUBench [15] an overview of the performance of a system's GPU can be obtained. **Fig. 1** shows the graphs for the most recent GPUs from Nvidia and ATI. Looking at the number of floating point operations available per second (Flops) it can be observed that the current performance leaders provide roughly 250 GFlops. This number was obtained from the GPUBench test `instrissue`. This test measures the number of MAD instructions that can be executed per second on the present GPU. Since each shader operates on 4-tuples of floating point values and each MAD operation constitutes two floating point operations (a multiplication and an addition), the values reported by `instrissue` are multiplied by 8 for conversion to GFlops. Compared to the *theoretical* peak performance from a state-of-the art CPU (7.4 GFlops / 3.8 GHz Intel Xeon [16]) it should be clear why a GPU implementation of a surgical simulation can potentially boost performance. Discussing potential performance gains merely based on the shader speed reported in **Fig. 1** does not provide a fulfilling picture however, as a typical GPU implementation of a surgical simulation would not be *compute bound*. More likely it would be *memory bound* - meaning that access to GPU memory would be the limiting factor. Consequently, **Fig. 1** contains three graphs showing the observed memory bandwidth on the most recent GPUs. They are based on data obtained running GPUBench's `floatbandwidth` test on the respective GPUs [15]. It can be seen that cache memory access is significantly faster than sequential and

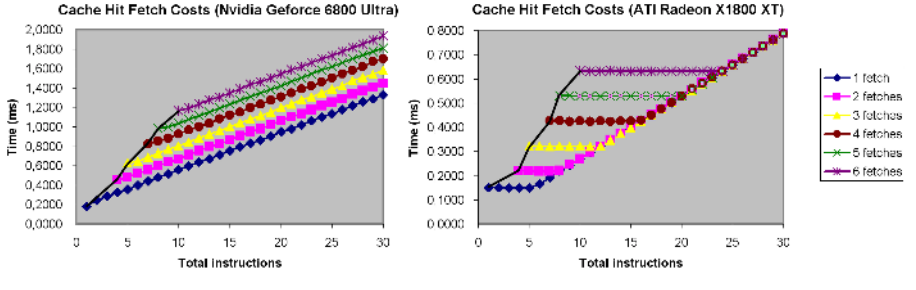


Fig. 2. Cache hit memory access costs as a function of the number of shader instructions. Data was obtained from the most recent generations of Nvidia (left) and ATI (right) GPUs using GPUBench [15]. The number of texture fetches in each test was varied from 1 to 6.

random memory access. The cache memory bandwidth constitutes an upper limit in memory bandwidth, hardly attainable in real-world applications. Depending on the memory coherence of a given application, the growth in memory bandwidth could instead follow the lines depicting the sequential or random memory access bandwidth. Thus, designing memory coherent algorithms is of outmost importance.

To discuss whether a given application is compute or memory bound, the literature (e.g. [3]) defines the *arithmetic intensity* of an application as the "amount of work" that is performed per memory access. Applications with high arithmetic intensity are most likely compute bound while low arithmetic intensity is an indication of a memory bound algorithm. To discuss this issue in more detail, we once again resort to GPUBench: The test `fetchcosts` shows the execution time of a GPU program as a function of the number of instructions. **Fig. 2** shows the results from this test on two GPUs. Note that each test is comprised of six sub-tests that perform one to six memory cache accesses each. We will discuss below the results obtained on an ATI Radeon GPU. A similar (but not entirely identical) discussion can be made for the Nvidia based GPU, but we leave this discussion for the reader to complete. First notice the horizontal line segments in the rightmost half of **Fig. 2**. They show that for each memory access, a number of "free" computations can be made without influencing the overall execution time. Only as the non-horizontal (diagonal) part of the graph is reached, there is a cost associated to issuing additional instructions. From the figure we can predict the execution time of an application consisting entirely of memory reads (solid black line). Notice that the slope of this line is much steeper than the slope of the diagonal. The diagonal constitutes the border between a memory bound and a truly compute bound application: An application with an arithmetic intensity that places it between the leftmost solid line and the diagonal is memory bound, while an application with an arithmetic intensity that places it on (or close to) the diagonal would be compute bound. As we shall see in the subsequent sections, surgical simulation algorithms implemented on the GPU are most likely memory bound, as the complexity in algebraic operations per memory access is limited. Experiments show however, that these GPU-based algorithms still significantly outperform their CPU-based counterparts.

3 Surgical Simulation on the GPU

Many computational models for deformable surfaces have been proposed in the existing literature. We refer to the surveys [17,18] for a detailed overview. We limit our description of GPU-based techniques to mesh-based deformable models (most often a mesh of triangles, tetrahedrons, or cubes) as this is the preferred approach in real-time surgical simulators that must handle arbitrary incisions and general changes in topology. For the remaining paper we refer to *nodes* as the discretized points defining a mesh. We present an overview of the required GPU-based techniques to implement the most common deformable models: finite element models and spring-mass models. The reader should subsequently be able to define custom modifications to these general models in GPU terms. The implicit linear elastic finite element

models presented by Bro-Nielsen et al in [19] are discussed in section 3.1. The explicit finite element model (tensor-mass model) presented by Cotin et al in [20] and the explicit spring-mass model (e.g. [20,21]) are discussed in section 3.2. Szekely et al used a cluster of processors in [22] to realise a laparoscopic surgery simulator. Many of the general considerations on the design of parallel algorithms for numerical computations on multiple CPUs transfer directly to the parallel processor we introduce in this paper, namely the GPU.

3.1 Implicit Finite Element Models

Using the notation from [19], finding the deformations in the implicit linear elastic finite element model reduces to solving either a static system on the form $\mathbf{Ku} = \mathbf{f}$ or a dynamic system on the form $\mathbf{M\ddot{u}} + \mathbf{C\dot{u}} + \mathbf{Ku} = \mathbf{f}$, where $\mathbf{M}$ and $\mathbf{C}$ are diagonal mass and damping matrices, $\mathbf{K}$ is a symmetric positive definite matrix representing the topology and stiffness of the discretized mass points, and $\mathbf{u}$ and $\mathbf{f}$ are the deformation and external force vectors respectively. No matter the choice of system, it can be rewritten on the form $\tilde{\mathbf{K}}\tilde{\mathbf{u}} = \tilde{\mathbf{f}}$ following a finite difference time discretization for the dynamic system [19]. If we let n denote the total number of mass point in the finite element discretization, $\tilde{\mathbf{K}}$ has dimensions $3n$ x $3n$ and entry k_{ij} encodes the connectivity and stiffness between nodes i and j. In its most elementary form $\tilde{\mathbf{K}}$ is sparse having non-zero entries only between connected mass points. Depending on the choice of spatial discretization, it can either be structured (banded) or unstructured. **Fig. 3** (left) illustrates a spatial discretization that leads to a banded matrix. Boxes (possibly consisting of six tetrahedra of fixed topology) are used as the basic spatial building blocks in a regular three-dimensional grid. The rightmost tetrahedralisation in **Fig. 3** on the other hand, leads to an unstructured sparse matrix. Finally, using the condensation technique described in [19], $\tilde{\mathbf{K}}$ can be transformed to a smaller *dense* matrix of boundary nodes. We have distinguished between the different layouts of $\tilde{\mathbf{K}}$ since they each call for their own distinct representation on the GPU. The three matrix layouts are 1) sparse (banded), 2) sparse (unstructured), and 3) dense. Solving the linear system of equations involves matrix and vector algebra. We discuss common linear algebra operations below for the different representation of $\tilde{\mathbf{K}}$. This is followed by a discussion on GPU-based solutions to the linear system.

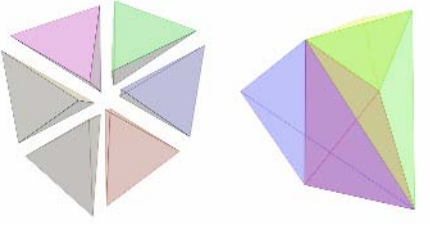


Fig. 3. Structured (left) and unstructured (right) tetrahedral meshes. The choice of tetrahedralisation influences the layout of the resulting stiffness matrix in a linear elastic finite element model.

The first formulation of a general framework for numerical algebra on the GPU was published by Thompson et al in [23] in a very "machine-near" language. Inspired by the BLAS and LAPACK libraries [24,25], Krueger et al subsequently published their initial work on a GPU-based counterpart [3,26].

Fig. 4 (top) shows their representation of ***sparse banded matrices***. Each band in an n x n–dimensional matrix **A** can be seen as a one-dimensional vector of length n. As explained in section 2, we convert this vector to a two-dimensional texture on the GPU. Similarly, vectors **b** and **x** of dimension n are stored in textures of identical dimensions to the bands of **A**. In many computations it is necessary to find the matrix-vector multiplication **x=Ab.** To achieve this we render a quad covering output texture **x** in multiple passes – one rendering pass for each band in **A**. **Fig. 4** (bottom) illustrates the three passes required for a tri-banded matrix-vector multiplication with this representation. In each pass the values in corresponding pixels in textures **A** and **b** are multiplied and added to **x**. In each pass (except the pass using the matrix diagonal) the texture coordinates used to look up pixels in **b** are shifted to account for the corresponding band position. Several optimisations to this basic scheme is possible and discussed in [3,26]. Since it is not possible to read from and write to the same texture during a pass, the accumulative writes to texture **x** must be implemented through two textures, one of which is bound as input and the other for writing in an alternating fashion. It is important to realise the parallel nature of the algorithm: For each of the three passes in the example in **Fig. 4**, the n entries in the result vector **x** are computed simultaneously, providing a significant speedup to CPU-based matrix-vector multiplications given a sufficient number of pixel pipes and texture memory bandwidth.

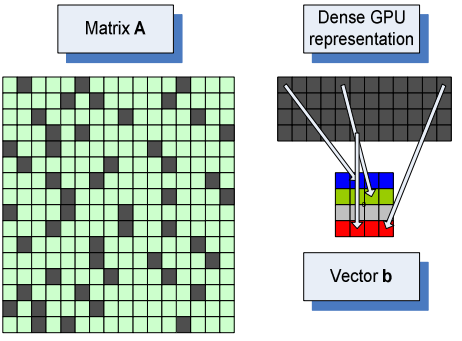


Fig. 5. Unstructured sparse matrix representation specialized from [27]. We assume exactly three non-zero entries per row in this example (grey). These values are stored in a "dense texture". For each pixel in this texture a pointer (i.e. texture coordinate) to the corresponding entry in vector **b** is stored (arrows).

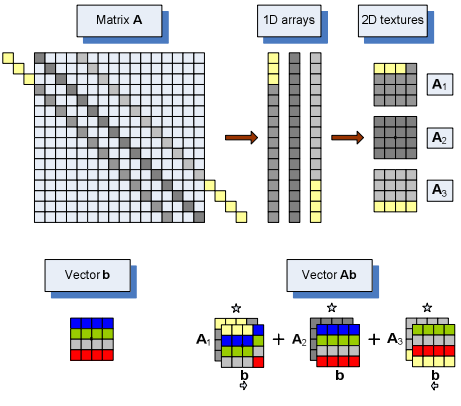


Fig. 4. GPU representation of a sparse banded matrix **A**, a vector **b**, and the corresponding matrix-vector multiplication (adapted from [26]). Top: Each band represents a one-dimensional array which is stored in a 2D texture on the GPU ($\mathbf{A}_1$-$\mathbf{A}_3$). Zeroes are prepended or appended depending on the position of the band in the matrix. Bottom: A vector **b** is defined and multiplied to **A.** Each band in **A** is multiplied with **b** pixelwise. The products are added to form **Ab**. Notice that the texture coordinates used to access **b** are offset corresponding to each band.

Sparse unstructured matrices are handled differently from the banded matrices above. Krueger et al [3,26] renders point based primitives to

implement such matrix-vector multiplications. We will instead describe a different approach using texture pointers however, as this relates nicely to this section's subsequent discussion of algorithm design that minimises memory bandwidth. We illustrate a specialisation of the general sparse unstructured matrix-vector multiplication by Bolz et al [27] to find the product **x=Ab**. We assume that a constant number of non-zero entries exist in each row of the sparse *n* x *n* dimensional matrix **A** of. In **Fig. 5** we use only three entries per row to reduce the size of the figure. We create a one-dimensional array of length *2·3·n* to represent **A**. *3n* entries are necessary to store the non-zeroes values in **A**. Furthermore, for each value we additionally store a pointer to the corresponding entry in textures **b**. As always, we represent this one-dimensional array as a two-dimensional texture on the GPU. We render a quad covering our output texture **x** to initiate the parallel computation of **Ab**. For each pixel we look up the three non-zero values in the corresponding row in the texture representation of **A** from the input texture coordinate. The texture representation of **A** furthermore provides us with three pointers (texture coordinates) that are used to look up the values in **b** corresponding to the non-zero entries in **A**. The results from the three multiplications are added and stored in **x**. Again, it is important that the reader recognizes the parallel nature of the algorithm, in which each entry in **x** is computedin parallel.

With the understanding of sparse matrix representations on the GPU, the reader should have the prerequisites to derive representations of ***dense matrices*** as these are more straightforward than those of sparse matrices. We refer to [3,28,29] for completeness.

We now return to solving the linear system $\tilde{\mathbf{K}}\tilde{\mathbf{u}} = \tilde{\mathbf{f}}$ that was defined initially in this section. As $\tilde{\mathbf{K}}$ is symmetric and positive definite, one approach to finding $\tilde{\mathbf{u}}$ is through the conjugate gradient algorithm. Using their respective frameworks for linear algebra, both Krueger et al and Bolz et al showed how to implement the conjugate gradient algorithm on the GPU in [26,27]. An alternative approach guaranteed to converge to the right solution for arbitrary starting configurations is the Gauss-Seidel iterative process (again since $\tilde{\mathbf{K}}$ is symmetric and positive definite). Contained in [26] is a short section discussing the implementation of this algorithm on the GPU. Closely related to Gauss-Seidel's method is the Jacobi method. In contrast to Gauss-Seidel's methods, Jacobi's method is ideally suited for a parallel implementation. For this reason it has been used intensely in previous GPGPU publications, e.g. in several chapters in [2,3], and in [30,31].

The representation of banded matrices as shown in **Fig. 4** results in a minimum number of texture fetches: only the actual values needed for a matrix-vector multiplication are read from texture memory. The unstructured sparse matrix representation on the other hand requires further texture lookups to perform a matrix-vector multiplication, as pointers are stored in textures and the corresponding values only obtained through an additional texture lookup. Looking back at the discussion related to **Fig. 2** it should be clear why the banded representation performs better than the unstructured alternative. It is simply due to the lower number of texture fetches involved. Consider also the limited number of algebraic operations performed per memory access in both approaches (that is the arithmetic intensity is low). Given the memory bandwidth on current GPUs both matrix representations are memory bound, although positioned differently in the graphs in **Fig. 2**. It was Fatahalian et al who

initially reported that memory access is indeed the limiting factor for dense matrix-matrix multiplications on the GPU [28].

The reader is encouraged to examine the GPU-based surgical simulator by Wu et al [32]. They used an implicit finite element solver through the conjugate gradient algorithm and obtained a two-fold acceleration. This work was done on an Nvidia Geforce 5950 Ultra however. As is clear from **Fig. 1**, both the GPU speed and memory bandwidth have increased five- to ten-fold since on the most recent GPUs.

3.2 Explicit Models: Spring-Mass and Tensor-Mass Models

We return to the general equation of Newtonian motion: $\mathbf{M\ddot{u}} + \mathbf{C\dot{u}} + \mathbf{Ku} = \mathbf{f}$ [19,20]. We now seek to solve the system of differential equations through an explicit time integration scheme rather than by the implicit method discussed in section 3.1. A particularly well suited explicit integration scheme is Verlet integration [33], a scheme in which the position of each mesh node for the subsequent time step is calculated from its positions in the two previous iterations and from an elastic force vector (acceleration vector). No additional information, e.g. velocities, needs to be stored and calculated. The force vector is calculated locally from each node's connectivity in the mesh. We denote the force vector corresponding to a spring-mass system as $\hat{\mathbf{F}}_i$ and the force vector relating to the tensor-mass system as $\tilde{\mathbf{F}}_i$. Using the notation from [20] these forces are then defined as

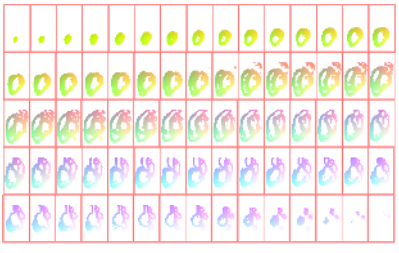


Fig. 6. Position texture, inspired from [35]. A regular 3D grid of nodes is mapped to a 2D texture. The colors of the individual pixels denote the corresponding particle's position.

$$\hat{\mathbf{F}}_i = \sum_{j \in N(\mathbf{P}_i)} k_{ij}\left(\left\|\mathbf{P}_i\mathbf{P}_j\right\| - l_{ij}^0\right)\frac{\mathbf{P}_i\mathbf{P}_j}{\left\|\mathbf{P}_i\mathbf{P}_j\right\|} \quad \text{and} \quad \tilde{\mathbf{F}}_i = \mathbf{K}_{ii}\,\mathbf{P}_i^0\mathbf{P}^i + \sum_{j \in N(\mathbf{P}_i)} \mathbf{K}_{ij}\,\mathbf{P}_j^0\mathbf{P}^j$$

where $\mathbf{P}_i$ denotes the position of node i, $\mathbf{P}_i^0$ is the initial (undeformed) position of node i, $\mathbf{P}_i\ \mathbf{P}_j$ is the vector between nodes i and j, l_{ij} and k_{ij} is the spring rest length and stiffness respectively between nodes i and j, and $\mathbf{K}$ is the rigidity (or stiffness) matrix of the linear elastic finite element model.

A two-dimensional GPU- and spring-mass based cloth simulation using Verlet integration was presented in [34]. It is sufficiently simple to be recommended for inexperienced GPU programmers. Mosegaard et al presented a three-dimensional, volumetric spring-mass based surgical simulator implemented on the GPU in [35]. Their paper compares two spring-mass implementations, one in which nodes were confined to a regular three-dimensional grid, and one in which node positions were unrestricted and springs explicitly represented in a *connectivity texture*. Overall, a twenty-fold acceleration over a similar CPU-based system was achieved for the first method, while the latter achieved a ten-fold acceleration. Their results were obtained on a Geforce 6800 Ultra. It is clear from **Fig. 1** that both the shader speed and memory bandwidth have increased significantly since and even better results could be

obtained on the most recent generations of GPUs. In the faster of the two methods they use a *position texture* to store the positions of the nodes in the spring-mass system. An example one such texture is depicted in **Fig. 6**. To initialise parallel computation of each time-step, a quad at the size of this texture is rendered in the output buffer. A depth test is used to prevent that calculations are wasted on the white (void) particles. The forces $\hat{\mathbf{F}}_\mathbf{i}$ are computed for each pixel (node): The two most recent position textures are provided as input textures to the pixel shader. By adding to the input texture coordinate the fixed offset to each neighbour, each neighbouring node's position can be looked up. As the nodes are restricted to a regular grid, the individual spring rest lengths are known and need not be looked up. Furthermore, the spring stiffness is also kept constant. Thus, the only texture fetches involved are those used to obtain the connected nodes' positions. I.e. they use only the minimal number of texture fetches. This is important in the light of the discussion concerning the cost of texture fetches (**Fig. 2**). Their alternative approach uses texture lookups to fetch the texture coordinate of each neighbour. This doubles the overall number of texture fetches, consequently reducing the simulation rate by a factor of two. Replacing the spring induced forces $\hat{\mathbf{F}}_i$ with the tensor-mass forces $\tilde{\mathbf{F}}_i$ would instead solve an explicitly formulated finite element model. For each connected node we need then an additional texture lookup in the stiffness matrix to obtain $\mathbf{K}_{ij}$. Consequently we can expect the tensor-mass model to run at half the speed of the spring-mass model. Compared to a CPU-based implementation however, it is still significantly faster. The most recent performance measurements are found in Sørensen et al [36], who report simulation rates exceeding 1 kHz using a Geforce 7800 GTX on a spring-mass system consisting of 20.000 nodes connected with 18 neighbours each in a regular volumetric mesh (grid).

The methods by Mosegaard et al are simple to implement and run fast but use a significant amount of texture memory: The position texture approach wastes memory representing void particles, while the connectivity texture approach allocates memory for a constant number of neighbours per node wasting texture memory if the number of neighbours varies significantly throughout the mesh. To conserve memory, Georgii et al used a stack of *valence textures* to encode different levels of connectivity in [37]. Unfortunately the algorithm reduces performance as well due to a much more complicated rendering scheme. This led Georgii et al to develop an edge-centric data structure instead that "iterates" over springs rather than nodes [38]. This reduces the arithmetic intensity of their algorithm since each spring force is now computed only once whereas they are computed twice in the previous methods [35,37]. As all the presented algorithms are most likely memory bound (**Fig. 2**) this is not a major advantage however. We are more interested in examining the number of texture fetches used in the edge-centric approach. On the regular mesh shown in **Fig. 6** we experienced that the number of texture fetches involved in the edge-centric approach versus the position texture approach are almost identical. The edge-centric approach currently runs only in 8 bit or 16 bit precision as a necessary blending operation is not supported in 32 bit precision on any GPUs yet. Also, the vertex processor is used intensely, a potential bottleneck. It will be up to the individual application to weigh the advantage of reduced memory consumption versus these precision and vertex processing issues.

3.3 Visualisation and Interaction

Depending on the chosen simulation model, the result of each time step is either a texture of node deformations (implicit model) or a texture of node positions (explicit model). In either case a deformed surface triangle-mesh of the modelled organ can be visualised from a static display list of the initial mesh configuration through a dedicated vertex shader [35]. For each vertex in the visualised surface mesh, the application provides the required texture coordinate to look up the corresponding deformation vector or particle position. The vertex shader can thus compute the deformed vertex position for the current time step.

From both section 3.1 and 3.2 it is clear that a structured spatial discretization of the simulated volume results in the fastest algorithms due to a minimum number of texture lookups required in each time step. This can however result in a jagged (stair-like) look of the modelled morphology as illustrated in **Fig. 7**. To overcome this problem Mosegaard et al proposed in [39] to fully decouple surface visualisation from the underlying volumetric simulation. They represent each vertex on the surface model by an offset from the nearest node in the simulation mesh. **Fig. 7** shows a smooth surface drawn through this method. The offset vectors are expressed in the tangent-space of the surface, and the surface thus correctly deforms based on the deformation of the associated nodes in the simulation mesh.

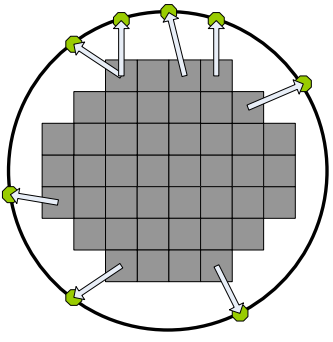


Fig. 7. Surface visualisation (circle) de-coupled from a volumetric simulation of a sphere discretized to a regular grid (grey). The green circles represent vertices on the surface mesh that can be sampled at any resolution. Each vertex is represented by an offset vector (arrow) from the nearest simulation node.

Interaction with a GPU-based surgical simulator is the final issue to be discussed in this paper. The overall question is whether to resolve user interaction on the CPU or on the GPU. If the CPU is chosen one must be careful not to transfer large amounts of data from the GPU to the CPU in each frame, as this would introduce a performance bottleneck. Consequently, interaction that involves computations on the current state of the simulation is probably best implemented on the GPU. Peripheral devices can only be communicated with through the CPU however, so a minimum amount of per-frame data transfer cannot always be avoided. Sørensen et al showed in [36] how to implement force feedback from a GPU-based simulator with limited performance penalty. Several groups have recently published algorithms for GPU accelerated collision detection [40-42].

Acknowledgements

We kindly acknowledge the funding we received from the Danish Research Council's Program Committee on IT-Research (grant #2059-03-0004).

References

1. General-Purpose Computation Using Graphics Hardware. http://www.gpgpu.org/.
2. Fernando, R. GPU Gems, Part VI. Addison-Wesley 2004.
3. Pharr, M. GPU Gems 2, Part IV-VI. Addison-Wesley 2005.
4. Owens, J.D., Luebke, D., Govindaraju, N., Harris, M., Krüger, J., Lefohn, A.E., and Purcell, T.J. A Survey of General-Purpose Computation on Graphics Hardware. State of the Art Reports, Eurographics 2005:21-51.
5. Strzodka, R., Doggett, M., and Kolb, A. Scientific Computation for Simulations on Programmable Graphics Hardware. Simulation Modelling Practice and Theory 2005;13(8):667-681.
6. GPGPU course, Siggraph 05. http://www.gpgpu.org/s2005/.
7. IEEE Visualization 2005 tutorial. http://www.gpgpu.org/vis2005/.
8. OpenGL extensions specifications. ARB_vertex_program and ARB_fragment_program. http://oss.sgi.com/projects/ogl-sample/registry/index.html.
9. Fernando, R. and Kilgard, M.J. The Cg Tutorial. Addison-Wesley 2003.
10. Rost, R.J. OpenGL Shading Language. Addison-Wesley 2004.
11. Buck, I., Foley, T., Horn, D., Sugerman, J., Fatahalian, K., and Hanrahan, P. Brook for GPUs: Stream Computing on Graphics Hardware. ACM Transactions on Graphics, Siggraph 2003;23(3):777-786.
12. Göddeke, D. GPGPU::Tutorial. http://www.mathematik.uni-dortmund.de/~goeddeke/gpgpu/index.html.
13. Möller, T.A. and Haines, E. Real-Time Rendering. A K Peters 2002.
14. Harris, M.J., Baxter III, V., Scheuermann, T., and Lastra, E. Simulation of Cloud Dynamics on Graphics Hardware. Siggraph/Eurographics Workshop On Graphics Hardware 2003:92-101.
15. Buck, I., Fatahalian, K., and Hanrahan, P. GPUBench: Evaluating GPU performance for Numerical and Scientific Application. General Purpose Computing on Graphics Processors, ACM Workshop 2004:C-20.
16. Dongarra, J. The Linpack Benchmark Report. http://www.netlib.org/benchmark/performance.ps.
17. Montagnat, J., Delingette, H., and Ayache, N. A review of deformable surfaces: topology, geometry and deformation. Image and Vision Computing 2001;19(14):1023-1040.
18. Gibson, S.F.F. and Mirtich, B. A Survey of Derformable Modeling in Computer Graphics. Technical Report, Mitsubishi Electric Research Lab.
19. Bro-Nielsen, M. and Cotin, S. Real-time Volumetric Deformable Models for Surgeyr Simulation using Finite Elements and Condensation. Computer Grahics Forum, Eurographics 1996;15:57-66.
20. Cotin, S., Delingette, H., and Ayache, N. A Hybrid Elastic Model Allowing Real-Time Cutting, Deformations and Force-Feedback for surgery Training and Simulation. The Visual Computer 2000;16(8):437-452.
21. Liu, A., Tendick, F., Cleary, K., and Kaufmann, C. A Survey of Surgical Simulation: Applications, Technology, and Education. Presence 2003;12(6):599-614.
22. Szekely, G., Brechbuhler, C., Hutter, R., Rhomberg, A., and Schmid, P. Modelling of Soft Tissue Deformation for Laparoscopic Surgery Simulation. Medical Image Analysis 2000;4:57-66.
23. Thompson, C., Hahn, C., and Oskin, M. Using Modern Graphics Architectures for General-Purpose Computing: A Framework and Analysis, 35th IEEE/ACM International Symposium on Micro Architecture 2002:306-3.

24. Dongarra, J., Du Croz, J., Hammarling, S., and Hanson, R. An extended set of FORTRAN basic linear algebra subprograms. ACM Transactions on Mathematical Software 1988;14:1-17.
25. Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A., and Sorensen, D. LAPACK Users' Guide. Society for Industrial and Applied Mathematics 1999.
26. Krueger, J. and Westermann, R. Linear algebra operators for GPU implementation of numerical algorithms. ACM Transactions on Graphics, Siggraph 2003;22(3):908-916.
27. Bolz, J., Farmer, I., Grinspun, E., and Schröder, P. Sparse matrix solvers on the GPU: conjugate gradients and multigrid. ACM Transactions on Graphics, Siggraph 2003;22(3):917-924.
28. Fatahalian, K., Sugerman, J., and Hanrahan, P. Understanding the efficiency of GPU algorithms for matrix-matrix multiplication. ACM Siggraph/Eurographics Conference on Graphics Hardware 2004:133-137.
29. Galoppo, N., Govindaraju, N.K., Henson, M., and Manocha, D. LU-GPU: Efficient Algorithms for Solving Dense Linear Systems on Graphics Hardware. ACM/IEEE conference on Supercomputing 2005:3.
30. Goodnight, N., Woolley, C., Lewin, G., Luebke, D., and Humphreys, G. A Multigrid Solver for Boundary Value Problems Using Programmable Graphics Hardware. ACM Siggraph/Eurographics Conference on Graphics Hardware 2003:102-111.
31. Rumpf, M. and Strzodka, R. Using Graphics Cards for Quantized FEM Computations. IASTED Visualization, Imaging and Image Processing 2001:193-202.
32. Wu, W. and Heng, P.A. A hybrid condensed finite element model with GPU acceleration for interactive 3D soft tissue cutting. Computer Animation and Virtual Worlds 2004;15:219-227.
33. Dummer, J. A Simple Time-Corrected Verlet Integration Method. http://www.gamedev.net/reference/programming/features/verlet/.
34. Green, S. Cloth Simulation on the GPU. http://developer.nvidia.com/object/demo_cloth_simulation.html.
35. Mosegaard, J. and Sørensen, T.S. GPU accelerated surgical simulators for Complex Morphology. IEEE Virtual Reality 2005:147-153.
36. Sørensen, T.S. and Mosegaard, J. Haptic Feedback for the GPU-based Surgical Simulator. Studies in Health Technology and Informatics, 14th Medicine Meets Virtual Reality 2006;119:523-528.
37. Georgii, J., Echtler, F., and Westermann, R. Interactive Simulation of Deformable Bodies on GPUs. Simulation and Visualisation 2005:247-258.
38. Georgii, J. and Westermann, R. Mass-Spring Systems on the GPU. Simulation Modelling Practice and Theory 2005;13(8):693-702.
39. Mosegaard, J. and Sørensen, T.S. Real-time Deformation of Detailed Geometry Based on Mappings to a Less Detailed Physical Simulation on the GPU. Immersive Projection Technology & Eurographics Virtual Environments Workshop 2005:105-110.
40. Govindaraju, N.K., Lin, M., and Manocha, D. Quick-CULLIDE: Fast Inter- and Intra-Object Collision Culling using Graphics Hardware. IEEE Virtual Reality 2005:59-66.
41. Wong, W.S.-K. and Baciu, G. GPU-based intrinsic collision detection for deformable surfaces. Computer Animation and Virtual Worlds 2005;16:153-161.
42. Choi, Y.-J., Kim, Y.J., and Kim, M.-H. Rapid pairwise intersection tests using programmable GPUs. The Visual Computer 2006;22:80-89

Simulation of Cardiac Activation Patterns for Checking Suggestions About the Suitability of Multi-lead ECG Electrode Arrays

Christoph Hintermüller, Michael Seger, Bernhard Pfeifer,
Gerald Fischer, and Bernhard Tilg

Institute for Biomedical Engineering,
University for Health Sciences, Medical Informatics and Technology,
Eduard Wallnöfer Zentrum I, 6060 Hall in Tyrol, Austria
Telephone: 0043/50/8648/3820; Fax: 0043/50/8648/3850

Abstract. In this study results gained for different electrode array schemes are compared which are used to image the activation time of the heart in a noninvasive way. The tested arrays have been selected based on sensitivity and effort-gain analysis. A cellular automaton was used for generating 6 different patterns resembling a sinus rhythm overlaid by an accessory pathway. The BSP was computed using a finite element approach. For generating the AT maps a boundary element model was used. It was found that methods for noninvasive imaging of the cardiac electrophysiology can profit from the increased details and features contained in the BSP maps recorded by a 125 lead array.

1 Introduction

Methods to image the electrical activation of the human heart in a noninvasive way, have been developed the past 15 years. One of the main approaches is activation time (AT) imaging [1,2,3,4]. The volume conductor models required were extracted from patient individual magnetic resonance imaging (MRI) scans recorded prior to the clinical treatment the patient had to undergo. As depicted in Fig. 1 it basically consists of four different compartments, the chest, the left and right lungs, the ventricular and atrial myocardium and the associated bloodmasses. Further, it contains information upon the electric conductivity of the different compartments and the positions of the electrodes the body surface potential (BSP) is recorded with. Especially locating the posterior electrodes is a rather tedious procedure dependent upon the applied electrode array scheme used for AT imaging as discussed in [5]. Their position was reconstructed from the MRI scans by locating and identifying vitamin E marker capsules. Whereas the positions of the frontal ones may be recorded using a digital position tracking systems such as FASTRACK (Polhemus Colchester, USA). Recording the posterior electrodes with the FASTRACK system would require the patient to sit upright which usually is not possible.

In order to gain precise AT maps it is necessary that the information content of the BSP recorded by the multi-lead electrode array is as high as possible.

M. Harders and G. Székely (Eds.): ISBMS 2006, LNCS 4072, pp. 105–112, 2006.

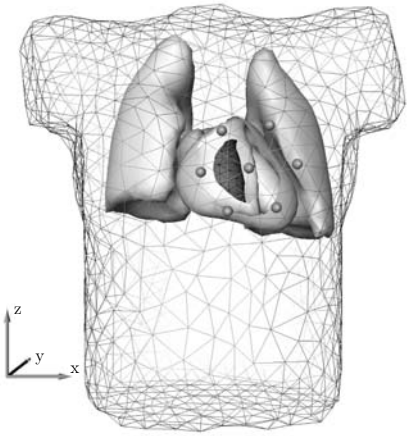


Fig. 1. Volume conductor model consisting of the atria, the ventricles, the corresponding blood masses, the lungs and the chest. The right ventricle is cut open to show the corresponding blood mass.

For the presented study this information is related to the electric activation of the heart. The AT maps gained as a solution to the electrocardiographic inverse problem for different cardiac patterns and patients were used to compare several electrode array schemes. It is advisable to preselect the schemes compared using the methods and criteria described in [5,6,7,8].

The arrays tested are derived from array scheme S_A described in [9] or from S_B described in [5]. The electrode arrays are labeled according to their base scheme indexed by the number of electrodes used.

2 Methods

For solving the forward and inverse problem a patient individual volume conductor model is used. Such a model consists of the patients torso, lungs, ventricles, atria and associated blood masses and the location of the electrodes. The volume conductor includes the compartment conductivities κ and κ_H for the heart. Using discrete boundaries with the interpolation coefficients n_i, $n_{j,H}$ and the potentials $V_i(t)$, $\varphi_{m,j}(t)$ and applying the Wilson Terminal formulation as done by [10] the relation between the transmembrane potential (TMP) φ_m on the heart surface and the BSP V measured by the electrodes can be expressed as

$$V_e(t) = \mathbf{IN}\varphi_m(t) = \mathbf{L}\varphi_m(t) \ . \tag{1}$$

The leadfield matrix $\mathbf{L}$ in (1) consists of the interpolation matrix $\mathbf{I}$ interpolating the discrete BSP to gain the electrode potential and the node field matrix $\mathbf{N}$ describing the patients volume conductor.

2.1 Finite Element Method (FEM)

Another approach to describe the volumeconductor is in terms of finite elements (FEM) instead of boundary elements. For the FEM approach the integral equation for the potentials φ_{H} and V is computed in the and not only on the boundaries of the compartments. This allows to consider anisotropic characteristics and structures in each of the compartments. This anisotropies can be expressed by location dependent conductivities $\kappa(r)$ and $\kappa_{\mathrm{H}}(r)$. The resulting matrix equation between the TMP in the heart and the BSP recordable in the lead positions is given by

$$V_{\mathrm{e}}(t) = \mathbf{I}\tilde{\mathbf{R}}^{-1}\mathbf{S}\varphi_{\mathrm{m}}(t) = \mathbf{L}\varphi_{\mathrm{m}}(t)\ . \tag{2}$$

Similar to (1) $\mathbf{I}$ interpolates the lead space potentials from the potential in 3D space. The matrices $\tilde{\mathbf{R}}$, $\mathbf{S}$ reflect the properties of the passive and active compartments of the volume conductor. A detailed description of the FEM approach (2) , the corresponding equations and its relation to the boundary element formulation (1) can be found in [11].

2.2 Cellular Automaton

In order to verify the findings made by local linear dependency (LLD) and effort gain (EG) analysis in [5] and in order to show the gain in resolution of AT imaging, cellular automaton models for the patients hearts were constructed. The ventricular model (Fig. 2(a)) incorporated the bundle of His (BH), the bundle branches (BB), the purkinje fibers and the ventricular myocardium (MC). Further, it includes a model of the heart fibers (Fig. 2(b)). Also it is necessary to specify the pacing sequence by defining the corresponding pacing sites (Fig. 2(c)).

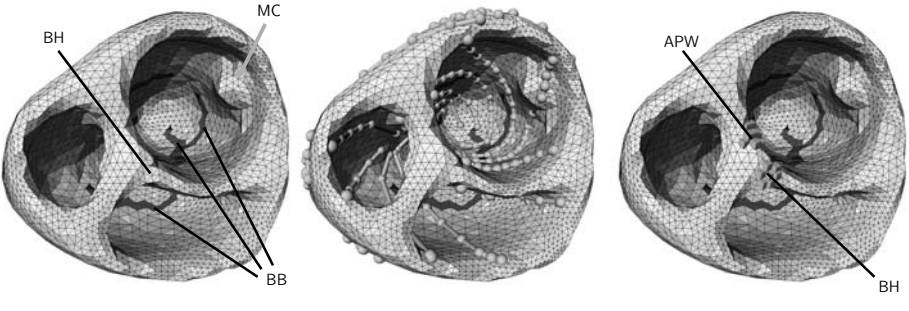


Fig. 2. Cellular automaton: (a) step 1: modeling of the conduction system and myocard characteristics by different tissue materials. Shown are the bundle of HIS (BH), myocard (MC) and bundle branches (BB); (b) step 2: defining fiber directions; (c) step 3: defining pacing sites for sinus rhythm (in BH) and an accessory pathway (APW) modeling a septal WPW.

Detailed information upon the cellular automaton and the corresponding models used to compute several AT and TMP patterns can be found in [12]. The resulting TMP maps were then used to solve the electrocardiographic forward problem.

2.3 The Inverse Problem

The BSP measured by the leads of the entire electrode array were then used to solve the electrocardiographic inverse problem applying an iterative conjugated gradient based solver as described in [13]. This solver is based on a sequential multistage minimisation process. Each stage i resembles a complete optimisation, respective minimisation problem of the form:

$$\min\{\|\mathbf{V} - \mathbf{L}\boldsymbol{\varphi}(\tau)\|^2 + \lambda_i^2 \|\boldsymbol{\Delta}\tau\|^2\} \; . \tag{3}$$

minimizing the sum of the residual norm between the measured lead potential $\mathbf{V}$, the forward solution $\mathbf{L}\boldsymbol{\varphi}(\tau)$ and a regularization term. The matrix $\boldsymbol{\varphi}(\tau)$ describes the action potential (AP) onset for each source point dependent upon its activation time τ. The Laplace operator $\boldsymbol{\Delta}$ accounts for the fact that a source point can only be activated by its neighbours.

3 Results

In order to verify the findings about the optimal electrode array layout made in [5] 6 different ventricular excitation patterns were simulated. Cellular automaton models were generated for the 11 patients (2 female, 9 male). The TMP maps have been calculated for sinus rhythm and Wolff Parkinson White syndrome (WPW) located left posterior, lateral and anterior, right lateral and on the septum near the atrio ventricular node and the bundle of His. The resulting TMP patterns were then used to solve the forward problem, resulting in the corresponding BSP maps.

In a next step random noise with a maximum peak to peak amplitude of $\sigma = 20\,\mu$V was added to the simulated ECG signals to mimic clinical conditions. The noisy signals were used to solve the electrocardiographic inverse problem by estimating the AT maps of the ventricles. In a final step the maps were compared to the AT maps generated by the cellular automaton. Figure 3 depicts the BSP patterns and the estimated AT sequence for a WPW located on the left posterior wall.

To verify that $S_{B,125S}$ produces the best results the computed BSP pattern maps (Fig. 3(d) - 3(f)) for the 6 WPW patterns were compared to the corresponding BSP (Fig. 3(b)). Figure 3 depicts this comparison for a left anterior WPW (Fig. 3(a)). A difference in the details captured by the different arrays can be observed. The pattern computed for $S_{A,62}$ is rather smooth showing only the major features of the BSP. More detailed features are present in the map from $S_{B,62}$ The most can be found in the one from $S_{B,125S}$. As the BSP maps for the remaining arrays are similar to the ones for $S_{A,62}$, $S_{B,62}$ and $S_{B,125S}$ they are not mentioned here in detail.

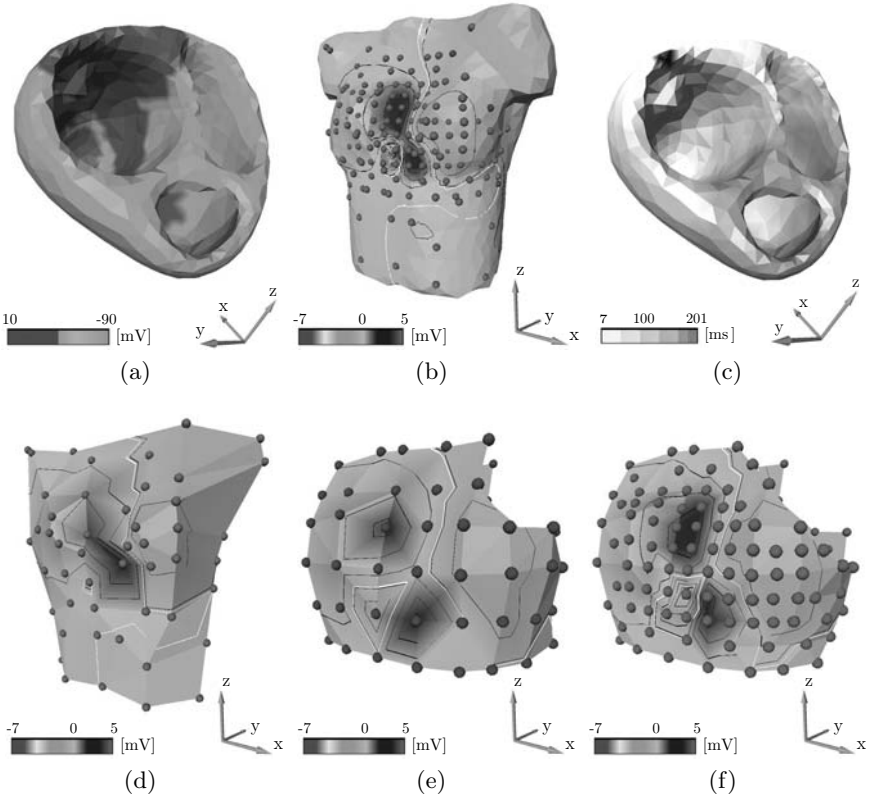


Fig. 3. Transmembrane potential pattern map generated by the cellular automaton (a) simulating a left anterior accessory pathway and the corresponding activation time pattern (c) computed from the BSP recorded by $S_{B,125S}$. The maps for the different arrays $S_{A,62}$ (d), $S_{B,62}$ (e) and $S_{B,125S}$ (f) capture the quadrupole pattern of BSP (b) at different levels of detail dependent upon the location and density of their electrodes. The maps (a), (b), (d), (e) and (f) were captured at 73 ms.

In a final test it was checked if the increased feature detail in the BSP maps from $S_{B,125S}$ improves the results for AT imaging. This was done by comparing the AT distributions found for the inverse problems corresponding to the 6 different simulated WPW patterns. Figure 4 shows them for a left posterior WPW pattern. Dependent upon the body shape and size of the patient the accessory partway could be located near the marked pacing site as for patient 2 (Fig. 4(e), (f)) or at some distance as for patient 1 (Fig. 4(b), (c)). In case of patient 1 the distance between the estimated location and the pacing cite is shorter for $S_{B,125S}$ than for $S_{B,62}$. The AT maps (Fig. 4(a), (d)) computed from the BSP recorded with $S_{A,62}$ showed the accessory pathway only in some cases. The computation of the solution to the inverse problem was repeated using a slightly changed model. For this the mean edge length of the body surface mesh was reduced to 2 cm to ensure that each electrode is located on a separate triangle.

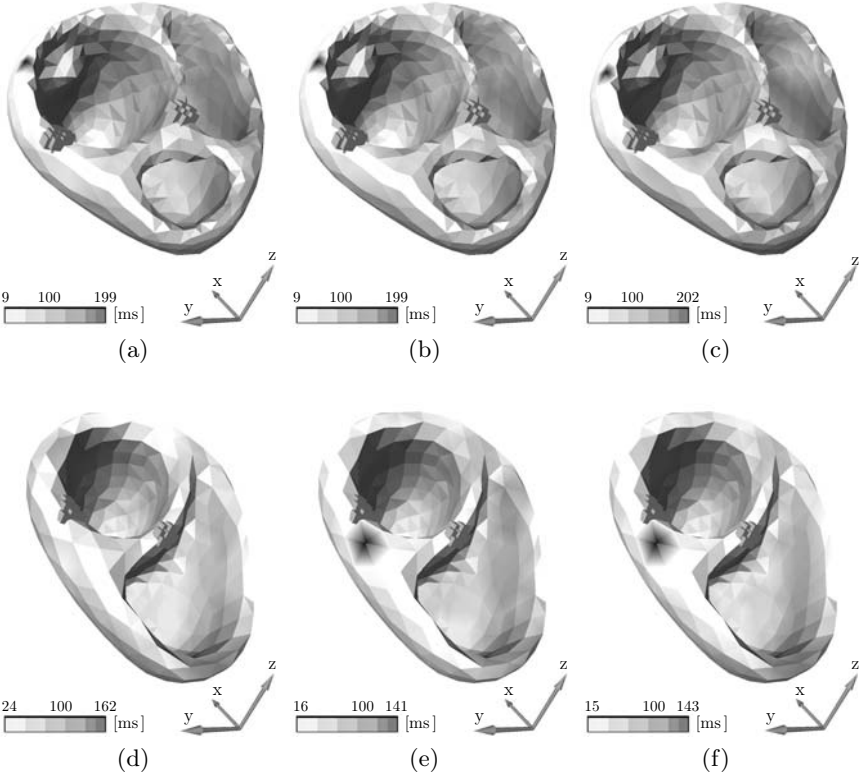


Fig. 4. Activation time patterns in two patients. The cubes in gray indicate the location of the left posterior accessory pathway and its location estimated by the conjugated gradient solver is depicted by the black spots. For the second patient (d), (e), (f) the pacing site corresponding to the pathway could be reconstructed properly, while it was shifted to the left for the first patient (a), (b), (c). (a) and (d) depict the solution gained for $S_{A,62}$ while the solutions for the arrays $S_{B,62}$ and $S_{B,125S}$ are depicted in (b), (c), (e) and (f).

4 Discussion

Looking at the BSP maps (Fig. 3) it was suggested that the arrays based upon the initial prototype $S_{B,62}$ are preferable for recording the BSP than the ones derived from $S_{A,62}$. In a first step presented in [5] the effort gain analysis was used to show that dense sampled BSP maps do improve the solution to the inverse problem. There it was found that the larger the rank R of the leadfield matrix $\mathbf{L}$ the more and higher spatial frequencies contained in the TMP $\varphi_m(t)$ it maps to the BSP $\mathbf{V}_e(t)$ measured by the electrodes. The higher these frequencies are, the smaller are the areas which correspond to a focal cardiac event. It was expected that BSP recordings made with an array exposing a large R lead to AT sequences depicting small focal events which may be invisible in case of low R. Figure 4 shows the

Table 1. Suggested number of nodes to use for the triangular meshes of different compartments for achieving proper inverse solutions

Nodes	Surface mesh
200 - 300	bloodmasses: atria and ventricles left and right lung
700 - 900	myocard: atria and ventricles
1500 - 2500	body surface

results gained from BSP recordings made by the arrays $S_{B,125S}$, $S_{B,62}$ and $S_{A,62}$. As can be seen exposing the highest rank $S_{B,125S}$ yields the best results whereas in case of patient 2 the event isn't visible to $S_{A,62}$ which has the lowest rank.

In order to be able to compute the AT maps for the 14 arrays for all 11 patients at a reasonable time the effective rank of the leadfield matrix $\mathbf{L}$ was replaced by its attainable rank. This is based upon the assumption that the attainable rank gained by the detector criterion described in [5] is equal to the real effective rank of $\mathbf{L}$.

Further, for some of the patient models the spacing of the triangular surfaces was too sparse. Due to non satisfying forward and inverse solutions the patient models as depicted in Fig. 4 were remeshed. The mesh sizes listed by Tab. 1 are a suitable tradeoff between the computational effort to solve the inverse problem and the accuracy of AT distribution maps.

In the results gained for the array schemes $S_{B,125S}$ and $S_{B,62}$, the estimated location of the pacing site was shifted to the left of its actual position. This was especially the case for the right lateral and left posterior WPW. This shift was observable in all 11 patients, with a varying distance between the accessory pathway and its estimated location. It is suggested that this shift is caused by the fact that the arrays $S_{B,125S}$ and $S_{B,62}$ only capture the BSP from the left anterior, lateral and posterior body surface. They do not provide any information upon the BSP on the right lateral chest surface. Looking on the local linear dependency maps computed in[5] it may be expected that this shift to the left can be reduced by recording the zero crossing of the BSP on the right lateral chest surface. To test this hypothesis $S_{B,125S}$ was extended by different sets of so called compensator electrodes (CS). It has to be noted that this effect can only be triggered when using an appropriate and precise patient torso model and ECG recordings with very low noise. Thus The CS electrodes can not be a replacement for a careful and proper preparation of the electroanatomic model and high quality recordings of the multi-lead ECG signals.

References

1. Huiskamp, G., Greensite, F.: A new method for myocardial activation imaging. IEEE Transactions on Biomedical Engineering **44** (1997) 433–446
2. Cuppen, J., Van Oosterom, A.: Model studies with inversely calculated isochrones of ventricular depolarization. IEEE Transactions on Biomedical Engineering **31** (1984) 652–659

3. Greensite, F.: Cardiac electromagnetic imaging as an inverse problem. Electromagnetics **21** (2001) 559–557
4. Dössel, O., Schneider, F.R.: Opportunities and limitations of non-invasive bioelectric imaging of the heart. In: European Medical and Biomedical Engineering Conference. (1999) 1226–1227
5. Hintermüller, C., Seger, M., Pfeifer, B., Fischer, G., Modre, R., Tilg, B.: Sensitivity and effort-gain analysis: Multi-lead ecg electrode array selection for activation time imaging. IEEE Transactions on Biomedical Engeneering (2006) (in press).
6. Dössel, O., Schneider, F., Müller, M.: Optimization of electrode positions for multichannel electrocardiography with respect to electrical imaging of the heart. In: Proccedings of the 20th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. Volume 20. (1998) 71–74
7. Lux, R.L., Smith, C.R., Wyatt, R.F., Abildskov, J.A.: Limited lead selection for estimation of body surface potential maps in electrocardiography. IEEE Transaction on Biomedical Engineering **25** (1978) 270
8. Usui, S., Araki, H.: Wigner distribution analysis of bspm for optimal sampling. IEEE Engeneering in Medicine and Biology **9**(1) (1990) 29–32
9. SippensGroenewegen, A., Peeters, H.A.P., Jessurun, E.R., Linnenbank, A.C., de Medina, E.O.R., Lesh, M.D., van Hemel, N.M.: Body surface mapping during pacing at multiple sites in the human atrium. Circulation **1997**(4) (1998) 369–380
10. Fischer, G., Tilg, B., Modre, R., Hanser, F., Messnarz, B., Wach, P.: On modeling the wilson terminal in the boundary and finite element method. IEEE Transactions on Biomedical Engineering **49**(3) (2000) 217–224
11. Johnson, C.R.: Computational and numerical methods for bioelectric field problems. Crit. Rev. Biomed. Eng. **25**(1) (1997) 1–81 Review.
12. Modre, R., Seger, M., Fischer, G., Hintermüller, C., Hayn, D., Pfeifer, B., Hanser, F., Schreier, G., Tilg, B.: Cardiac anisotropy: Is it negligible regarding noninvasive activation time imaging? IEEE Transactions on Biomedical Engineering (2004) (accepted).
13. Fischer, G., Pfeifer, B., Seger, M., Hintermüller, C., Hanser, F., Modre, R., Tilg, B., Trieb, T., Kremser, C., Roitinger, F.X., Hintringer, F.: Computational efficient noninvasive cardiac activation time imaging. Methods of Information in Medicine **44**(5) (2005) 674 – 686

Simulation of the Continuous Curvilinear Capsulorhexis Procedure

Kathrin Weber[1], Clemens Wagner[3], and Reinhard Männer[1,2]

[1] Institute for Computational Medicine, University of Mannheim, Germany
http://www.icm-mannheim.org
[2] Department of Computer Science V, University of Mannheim, Germany
http://www-li5.ti.uni-mannheim.de
[3] VRmagic GmbH, Mannheim, Germany
http://www.vrmagic.com

Abstract. The capsulorhexis procedure is a key part of cataract surgery. During capsulorhexis, a circular opening is torn into the lens capsule. For managing a circular tear, the medical literature details a set of instructions for the inexperienced surgeon. We have translated these instructions into a descriptive algorithm that simulates the behavior of the capsulorhexis tear. The lens capsule is modeled by a mass-spring mesh. To enable smooth tear propagation within the mesh, we adapted an algorithm originally developed for interactive cutting in triangulated surfaces that constantly re-triangulates around the tear end. The capsulorhexis procedure is implemented as a module for the ophthalmosurgical training simulator EYESI.[1]

1 Introduction

Pathological or aging processes may cloud the lens of the human eye. Cataract surgery replaces the clouded lens (the *cataract*) with an artificial lens implant. For that purpose, the *lens capsule* is opened, a clear, membrane-like structure that encloses the lens. After removing the lens, the open capsule serves as a container for the artificial lens. For preserving the stability of the capsule, Gimbel and Neuhann developed the Continuous Curvilinear Capsulorhexis [4]: An initial radial tear is set by an incision instrument and defines radius and position of the opening. The loosened membrane patch is grabbed by a forceps or a hook and is moved in a way that the tear proceeds circularly (see Fig. 1). The anatomy of the eye makes this task difficult: The equator of the human lens is adnate with *zonular fibers* that are under constant tension and control the focusing of the eye. This tension also affects the lens capsule so that the capsulorhexis tear tends to propagate towards the periphery instead of following the direction of pulling.

To teach an inexperienced surgeon how to control the tear, we developed a capsulorhexis module for the ophthalmosurgical training simulator EYESI [14]

[1] EYESI is a product of VRmagic GmbH, Mannheim, Germany.

M. Harders and G. Székely (Eds.): ISBMS 2006, LNCS 4072, pp. 113–121, 2006.

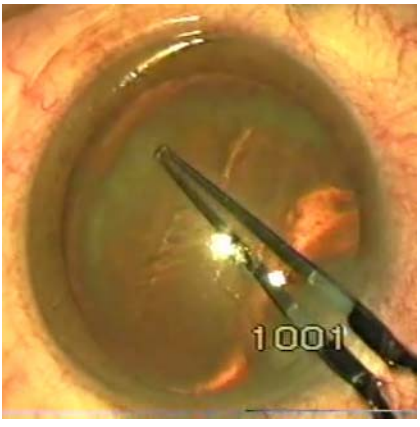


Fig. 1. A real capsulorhexis

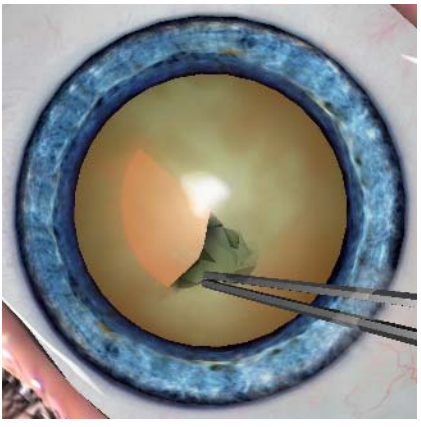


Fig. 2. A virtual capsulorhexis

(see Fig. 2). A mass-spring mesh serves as model for the deformation of the lens capsule. An algorithm for descriptive tear propagation in combination with an algorithm for interactive cutting models the capsulorhexis tear.

After a survey of related work, we present our implementation in Sect. 3 that cuts into two subsections: Sect. 3.1 concerns the descriptive algorithm, Sect. 3.2 describes the mesh adjustments which are necessary for a smooth tear propagation. Presentation of the results and an outlook conclude the paper.

2 Related Work

Related work covers developments of cataract surgery simulators within other research groups, modeling techniques concerning the simulation of deformable objects, and algorithms for cutting in triangulated surfaces.

2.1 Cataract Surgery Simulators

Several current research projects aim at realizing a cataract surgery simulator. The capsulorhexis procedure *"is universally acknowledged as one of the fundamental elements of modern phaco surgical techniques"* [12]. Hence, all these projects include the development of a capsulorhexis training module. However, the available publications just outline the implemented features and do not explicitly discuss the modeling of the capsulorhexis tear.

The Center for Advanced Studies, Research and Development in Sardinia is working on the EYESIM project. A mass-spring mesh is used for the simulation of the lens capsule, tearing is modeled by breaking overstretched springs [1]. The INRIA research project-team ALCOVE is developing a cataract surgery simulator that provides a capsulorhexis training based as well on a tearable mass-spring mesh [11]. This approach is also applied within the capsulorhexis

module of Webster et al. [15]. Recently, Webster et al. ported their module to the open API of the EYESI simulator which also served as platform for our implementation.

2.2 Physically Based Deformable Models

For the simulation of deformable materials, a number of physically based models have been established [8]. Recently, energy minimization models and meshless simulations have gained importance [13,7]. Mass-spring and Finite Element Models (FEM) are the most common methods used in surgical simulation [5]. The main advantage of FEM is that they allow physically accurate simulations based on the laws of continuum mechanics. To accelerate FEM for real-time applications, it is custom to linearize the involved strain and stress tensor. Especially concerning the simulation of highly deformable objects, this simplification may lead to physically incorrect and visually disturbing effects.

2.3 Cutting in Triangulated Surfaces

The smooth and steady propagation of a single tear requires a continuous retriangulation around the tear end. In the literature, this problem is usually associated with cutting approaches. In this context, subdivision methods are very popular which refine those elements that are in contact with a scalpel [2,6]. These methods typically increase the mesh size and tend to produce degenerated elements. Ganovelli and O'Sullivan reduced the number of degenerated triangles by collapsing very short edges [3]. Since this solution does not repair all inconsistencies, Nienhuys and van der Stappen proposed to combine local mesh refinement with local on-the-fly Delaunay Triangulation [10]. It enables smooth extension of the cutting edge while maintaining the mesh quality. Therefore, we use this algorithm to handle topological changes along the capsulorhexis tear (see Sect. 3.2).

3 Simulating the Capsulorhexis Procedure

During capsulorhexis, the detached membrane patch is folded and crumpled by the surgical instrument. The stress distribution within the lens capsule determines the propagation of the tear. A physically accurate simulation of the lens capsule could facilitate the physically accurate prediction of the tear propagation. This would require a non-linear model that, in addition, must deal with topological changes along the tear. However, the computational cost of such a model is too high for the necessary update rate of a real-time application ($\geq$ 30Hz). Thus, we subdivide the problem and separate the simulation of the tear from the simulation of the lens capsule.

Based on the medical literature, we have developed a descriptive algorithm that controls the propagation of the capsulorhexis tear (see Sect. 3.1). Thus, the

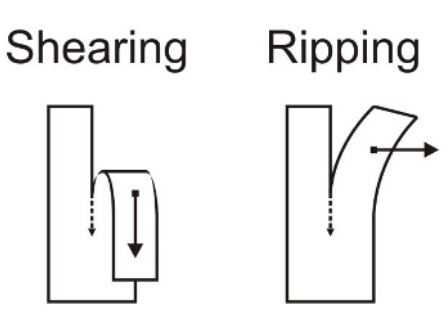


Fig. 3. Capsulorhexis techniques

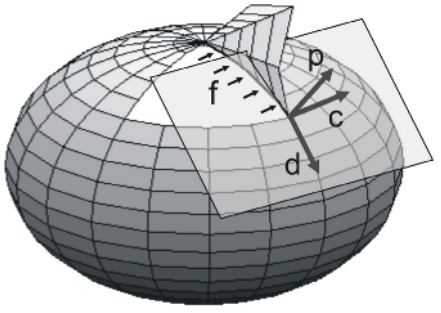


Fig. 4. Tangent plane on the lens

simulation of the membrane patch is of secondary importance so that a mass-spring mesh is quite adequate as basic model for the lens capsule. Stress within the membrane patch is only considered to trigger the tearing process. Apart from that, direction and velocity of the tear propagation are completely independent from the simulation of the lens capsule.

Each time a new tear direction has been calculated, the tear end propagates a small constant distance along this direction. Subsequently, the mass-spring mesh is updated accordingly (see Sect. 3.2).

3.1 Descriptive Propagation of the Capsulorhexis Tear

As the lens capsule is under constant tension, the capsulorhexis tear tends to propagate towards the periphery of the lens. To counter this drift and to manage a circular tearing edge, the technical literature describes two different techniques: *shearing* and *ripping*. Seibel [12] characterizes these two techniques as follows: In order to trigger shearing (Fig. 3, left), the surgeon has to flatten the membrane patch such that the inner surface of the membrane points upwards. Pulling the folded patch causes the capsulorhexis tear to follow approximately the movement of the instrument. However, shearing only enables small-sized deviations from the current tear direction and is unsuitable for the rescue of an already misrouted tear. In contrast, ripping (Fig. 3, right) enables relatively abrupt changes in the tear propagation. In order to switch to ripping, the surgeon has to unfold the patch such that its inner surface faces the lens. To achieve a circular tear propagation during ripping, the surgeon must pull towards the center of the lens. This behavior appears strange and the inexperienced surgeon needs practice to get used to it.

In our implementation, a parameter called `Indicator` determines whether the surgeon initiates shearing or ripping. We observe those triangles of the mass-spring mesh that are both adjacent to the tear end and part of the already detached membrane patch. If the according triangle normals face the lens surface, shearing will be triggered (`Indicator := Shearing`) otherwise ripping (`Indicator := Ripping`).

Basic Parameters of Tear Propagation. Concerning both shearing and ripping, the new direction for the propagating tear depends on the current tear direction and the direction of pulling. These parameters have to be measured in an appropriate manner: Since the lens capsule is attached to the lens with its smooth, convex surface, the current end of the tear and the according surface normal of the lens define a tangent plane which we use as subspace for our calculations. Projection of the tearing edge onto the plane yields the current tear direction `CurrDir` (vector **c** in Fig. 4) and projection of the instrument's tip yields the current direction of pulling `PullDir` (vector **p** in Fig. 4). Also the direction of the drift `DriftDir`, that points towards the periphery, can be defined within the tangent plane (vector **d** in Fig. 4). The new direction of the tear propagation is determined by rotating `CurrDir` around the surface normal of the tangent plane. The final rotation angle is the sum of two signed angles: `DriftAngle` denotes a rotation towards `DriftDir`, `PullAngle` denotes a rotation towards `PullDir`. The calculation of these two rotation angles depends on the value of `Indicator`.

Shearing. If the surgeon applies this technique, the drift only has a minor effect on the tear propagation. Thus, during shearing `DriftAngle` is set to a small, constant value for the rotation of `CurrDir` towards `DriftDir`. On the other hand, the value of `PullAngle` corresponds to the angle enclosed by `CurrDir` and `PullDir` and defines the rotation of `CurrDir` towards `PullDir`. Since shearing may provoke only small changes in `CurrDir`, `PullAngle` must not exceed a predefined maximum. The sum of both `DriftAngle` and `PullAngle` yields the final rotation angle for `CurrDir`.

Ripping. During ripping, the drift strongly affects the propagating tear. Therefore, `DriftAngle` is higher than during shearing. Experienced surgeons describe that ripping sometimes requires a direction of pulling that is nearly perpendicular to the desired tear propagation. In order to model this behavior, `PullAngle` is only assigned a predefined fractional amount of the angle enclosed by `CurrDir` and `PullDir`. Since ripping enables relatively abrupt turnarounds of `CurrDir`, `PullAngle` is constrained by a maximum that is higher than the according maximum during shearing.

3.2 Mesh Adjustment During Tearing

Every time the position of the tear end node is changed, the surrounding triangles have to be adjusted. In addition, another piece of the lens capsule will be detached from the lens and the membrane patch grows by this piece.

Progress of the Tearing Edge. The cutting algorithm proposed by Nienhuys et al. defines a scalpel that can be moved by the user [10]. In our training module it is the tearing algorithm that controls the movement of the "scalpel". Aside from this point, the original cutting algorithm can be used for our purpose, thus we describe the remeshing process only briefly:[2]

[2] We refer the reader to the technical report [9] that is more detailed than the homonymous paper [10].

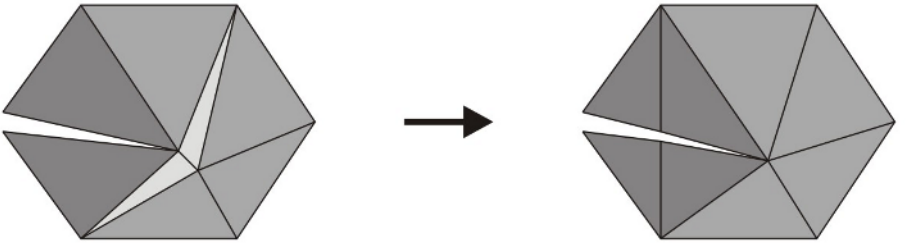


Fig. 5. Nodes close to the tear end will be removed and long tearing edges will be split

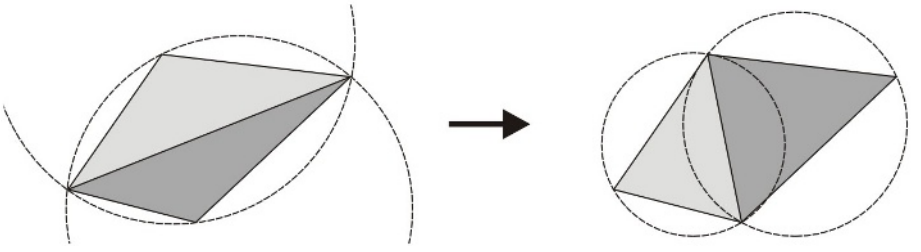


Fig. 6. An edge that is not Delaunay will be flipped

1. If the distance between the propagating tear end node and an adjacent node is less than a predefined threshold parameter, the adjacent node will be removed. Thus, the two light-gray triangles in Fig. 5 disappear.
2. If the length of the tearing edge behind the tear end is greater than a predefined threshold parameter, the corresponding edges will be split. Thus, the algorithm subdivides the two dark-gray triangles in Fig. 5.
3. To assure mesh quality around the tear end node, the *Delaunay Criterion* is used. An edge is called *Delaunay* if the circumcircles of the two adjacent triangles do not contain the respective remaining triangle node. If an edge is not Delaunay, it will be *flipped* as shown in Fig. 6 so that the minimal angle within the two triangles increases. Every time the tear end node progresses, the surrounding edges will be checked and flipped if necessary.

Progress of the Membrane Patch. The membrane patch defines a *tearing front* that separates the detached membrane from the part of the lens capsule which is still attached to the lens surface. The tearing front is illustrated in Fig. 4 by **f** and has to propagate along with the tear end node which always defines the outer end of the front. As long as the capsulorhexis covers less than 360 degrees, the inner end of the tearing front is given by a node in the middle of the mesh. Fig. 7 exemplifies this state of a virtual capsulorhexis and shows a wire frame of the lens capsule. If the diameter of the capsulorhexis is too small, the surgeon must increase it by continuing the capsulorhexis beyond 360 degrees.

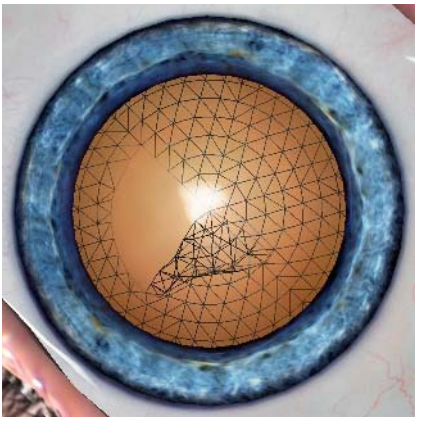


Fig. 7. A capsulorhexis that covers less than 360 degrees (number of triangles: 486)

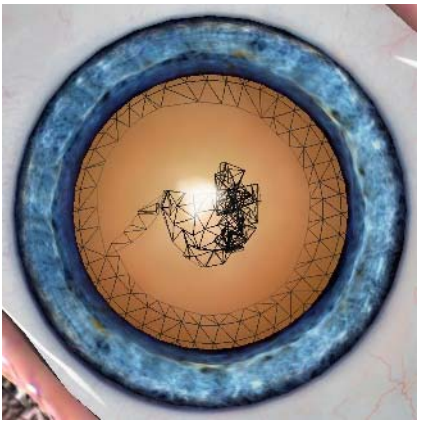


Fig. 8. Continuation of the capsulorhexis shown in Fig. 7 beyond 360 degrees (number of triangles: 530)

Once the capsulorhexis exceeds 360 degrees, the inner node of the tearing front also detaches from the lens as indicated in Fig. 8.

Every node of the mesh holds a boolean flag `Attached`. If `Attached` is `TRUE`, the corresponding node is excluded from the mass-spring simulation and thus, attached to its initial position on the lens surface. A flag switches from `TRUE` to `FALSE` at the time when the tearing front passes the corresponding node. The tearing front is given by a line segment that connects the inner and outer end of the front and that is projected onto the lens surface.

4 Results

The proposed tearing algorithm models the specific behavior of a capsulorhexis tear, including the distinction between shearing and ripping: If the surgeon proceeds inattentively, an unintentional switch from shearing to ripping may occur. At this moment, the direction of pulling is not adequate for ripping which leads to a tear propagation towards the periphery – an effect that is described in the literature. On the other hand, the medical literature recommends controlled ripping to redirect a misrouted capsulorhexis tear. Within our model, only ripping allows for relatively abrupt turnarounds of the current tear direction. Thus, the surgeon learns to apply planned ripping maneuvers for the rescue of an already errant capsulorhexis tear.

The combination of the algorithm with online remeshing around the tear end node enables the simulation of a steady tear propagation that hides the polygonal structure of the mass-spring mesh. However, the resolution of the mesh must be

adapted to the threshold parameters of the underlying cutting algorithm. Fig. 7 and Fig. 8 exemplify a resulting tearing edge.

Due to the computational efficiency of the mass-spring model and the descriptive algorithm, the complete EYESI module fulfills the real-time condition. As the propagation of the tear is separated from the simulation of the lens capsule, the choice of the mass-spring parameters and the time integration scheme is relatively uncritical.

5 Discussion and Outlook

We have combined an algorithm for descriptive tear propagation with an algorithm for interactive cutting in order to simulate a tearing membrane. The implementation runs as a cataract training module on the EYESI system and is already in use for surgical training.

The validation by ophthalmosurgeons has shown that the simulated behavior of the tear is realistic enough to meet the demands on a capsulorhexis training. However, the descriptive nature of the model still restricts the possibilities of the surgeon: The descriptive algorithm is mapped onto a single tear that is automatically initialized when starting the module. Currently, it is not possible to incise the membrane during the simulated procedure at arbitrary positions.

In order to enable the interactive use of an incision instrument, current work includes the simulation of multiple tears. This will allow for the integration of the initial capsulorhexis incision as part of the training task. Besides, we are working on the simulation of viscoelastic fluid injection that affects the behavior of the tissue.

References

1. M. Agus, E. Gobbetti, G. Pintore, G. Zanetti, and A. Zorcolo. Real-time cataract surgery simulation for training. In *Eurographics Italian Chapter*. Eurographics Association, 2006.
2. D. Bielser and M. H. Gross. Interactive simulation of surgical cuts. In *Pacific Graphics*, pages 116–125. IEEE Computer Society Press, 2000.
3. F. Ganovelli and C. O'Sullivan. Animating cuts with on-the-fly re-meshing. Eurographics Short Presentation, 2001.
4. H. V. Gimbel and T. Neuhann. Development, advantages, and methods of continuous circular capsulorhexis technique. *Journal of Cataract and Refractive Surgery*, 16(1), 1990.
5. A. Liu, F. Tendick, K. Cleary, and C. Kaufmann. A survey of surgical simulation: applications, technology, and education. *Presence: Teleoperators and Virtual Environments*, 12(6):599–614, 2003.
6. A. Mor and T. Kanade. Modifying soft tissue models: Progressive cutting with minimal new element creation. In *Medical Image Computing and Computer-Assisted Intervention*, pages 598–607. Springer-Verlag, 2000.
7. M. Müller, R. Keiser, A. Nealen, M. Pauly, M. Gross, and M. Alexa. Point based animation of elastic, plastic and melting objects. In *ACM SIGGRAPH / EUROGRAPHICS Symposium on Computer Animation*, 2004.

8. A. Nealen, M. Müller, R. Keiser, E. Boxermann, and M. Carlson. Physically based deformable models in computer graphics. In *Eurographics, State of the Art Report*, pages 71–94, 2005.
9. H.-W. Nienhuys and A. F. van der Stappen. A delaunay approach to interactive cutting in triangulated surfaces. Technical Report UU-CS-2002-044, Utrecht University, Institute for Information and Computing Sciences, 2002.
10. H.-W. Nienhuys and A. F. van der Stappen. A delaunay approach to interactive cutting in triangulated surfaces. In *Algorithmic Foundations of Robotics V*, pages 113–129. Springer-Verlag, 2003.
11. Project-Team ALCOVE. Interacting with complex objects in collaborative virtual environments. Activity Report INRIA, 2004.
12. B. S. Seibel. *Phacodynamics - Mastering the Tools and Techniques of Phacoemulsification Surgery.* SLACK Incorporated, fourth edition, 2005.
13. M. Teschner, B. Heidelberger, M. Müller, and M. Gross. A versatile and robust model for geometrically complex deformable solids. In *Computer Graphics International*, pages 312–319. IEEE Computer Society, 2004.
14. C. Wagner, M. Schill, and R. Männer. Intraocular surgery on a virtual eye. *Communications of the ACM*, 45(7):45–49, 2002.
15. R. Webster, J. Sassani, R. Haluck, R. Shenk, M. Harris, J. Blumenstock, J. Gerber, C. Billman, and A. Benson. Simulating the continuous curvilinear capsulorhexis procedure during cataract surgery on the EYESI system. In *Medicine Meets Virtual Reality*, pages 592–595, 2005.

Annotated Surgical Manipulation for Simulator-Based Surgical Skill-Transfer Using SiRE – Simulation Record Editor

Mikko Rissanen[1], Yoshihiro Kuroda[2], Megumi Nakao[3], Naoto Kume[1], Tomohiro Kuroda[4], and Hiroyuki Yoshihara[4]

[1] Graduate School of Informatics, Kyoto University *
[2] Graduate School of Medicine, Kyoto University *
[3] Graduate School of Information Science, Nara Institute of Science and Technology **
[4] Department of Medical Informatics, Kyoto University Hospital *
* Kyoto Univ. Hospital, 54 Shogoin Kawahara-cho, Sakyo-ku, Kyoto, 606-8507, Japan
{mikko, ykuroda, kume, tkuroda, lob}@kuhp.kyoto-u.ac.jp
** 8916-5, Takayama, Ikoma, Nara, 630-0192, Japan
meg@is.naist.jp

This paper outlines the efforts for developing a framework for annotation of recorded surgical procedures toward the goal of simulator-based skill-transfer in surgery. The framework is based on SiRE – Simulation Record Editor – an application that allows medical teachers to produce variations and make perfect example surgical procedures even without surgical manipulation skills. By using SiRE, physical properties of the recorded manipulation can be modified interactively and accurately. Edited examples can be annotated as they are intended as self-learning media in surgical training of the future.

1 Introduction

Research of medical Virtual Reality (VR) has been focusing on the realism of visual and haptic feedback of simulation (for example [1, 2]). However, in the medical field, realism is not the only key element of successful simulator-based training. Modelling and evaluation of surgical skills [3, 4, 5] are emerging as a major research field. Studies on manual skill-transfer have introduced a record-and-play approach with the aid of haptic devices. Such studies aim for the benefits of using digital data [6, 7]:

- Freedom of time, location and number of teachees
- Direct and exact feeling of the model with unlimited repetition
- Flexibility of learning strategies
- Recorded digital data can be evaluated and modified

This approach is usually demonstrated in writing with a pen [6, 8]. If similar approach was possible in surgery, the dominant master-apprentice teaching methodology could be changed for the better. In the complex surgical field, however, to understand the recorded example manipulation is not trivial. While research on using

M. Harders and G. Székely (Eds.): ISBMS 2006, LNCS 4072, pp. 122 – 131, 2006.

VR for understanding complex concepts continues [9], the need for annotation to describe surgeon's line of thought has been acknowledged [10, 11]. So far, annotation has been applied to the traditional teaching media, such as video, and the main task of the authors is to investigate annotation as an essential part of surgical training simulation.

A framework for surgical skill-transfer based on annotation of recorded surgical simulation is being developed. The example manipulation and comments are included into a Simulation Record (SR) which is accessible through the training simulation. In addition, the recorded example surgical procedures can be made interactive at will so that the user can perform at any stage of the recorded simulation. The goal of this approach is to create a virtual self-learning environment that would reduce the need for time-consuming one-on-one instruction. Case-specific knowledge and the manual skill are encapsulated into Annotated Simulation Records (ASRs).

Before annotation, it is important to have meaningful recorded example manipulation. SiRE – Simulation Record Editor – allows editing of recorded physics-based simulation. Two scenarios for editing were identified: making of variations and perfecting example surgical procedures. With SiRE, the recorded manipulation can be fine-tuned so that errors would be omitted out in the example. This possibility is only present in VR and it reduces the need for annotation on the differences between what was done and what should have been done during the surgical procedure. Variable approaches can be created from the original SR by scaling its trajectory data. The scaling allows even a person without true manipulation skills to edit SRs. Recorded manipulation can be edited interactively and accurately by using two editing methods: re-recording and setting key values that are used for scaling the original trajectory. The simulation engine can be utilized for setting the key values.

The paper is organized as follows. First, a brief overview of the annotation framework is given. The remaining sections discuss a part of the framework, a simulation record editor, in more detailed manner including preliminary results.

2 Related Work

To achieve the fore-mentioned benefits of digital data in surgical training is challenging. The medical field is complex (in nature) and the need for annotation of surgical manipulation has been realized. The Virtual Rounds concept [10] highlighted the need for a full simulation system instead of mere simulation engine in medical training. In that concept, annotation is an important aspect that facilitates asynchronous teaching and self-learning process. Immersive Electronic Books (IEBooks) [11] supports self-learning from recorded simulation. IEBooks are surgical procedures recorded with video cameras by using image processing techniques. IEBooks can be annotated by linking text, audio messages, video clips and images. Past surgical procedures can be viewed in an immersive virtual environment. IEBooks are intended as a learning environment for surgical decision-making. Therefore, interaction of the user is restricted to three-dimensional viewing without any support for surgical manipulation with haptic devices.

In order to facilitate surgical training in various cases, scene building applications have been developed [12, 13, 14]. These systems support changeable and modifiable

anatomy and interaction models. This paper introduces an application that allows changeable and modifiable example manipulation.

A mechanism for editing recorded simulation using suitable data formats and managing their synchronization has been proposed [15]. The editing of SRs has not been clearly addressed in the studies so far from the user's point of view. This paper focuses on the algorithms that enable editing suitable for the surgical context.

3 Annotation Framework

The design of the annotation framework for simulator-based surgical skill-transfer has been affected by the need for annotation and the benefits of digital data, especially recorded VR. In this paper, only one of the major elements of the framework, authoring of SRs by editing physics-based simulation, is discussed in detail.

3.1 The Concept

Figure 1 presents an approach for a novel teaching and learning methodology based on ASRs. Surgical procedures are captured by recording the user's manipulation on the simulator at the best possible accuracy. SR contains the manipulation as well as the visual playback which is easy to control for the user. The SR is annotated by the teacher to demonstrate pedagogical points within the SR's scope. The annotation data acts as knowledge transfer medium whereas the recorded manipulation contains the

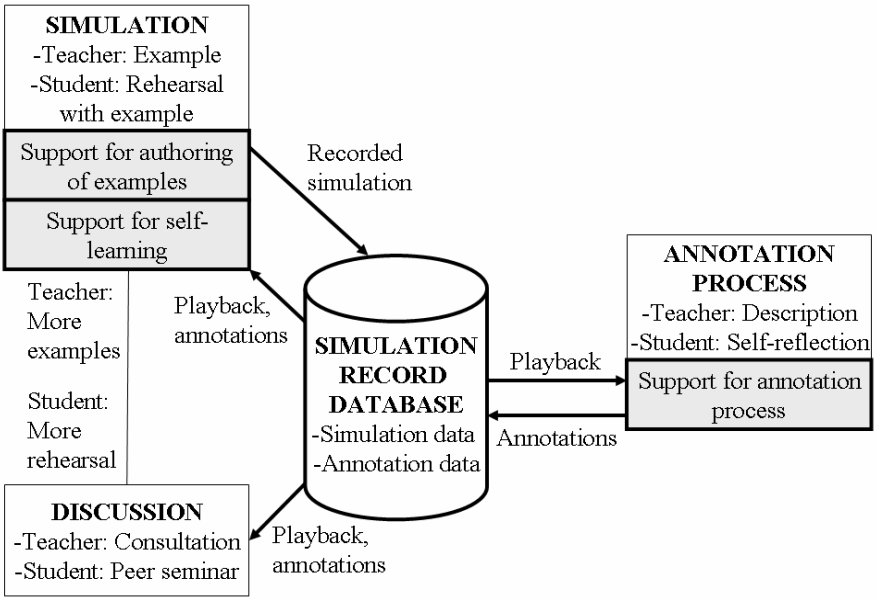


Fig. 1. ASRs for simulator-based surgical skill-transfer. The system's support has three targets: authoring of SRs, annotation of SRs and self-learning using ASRs.

motor-skill. A self-learning scenario is produced for the students who can gain knowledge from the annotations and interact with the same simulated surgical setting. The framework is designed for indirect simulator-mediated teaching of surgery.

Processes for the production of ASRs are *recording*, *editing*, and *annotation*. Processes can be shared between individuals so that a surgeon is not required to be involved in all the phases. Instead, editing and annotation can be left to a general medical teacher in the cases that do not require description of the surgeon's line of thought. The medical teacher can produce examples after the surgeon's manipulation has been recorded.

3.2 The Role of Annotation

Traditionally, text-books have been annotated by underlining, drawing, sketching and commenting. The notes have definite value for an individual studying the content. In addition, personal annotations can be useful to others, and annotations can make consensus within the readers explicit [16]. Nowadays, annotation of any multimedia objects is possible. Annotation of VR is a natural extension of this development.

Both knowledge and manipulation skill can be mediated through the ASR which acts as *reference of the simulation system* [10]. ASRs can be used at several levels. For example, an ASR could be annotated to introduce the basic concepts of the scenario. At this point, the existing digital teaching material can be linked to specific segments of the ASR. Later, annotation could reach another level of details for self-learning the surgical procedure as a whole. Students can annotate their experiences as learning diaries, which allows *self-reflection* [10] by making notes of one's progress.

The authors highlight the importance of the relationship between surgical manipulation and its target. The annotation process aims to define the targets of each segment of manipulation. The framework's objective is to detect the relationships automatically after the targets have been defined by the annotator. This would construct a model of correct manipulation that can be used as evaluation metrics in each segment. Self-learning would be supported by pointing out the user's mistakes compared to the model. In addition, variations of the manipulation could be generated automatically after the targets have been determined in a new anatomical setting.

3.3 Design of SiRE – Simulation Record Editor

VR is characterized as a rich and flexible presentation. VR can render high resolution and photorealistic images and realistic haptic sensation. On the other hand, VR can simplify and facilitate understanding the nature of the situation and the manipulation.

In traditional methods, anatomy textbooks and cadavers are the predominant learning media. With these media, perfecting the example manipulation is impossible. Only VR gives the possibility of unlimited repetition of realistic cases that can be improved from the original recorded data.

Surgeons are usually occupied with the clinical work, which often conflicts with teaching responsibility. SiRE allows an different person, a general medical teacher, to make variations of an original recorded manipulation performed by the surgeon.

3.4 Data Model

As shown in Figure 2, SiRE's data model is divided into two basic layers: raw input data and visual playback medium. The raw input data covers the complete time series of device-specific data, typically position and posture of manipulator points. For example, with a glove-shaped interface positions and angles of fingers' joints are recorded. The playback medium contains time series of 3D structure of the model. Moreover, the playback can contain any additional information that the simulation engine is capable of simulating, such as force and colliding regions.

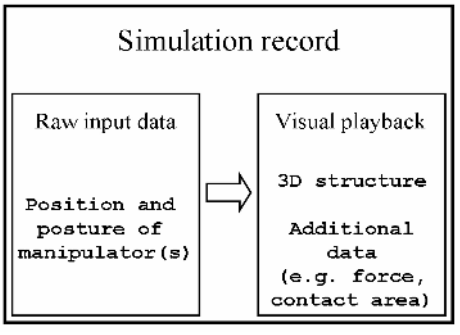


Fig. 2. Composition of simulation record

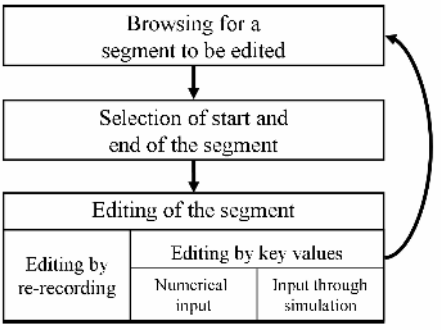


Fig. 3. Process for editing SRs with two editing methods

3.5 Design for Editing

For making variations and perfecting examples of surgical manipulation, the following requirements were determined:

- Replacement of total failures in the original manipulation.
- Making variations by replacing selected sections of the original manipulation.
- Correction of very small mistakes with high accuracy.

The need for true manipulation skill should be minimized, so that a person other than a surgeon could be responsible for the editing.

In addition, the editing process should be easy to use. The user should be able to modify an incorrect location of contact between the skin and the scalpel by simply pointing a new location with the haptic interface used in the simulator. Figure 3 shows the editing process of SRs. A segment to be edited is browsed and the segment's start and end points are determined. Then, the segment is edited in the following methods.

Re-recording is an obvious option for an editing method. The original segment is replaced by a new manipulation recorded by the user. However, re-recording can result in manipulation that is not fully continuous. Therefore, differences between the start and the end point of the new and the original manipulation are interpolated. At the moment, the design of SiRE does not address the problem of discontinuity. Re-recording is suitable for correcting total failures and for recording alternative approaches that require very different manipulation than the original one.

Editing by key values is an editing method that modifies the original manipulation. A new value given by the user is used for calculating a new raw-data trajectory to result in a satisfactory simulation. The key value can be given numerically by modifying values of physical properties of the simulation visible in the GUI, or by interacting with the simulation directly. When interacting with the simulation directly, at first, the user is given control of the simulation at the segment's start situation. A state of the user-controlled manipulation, chosen by the user, is recorded. The state is considered as the new key value, by which the original manipulation is calculated. This approach allows editing another person's manipulation without affecting its individual-dependant properties, such as smoothness of the trajectory or velocity, since the original manipulation is modified from the original recorded data. Numerical editing provides the best accuracy for the new key values, but lacks interaction with the simulation. It is a suitable method for fine-tuning the manipulation and numerical tasks, such as normalization of segments. An example of normalization is setting power level in each segment to a limit value in subtle palpation to avoid excessive force that would result in harmful manipulation. Figure 4 shows an example of how a contact location can be edited by defining a key value.

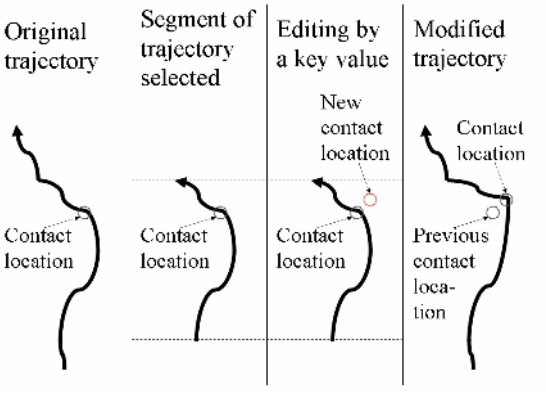


Fig. 4. Editing by setting key values. The trajectory is scaled by the given contact position.

3.6 Algorithm for Scaling Trajectories

When scaling a trajectory to reach a certain point in the simulation space, the properties of manipulation that are essential in the surgical context have to be taken into account. The trajectory should preserve its smoothness and average velocity after scaling. Therefore, when using position time-series sampled at a constant interval, additional points have to be added when the target point (p_t) is further than the trajectory's furthest point (p_n). Consequently, points have to be deleted when the trajectory should be shortened. The algorithm presented in Figure 5 is suitable for such scaling. When the furthest point p_n has been determined, the trajectory is divided in two so that both parts approach p_n. The algorithm is applied to both parts separately. In order to lengthen the trajectory, points are duplicated when lengthening

the trajectory and consequently removed when shortening. Thus, smoothness and average velocity remain almost the same as in the original trajectory. (1) is executed in a loop until the remaining distance to the target point becomes smaller than the distance added. The last addition is performed with (2) to ensure perfect accuracy, which is the only modification that affects the average velocity.

Besides using the algorithm for scaling the trajectory to reach to a certain location defined by the user, it can be applied for scaling on the basis of other properties of the simulation. For example, as in Figure 6, power the user-given value. A tangent vector is formed of the first and the last collision points (p_{col1} and p_{col2}). The vector determines the direction of scaling. Pre-simulation is executed so that a temporary position proceeds along the tangent vector until the user-given power is met at p_c. Distance between the vector $p_{col2}-p_{col1}$ and p_c is calculated with a cross product and the target position p_t is calculated from the difference of distance to the vector $p_{col2}-p_{col1}$ from the points p_c and p_n. The direction of scaling can also be determined simply to p_t-p_{col1}, or some other vector, depending on the case.

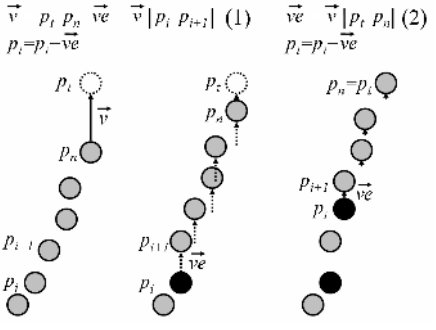


Fig. 5. Scaling algorithm for trajectories

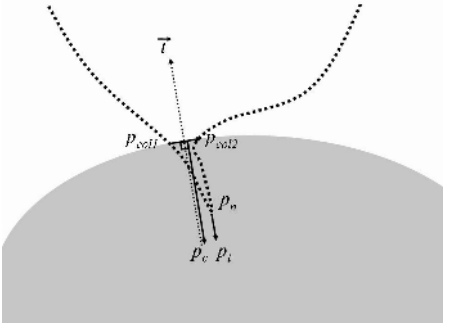


Fig. 6. The target position p_t defined by a power value

4 Experiment

Preliminary experiment evaluates the scaling algorithm with two criteria: smoothness of motion and average velocity. If both factors remained about the same as in the unedited trajectory, the scaling algorithm can be used without the risk of losing the properties of the recorded manipulation that are important in the medical context. Shaking of the hand can be a valuable clue for learning when reviewing recorded manipulation. Smoothness was evaluated with time-integrated squared jerk metrics [17] which compares the difference of acceleration in the trajectory. Smaller value means smoother trajectory. Average velocity should also remain the same.

The experiment procedure is straight-forward. At first, a trajectory is recorded in a simulation and evaluated with the two above-mentioned criteria. The trajectory is scaled up by giving power values at a constant interval. The scaled trajectories are finally compared to the original trajectory.

SiRE was implemented as an add-on to the Medical Virtual reality simulation Library (MVL) [14] system. The MVL runs on a Xeon 3.2 GHz dual processor with 4 GB RAM. The haptic user interface is PHANToM Desktop. The raw input data (position of the 1-point manipulator) was recorded the rate of the haptic loop at about 1000 Hz. The recorded trajectory consisted of a light push against a simulated soft-tissue model with a 0.78 N maximum power. Power value for scaling was increased by 0.2 N up to 1.98 N so that the trajectory gained length. Due to restrictions of the given trajectory information (1 ms sampling) and the mathematical nature of the algorithm, trajectory's shape was not evaluated.

5 Results

Figure 7 presents the results. Both smoothness and average velocity remained about the same as in the original trajectory. Scaling of trajectory with the algorithm does not reduce the properties of the manipulation essential for surgical assessment. Therefore, by using this algorithm SiRE allows person without manipulation skills to edit SRs without hindering possibilities for objective assessment of the manipulation.

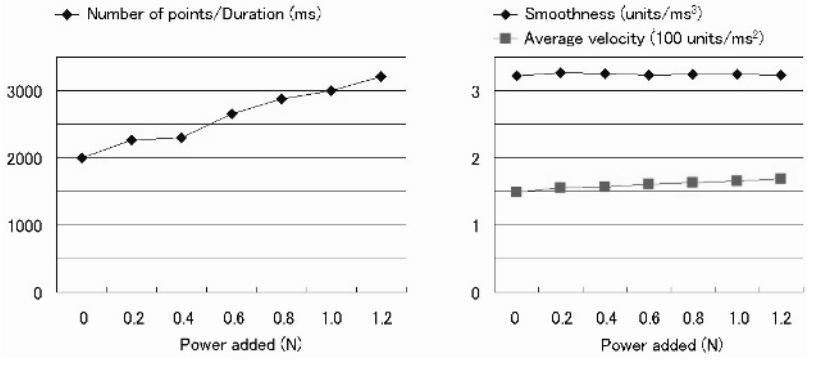


Fig. 7. Results of the experiment. Left: Increase of points in the trajectory. Right: Smoothness and average velocity. The first element at 0 N is the original unedited trajectory. Slight variance in accuracy is due to rounding.

6 Conclusion

The development of a framework for annotation of recorded surgical procedures was summarized briefly in this paper. The framework includes SiRE – Simulation Record Editor – that allows medical teachers to produce variations and perfecting example surgical procedures from recorded physics-based simulation. SiRE has two editing methods: re-recording and setting of key values through interactive simulation or numerical input. Algorithm for the latter method was validated to preserve the essential properties of manipulation, smoothness and average velocity. The algorithm enables intuitive and accurate editing of SRs even without true manipulation skills. In Figure 8, a SR is being perfected by adding more power into a selected segment.

Next target for the research on SiRE are algorithms that ensure natural continuity of manipulation when new segment of SR is re-recorded and connected to the original SR. The final goal of this research project is to demonstrate that ASRs can act as an indirect teaching media in surgical training. The framework will be available in 2007.

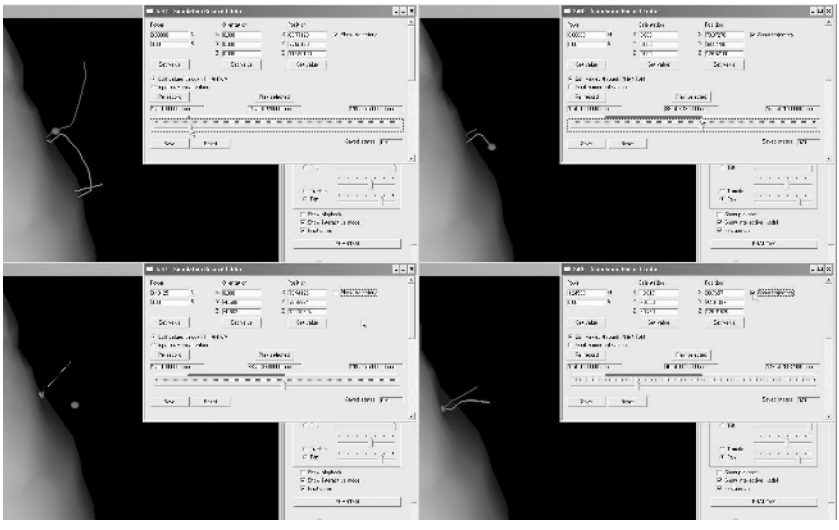


Fig. 8. SR representing palpation of the aorta is perfected by scaling the trajectory. Top left: trajectory of the whole SR is visualized. The SR contains two contacts with the target. Segment's start is determined. Top right: Segment's end determined (segment's borders are shown on top of the timeline). Bottom left: Approaching the new key value. Trajectory is scaled when Set-button is clicked. Bottom right: Updated segment with the scaled trajectory.

Acknowledgements

This research is supported by Grant-in-Aid for Scientific Research (S) (16100001), Young Scientists (A) (18680043) and Exploratory Research (18659148) from The Ministry of Education, Culture, Sports, Science and Technology, Japan, and Nakajima Fund.

References

1. Berkley J, Turkiyyah GM, Berg D, Ganter MA, Weghorst S (2004) Real-time finite element modeling for surgery simulation: an application to virtual suturing. IEEE Transactions on Visualization and Computer Graphics 10(3):314–325
2. Nakao M, Kuroda T, Komori M, Oyama H (2003) Evaluation and user study of haptic simulator for learning palpation in cardiovascular surgery. In: Int. Conference on Artificial Reality and Telexistence (ICAT). The Virtual Reality Society of Japan, pp. 203–208

3. Satava RM, Cuschieri A, Hamdorf J (2003) Metrics for objective assessment. Surgical Endoscopy 17:220–226
4. Cristancho SM, Hodgson AJ, Pachev G, Nagy A, Panton N, Qayumi K (2006) Assessing cognitive & motor performance in Minimally Invasive Surgery (MIS) for training & tool design. In: Medicine Meets Virtual Reality (MMVR). Westwood JD et al. eds. IOS Press, pp. 108–113
5. Mackel T, Rosen J, Pugh C (2006) Data mining of the E-Pelvis simulator database: A quest for a generalized algorithm capable of objectively assessing medical skill. In: Medicine Meets Virtual Reality (MMVR). Westwood JD et al. eds. IOS Press, pp. 355–360
6. Teo CL, Burdet E, Lim HP (2002) A robotic teacher of Chinese handwriting. In: Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. IEEE Computer Society Press, pp. 335–341
7. Kikuuwe R, Yoshikawa T (2001) Haptic display device with fingertip presser for motion/force teaching to human. In: IEEE International Conference on Robotics and Automation (ICRA). IEEE Computer Society Press, pp. 868–873
8. Saga S, Vlack K, Kajimoto H, Tachi S (2005) Haptic video. In: SIGGRAPH, DVD-ROM
9. Alverson DC, Saiki SM Jr., Caudell TP, Goldsmith T, Stevens S, Saland L et al. (2006). Reification of abstract concepts to improve comprehension using interactive virtual environments and a knowledge-based design: A renal physiology model. In: Medicine Meets Virtual Reality (MMVR). Westwood JD et al. eds. IOS Press, pp. 13–18
10. Shaffer D, Meglan D, Ferrell M, Dawson S (1999) Virtual Rounds: Simulation-based education in procedural medicine. In: SPIE Vol. 3712: Battlefield Biomedical Technologies
11. Welch G, State A, Ilie A, Low K-L, Lastra A, Cairns B, Towles H, Fuchs H, Yang R, Becker S, Russo D, Funaro J, van Dam A (2005) Immersive electronic books for surgical training. IEEE Multimedia 12(3):22–35
12. Montgomery K, Bruyns C, Brown J, Sorkin S, Mazzella F, Thonier G et al. (2002) Spring: A general framework for collaborative, real-time surgical simulation. In: Medicine Meets Virtual Reality (MMVR). Westwood JD et al. eds. IOS Press, pp. 296–303
13. Acosta E, Temkin B (2004) Build-and-Insert: anatomical structure generation for surgical simulators. In: International Symposium on Medical Simulation (ISMS). Cotin S, Metaxas DN eds. Springer, pp. 230–239
14. Kuroda Y, Nakao M, Kuroda T, Oyama H, Yoshihara H (2005) MVL: Medical VR simulation library. In: Medicine Meets Virtual Reality (MMVR). Westwood JD et al. eds. IOS Press, pp. 273–276
15. Rissanen MJ, Kuroda Y, Kume N, Nakao M, Kuroda T, Yoshihara H (2006) Interactive authoring of example surgical procedures from recorded physics-based simulation. In: Computer Assisted Radiology and Surgery (CARS) (To appear)
16. Marshall C (1998) Toward an ecology of hypertext annotation. In: ACM Hypertext '98. ACM Press, pp. 40–49
17. Cotin S, Stylopoulos N, Ottensmeyer M, Neumann P, Rattner D, Dawson S (2002) Metrics for laparoscopic skills trainers: The weakest link! In: Lecture Notes In Computer Science; Vol. 2488: 5th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI), Part I. Dohi T, Kikinis R eds. Springer-Verlag, pp. 35–43

Real-Time Area-Based Haptic Rendering for a Palpation Simulator

Ki-Uk Kyung[1], Jinah Park[2], Dong-Soo Kwon[3], and Sang-Youn Kim[4]

[1] POST-PC Research Group, Digital Home Research Division, ETRI
161 Gajeong-dong, Yuseong-gu, Daejeon, 305-700, Korea
kyungku@etri.re.kr
[2] Computer Graphics & Visualization Laboratory, ICU, Korea
jinah@icu.ac.kr
[3] Human-Robot Interaction Research Center, KAIST,
373-1 Guseong-dong, Yuseong-gu, 305-701 Daejeon, Korea
kwonds@kaist.ac.kr
[4] School of Internet Media Engineering, Korea University of Technology and Education
307 Gajeonri, Byeongcheon-myeon, Cheonan-si, 330-708 Chungnam, Korea
sykim@kut.ac.kr

Abstract. In palpation procedure, medical doctors push and rub the organ's surface and they are provided the sensation of distributed pressure and contact force (reflecting force) for discerning doubtable portion. This paper suggests a real-time area-based haptic rendering model to describe distributed pressure and contact force simultaneously and present a haptic interface system to generate surface property in accordance with the haptic rendering algorithm. We represent the haptic model using the shape-retaining chain link (or S-Chain) framework for a fast and stable computation of the contact force and distributed pressure from a volumetric virtual object. In addition, we developed a compact pin-array type of tactile display unit and attached it to PHANToM^{TM} haptic device to complement each other. In order to evaluate the performance of the proposed scheme, related experiments have been conducted with non-homogenous volumetric cubic objects consisting of approximately 500,000 volume elements at a haptic update rate of 1000 Hz. The experimental results show that compared to the point-contact the area-contact provides the users with more precise perception of the shape and softness of the object's composition, and that our proposed system satisfies the real-time and realism constraints to be useful for virtual reality applications.

1 Introduction

Palpation is a diagnosis method to examine or explore the state of organs or tissues underneath by touching on the surface of a body. For constructing realistic palpation simulators, we must consider a high resolution volumetric model and real-time feedback force computation method. However, with these methods, it is difficult to

M. Harders and G. Székely (Eds.): ISBMS 2006, LNCS 4072, pp. 132 – 141, 2006.

simulate the haptic behavior of a high resolution volumetric object consisting of hundreds of thousands volume elements in real-time. In addition, for more realistic palpation simulators, tactile information coupled with kinesthetic information should be provided. There have been some efforts to develop palpation simulators which provide users with haptic feedback. However, up to now, the main type of target objects for palpation simulators has been a layered surface model [1,2,3] because it is difficult to compute feedback force from the volumetric model within a much shorter update. Recently, since is necessary to consider a volumetric haptic model and real-time force computation method to distinguish between soft portions and hard portions in an object by touching, several works have suggested simplified and efficient methods. Costa and Balaniuk [4] presented a new modeling method, the Long Element Method (LEM), for deformable objects in order to meet the computational requirements for volumetric objects. Balaniuk and Salisbury [5] introduced the Radial Elements Method (REM) based on the LEM for the simulation of deformable objects. Mollemans et al. [6] developed a tetrahedral mass-spring model for the fast deformation of the soft tissue. However, as before, more realistic and efficient volumetric deformation method is required. In aspect of haptic display, compared to force feedback interfaces, haptic technology concerning tactile display is only in its early stages. There have been several remarkable research works on developing tactile display with various actuating mathod[7,8,9,10,11], but in order to develop palpation simulator, more compact design and sufficient deflection are required.

In order to compute the deformation of a volumetric object and its feedback force in real time, we adopted a fast volume haptic rendering method based on the shape-retaining chain linked model (or the S-chain model)[12,13,14]. In this paper, a virtual object is represented by the S-chain model for real-time simulation, and the algorithm is expanded to the concept of contact area rather than point. Additionally, a small tactile display unit was developed and it was attached to PHANToM^{TM} in order to realize a palpation simulator providing kinesthetic force and distributed pressure simultaneously.

2 Area-Based Haptic Rendering

One of the dominant factors to detect a doubtable portion by palpation is distributed pressure on the finger pad. Fig. 1(a) shows a cross-sectional view of palpating the surface of an organ containing a hard portion. When a finger passes on the surface containing the portion, the hardness and the asymmetric distributed pressure are transmitted to the finger. Fig. 1(b) shows a conceptual drawing for the concurrent feedback of tactile feeling and feedback force. A tactile display provides the distributed pressure on a finger pad and a kinesthetic force feedback device, e.g. PHANToM^{TM}, provides the softness of the organ to the hand. If the tactile display is made small and light enough to attach at the end of the force feedback device, the haptic rendering based on haptic interaction area(HIA) can be realized.

In order to realize above mentioned scheme in volume rendered virtual environment, let's consider an orthogonal-crossectional view of a non-homogeneous deformable

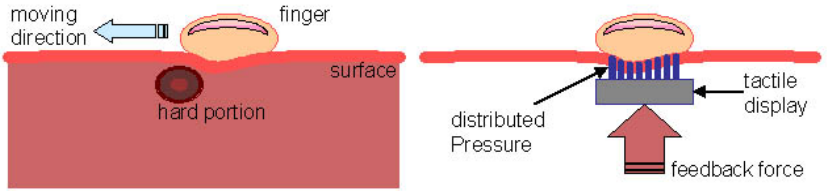


Fig. 1. Conceptual Drawing of Palpation (a): surface deformation by pressing and rubbing (b): providing the distributed pressure display using tactile display

object that interacts with a haptic device at an area (not a single point) as shown in Fig. 2.(a). This contact area includes four nodes (n_1, n_2, n_3, and n_4). The left two nodes(n_1 and n_2) are included in the soft portion and the right two nodes (n_3 and n_4) are included in the hard portion. Small black circles are interaction nodes with a user's finger pad. The forces (f_1, f_2, f_3, and f_4) generated at the nodes are calculated with a haptic model (a mass-spring model, an FEM, or the S-chain model), and they become elements for constructing distributed pressure. This distributed pressure is conveyed to a user's finger pad during palpating an object surface. The strokes of the pins are proportional to the stiffness of an object and user's penetration depth (Xp). Fig. 2.(b) shows a representative model when a human operator interacts with an object. In Fig. 2.(b), s_i is the stroke of the i-th pin in a tactile display, and Ko and k_s are the stiffness value of the target object and the human finger, respectively. For obtaining the realistic behavior of the target object, a human finger must be modeled with an FEM. However, due to a computational burden, we assumed that the human finger is modeled as simple spring. In this model, the force at each contact node can be computed by equation 2.1. It is used to make the shape of the user's finger pad and

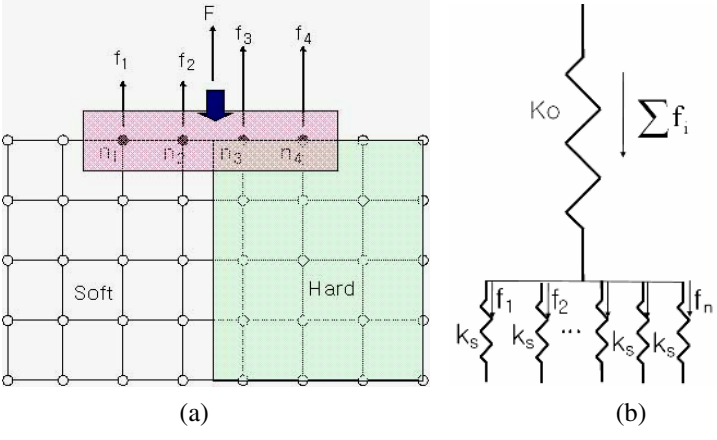


Fig. 2. (a) Non-homogeneous model composed of a soft material and hard material (b) Representative model when a human operator interacts with an object

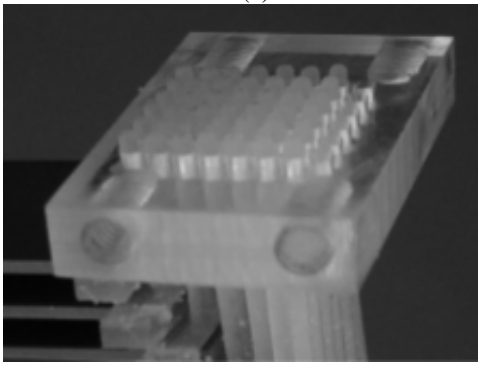
is utilized tactile information. The kinesthetic force could be derived from the real-time S-chain model[12,13,14].

$$f_i = (X_p + s_i) K_o = s_i k_s \tag{2.1}$$

3 Implementation of the Simulator System

In order to implement a new system which provides both kinesthetic feedback and tactile feedback simultaneously, we refered to the requirement for the developing of a

(a)



(b)

Fig. 3. Implemented Tactile Display (a) User grasps tactile display unit which connected to the force feedback device and put his or her index finder on the pin array part. (b) tactile display (case stripped).

compact tactile display unit[16]. Our tactile display unit and fixture were modified to be attached to the gimbal of PHANToM^{TM}. The unit is comprised of a pin array and piezoelectric bimorphs for normal stimulation (See Fig.3). Each piezoelectric bimorph has a displacement larger than 1 mm and a low operating input voltage (60V). In addition, its response time is on the order of millisecond (The system bandwidth is about 200Hz) and it can provide force up to 0.5N. Srinivasan has studied these skin mechanics and we could find the perceivable condition of stimulus [16]. The specifications of the tactile stimulator with piezoelectric bimorphs were verified to meet those requirements. We also experimentally confirmed that the actuator does not deflect by the normal rubbing on the surface to sense the texture. Johnson addressed that the modulation of SA 1 firing rates beginning at 0.5mm wide gaps parallels closely the human psychometric function for discriminating grating orientation [17]. From his study, we decided the diameter of pins and pin spacing. Piezoelectric bimorphs are clamped with 1 mm spacing and the 6 x 8 pin array is attached to the tip of each bimorph. The pin spacing is 1 ~ 1.5 mm and the diameter of each pin is 0.5 or 0.7 mm enabling the display of a texture of 8 mm wide. The size of the tactile display unit is approximately 50x14x18 mm and the weight is lower than 30 grams. The system generates counter balancing force in z-direction.

The haptic and graphic simulation programs were written in the form of a multi-threading technique, which enabled the haptic loop to run with the graphic loop at the same time and caused by optimal usage of the CPU capability. The multi-threading technique is quite useful in achieving real-time performance and stable haptic interaction. The haptic update rate of the proposed system is about 1 kHz. The haptic and the graphic threads share a same database that have the information of the object. Our system includes a 3D virtual model represented by the S-chain model for the real-time performance and PHANToM haptic device with tactile display for providing a sense of touch of the virtual object.

4 Empirical Study on Effectiveness of Proposed Palpation Simulator

4.1 Experimental Method

The proposed algorithm was applied to 3D cubic volumetric objects consisting of about 500,000 nodes. In our experiment, we set nodes on the bottom surface of the object as the constraint nodes and assume the shape of HIA is square. In order to compare the effectiveness of our proposed method with point-based method in palpation simulator, we performed a set of experiments with 20 human subjects to show that the subjects distinguish hard portions from the softer one using our approach. Six of twenty subjects are female, and the rest of the subjects are male. Each subject felt the sense of touch with a virtual object during 10 minute in order to orient the subject to our virtual environment.

After preliminary training, the subjects interacted with two 3D cubic objects represented by the proposed algorithm. At first, they explored the object with point-based haptic rendering and then explored the object with area-based haptic rendering with tactile feedback. In our experiments, soft portion is considered as normal tissue of

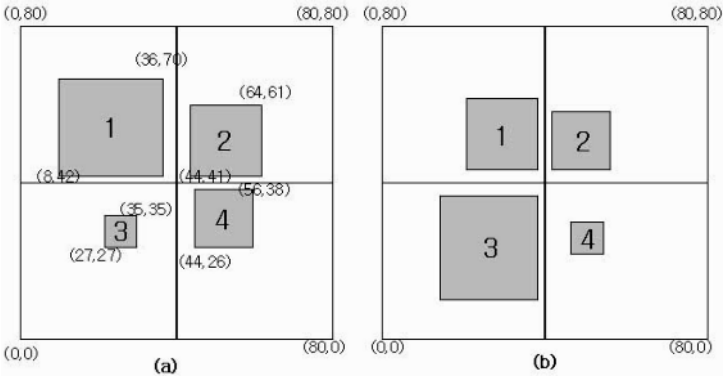
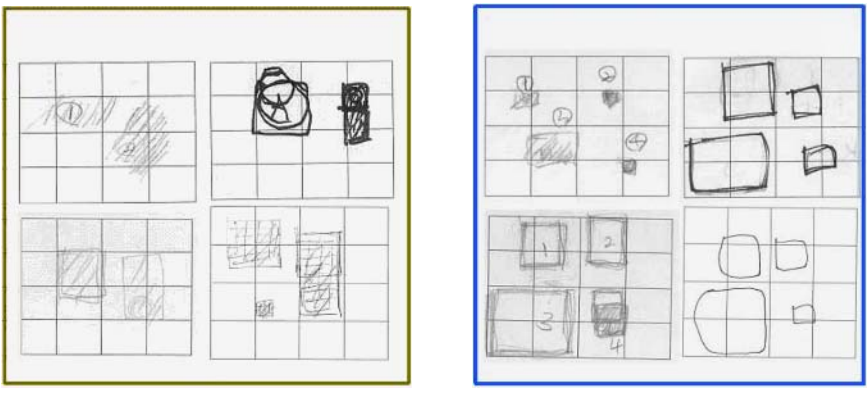


Fig. 4. Projection-view of the object (a) the object interacted with only force feedback device (b) the object interacted with force feedback device and embedded tactile display

human organ and hard potions are considered as tumors. The participants have been guided to draw the positions and sizes of the tumors that they have found. Fig. 4 shows distributions of hard portion(dark area). Different pattern has been suggested for point-based display and coupled display respectively. The subjects interacted with a 3D cubic object as shown in Fig. 4(a) and then they interacted with a cubic object as shown in Fig.4(b). The subject who explored the object as shown in Fig. 4(a) could understand the internal structure of the object. Therefore, we rotated the cubic object when the subjects interacted with the object as shown in Fig.4(b) via area-based haptic rendering.

4.2 Experimental Results and Analysis

In the experiment using point-based haptic rendering, all human subjects found the tumors 1 and 2. However, 8 persons out of 20 subjects could identify tumor 3 and only one person could find the tumor 4. Fig. 5.(a) shows the representative pictures drawn by the subjects when the point-based haptic rendering is applied. It is difficult to find the tumor 4 because the position of the tumor 4 is close to that of tumor 2. If the subjects push the soft portion between the tumor 2 and tumor 4, they feel the larger force at the interaction point than the other soft portion. Therefore, the tumors 2 and 3 appear to be one big tumor because of touch illusion. It is not easy to find the tumor 3 because its size is rather small. As shown in Fig. 5.(b), all human subjects were able to locate at least the tumors 1, 2, and 3 with the haptic information rendered by our area-based haptic rendering. 12 persons out of 20 subjects could find all tumors with the area-based haptic rendering. When the human subjects pushed or pulled the portion between the tumor 2 and tumor 3, they could easily distinguish tumors 2 and 3 (hard regions) from softer ones because of the distributed pressure (tactile feeling). The mean number that the human subjects found the tumors is 2.45 (61.25%) with the point-based haptic rendering, whereas the mean number is 3.6 (90.0%) with the area-based haptic rendering. Therefore, our area-based haptic rendering increases ability to distinguish hard portions from the softer ones.



(a) (b)

Fig. 5. (a) Representative pictures drawn by four subjects when the point-based haptic rendering is applied to the experiment. (b) Representative pictures drawn by four subjects when the area-based haptic rendering is applied to the experiment.

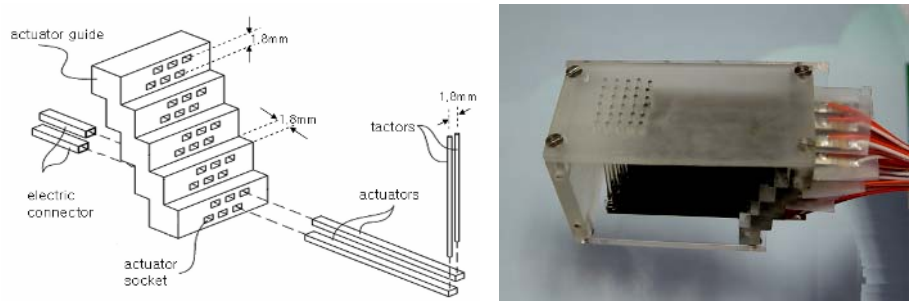
In palpation, it is important to find not only the number of tumors but also the location and size of tumors. To investigate precision in the size of the detected tumors, we compared size error between the size of tumor placed in the object and that of tumor that the subjects found. Table 1 shows the result of the size error. S_i means the size (in node unit) of the i-th tumor and S_{hi} means the size of the i-th tumor that the human subjects draw. The size error was calculated about tumors 1 and 2 in the case of point-based haptic rendering and about tumors 1, 2, and 3 in the case of area-based haptic rendering. From these results, our proposed haptic rendering scheme could be efficient to find the portions that have different material property in the object.

Table 1. Size discrepancy of the each tumor

	Point-based haptic rendering			Area-based haptic rendering		
	S_i	S_{hi}	Error	S_i	S_{hi}	Error
Tumor 1	784	267.8	33.8%	784	89.8	11.4%
Tumor 2	400	123.4	30.9%	400	41.3	10.3%
Tumor 3				144	20.1	14.0%

5 Conclusion and Discussion

This paper makes the following contributions to area-based haptic rendering: (1) describing a haptic rendering method that conveys the distributed pressure to the user's finger pad; (2) suggesting an interface that provides distributed pressure and contact force simultaneously; and (3) applying proposed force and tactile feedback method to volumetric deformable object. The experimental results show that the area-based haptic rendering method has excellent potential for identifying, recognizing, and classifying



(a) (b)

Fig. 6. A planar distributed tactile display. It has 5x6 individually operating pins. (a) The side view of the tactile display assembly. (b)Contact interface of the implemented tactile display.

target objects by touching. If our area-based haptic rendering scheme and haptic interface system are incorporated into haptic simulator, users will be provided immersive and realistic feeling in accordance with their manipulation of target objects.

It must be obvious that the proposed scheme combining force feedback and tactile feedback in this paper could be effective. However, since the tactile display in section 3 only controls columns independently, exploring in the y dimension is a different interaction modality than exploring in the x direction. In order to make up this discrepancy, a planar distributed tactile display has been developed[18] and applied to PC mouse[19]. Fig. 6 shows the apparatus that has 5 rows and 6 columns of pins (tactors) on 1.8 mm centers. The pin (diameter = 0.7 mm) attached to each actuator gives a normal deflection in response to electrically induced bending of the bimorph. The size of the cover case is 40 mm × 20 mm × 23 mm. The 30 stacked actuators are piezoelectric bimorphs. The maximum deflection is greater than 700μm and the bandwidth is about 350Hz. The blocking force of each is 0.06N. This work is an improvement of a previous work by Summers et al[20]. They developed a broadband tactile array (10×10) using piezoelectric bimorphs, and reported results for stimulation frequencies of 40Hz and 320Hz, with the maximum displacement of 50μm[18]. The specifications of our tactile stimulator with piezoelectric bimorphs were verified to ensure that it deforms the user's skin within 32SldB (sensation level in decibels above threshold) [16, 21]. Efforts to minimize the weight of the materials and wiring produced a finished design with a total weight of only ~11 grams. The contact area is about 9.7mm×7.9mm. The wide range of amplitudes and frequencies that this hardware supports makes it possible to apply almost any desired spatiotemporal pattern to the finger pad. As our future work, this tactile display will be attached to force feedback device and the more innovative and efficient algorithm will be applied using the integrated system.

Acknowledgment

This work has been supported by Korean Government (MOEHRD, R05-2003-000-10547-0) and "Development of Smart Haptic Interface Device Project (2005-S-064, KIMC)".

References

1. Burdea, G., Patounakis, G., Popescu, V, Weiss, R.: Virtual Reality-based Training for the Diagnosis of Prostate Cancer. IEEE Transactions on Biomedical Engineering, Vol. 46(10). (1999) 1253-1260
2. Crossan, A., Brewster, S.A., Glendye, A.: A Horse Ovary Palpation Simulator for Veterinary Training. Proceeding of PURS 2000, (2000) 79-86
3. Baillie, S., Crossan, A., Reid, S., Brewster, S..: Preliminary Development and Evaluation of a Bovine Rectal Palpation Simulator for Training Veterinary Students. Cattle Practice, Vol 11(2), (2003) 101-106
4. Costa, I.F., Balaniuk, R.: LEM-An Approach for Real time Physically Based Soft Tissue Simulation. IEEE International Conference on Robotics & Automation, (2001) 2337-2343
5. Balaniuk, R., Salisbury, K.: Soft-Tissue Simulation Using the Radial Elements Method. International Symposium on Surgery Simulation and Soft Tissue Modeling (IS4TM), Juan-Les-Pins, France. (2003) 48-58
6. Mollemans, W., Schutyser, F., Cleynenbreugel, J.V., Suetens, P.: Tetrahedral Mass Spring Model for Fast Soft Tissue Deformation. IS4TM, (2003) 145-154
7. Shimojo, M., Shinohara, M., Fukui, Y.: Human Shape Recognition Peformance for 3-D Tactile Display. IEEE Transaction on Systems, Man, and Cybernetics – PART A: Systems and Humans, Vol. 29(6), (1999).637~644
8. Asamura, N., Yokomu, N., Shinoda. H.: Selectively Stimulating Skin Receptors for Tactile Display. IEEE Computer Graphics and Applications, Vol. 18(6). (1998).32-37
9. Hayward, V., Cruz-Hernandez, M.: Tactile Display Device Using Distributed Lateral Skin Stretch. Proceeding of. ASME, Vol. DSC-69-2. (2000) 1309-1314
10. Konyo, M., Tadokoro, S., Takamori, T.: Artificial Tactile Feel Display using Soft Gel Actuators. Proceeding of. IEEE ICRA 2000, (2000). 3416-3421
11. Kajimoto, H., Kawakami, N., Maeda, T., Tachi, S.: Tactile Feeling Display using Functional Electrical Stimulation. Proceeding of ICAT '99, (1999) 107-114
12. Kim, S.Y., Park, J., Kwon, D.S.: The Real-Time Haptic Simulation of a Biomedical Volumetric Object with Shape-retaining Chain Linked Model. IEICE Transaction on information and systems, Vol.E88-D(5). (2005) 1012-1020
13. Kim, S.Y., Park, J., Kwon, D.S.: Real-time Haptic Rendering of High-Resolution Volumetric Deformable Object in a Collaborative Virtual Environment", Advanced Robotics vol(19), No(9), (2005) 951-975
14. Kim, S.Y., Park, J., Kwon, D.S.: Area-Contact Haptic Simulation", LNCS(Lecture Notes in Computer Science) vol (2678), pp 108-120. 2003
15. Kyung, K.U., Choi, H., Kwon, D.S., Son, S.W.: Interactive Mouse Systems Providing Haptic Feedback During the Exploration in Virtual Environment. Lecture Notes in Computer Science. Vol. 3280 (2004) 136-146
16. Srinivasan, M.A.: Surface Deflection of Primate Fingertip under Line Load. Journal of Biomechanics, Vol.22(4). (1989) 343-349
17. Johnson, K.O.: Tactile Spatial Resolution. I. Two-point discrimination, gap detection, grating resolution, and letter recognition. Journal of Neurophysiology, Vol. 46(6). (1981) 1177-1191
18. Kyung K.U., Ahn, M., Kwon, D.S., Srinivasan, M.A.: A Compact Planar Distributed Tactile Display and Effects of Frequency on Texture Judgment. Advanced Robotics, Vol. 20(5). (2006) 563-580

19. Kyung, K.U., Kwon, D.S., Yang, G.H.: A Novel Interactive Mouse System for Holistic Haptic Display in a Human-Computer Interface. International Journal of Human Computer Interaction. Vol. 20(3). (2006) 247-270
20. Summers, I.R., Chanter, C.M.: A Broadband Tactile Array on the Fingertip. Journal of the Acoustical Society of America. Vol. 112. (2002_ 2118-2126
21. Diller, T.T., Schloerb, D., Srinivasan, M.A.: Frequency Response of Human Skin in Vivo to Mechanical Stimulation," RLE Technical Report, MIT 648 (2001)

New Methods for Video-Based Tracking of Laparoscopic Tools

Alicia M. Cano, Pablo Lamata, Francisco Gayá, and Enrique J. Gómez

Grupo de Bioingeniería y Telemedicina (GBT),
ETSIT, Universidad Politécnica de Madrid
c/ Ciudad Universitaria s/n, 28040 Madrid, Spain
{acano, lamata, fgaya, fpozo, egomez}@gbt.tfo.upm.es
http://www.gbt.tfo.upm.es

Abstract. New methods for video-based tracking of laparoscopic instruments are presented. This aims to contribute for an objective evaluation of surgical skills, and for enabling augmented reality features in laparoscopic surgery. An optical geometrical model of the laparoscopic setting is developed, what is the basis for two methods proposed for assessing the 3D position of tools' tips. The first exploits the properties of the vanishing point of the tool in the image plane, and the second the apparent tool width with the distance to the camera. Ground-truth sequences are recorded and analysed, and preliminary results demonstrate the feasibility of these methods. Video-based approach constitutes a very promising alternative to mechanical, optical or electromagnetic tracking devices.

1 Introduction

Laparoscopic surgery is currently performed in clinical routine at hospitals. Nevertheless there is a crescent pressure to have transparent training programs, with objective metrics of surgical skill and alternatives that might be used at any time [1]. Laparoscopic training programs lack objective metrics for skills' assessment. Analysis of path and movements travelled with laparoscopic tools is an important source of objective parameters [2].

On the other hand laparoscopy offers a reduced workspace and a limited sensory interaction. One interesting improvement in this surgical technique would be the use of augmented reality techniques. This could provide useful information for guiding the surgical procedure or preventing delicate areas. A clear requisite for such applications is the knowledge of the position of laparoscopic tools in real time.

Therefore, the problem addressed is the tracking of laparoscopic instruments with only the information of the conventional video signal captured by an endoscope. This aims to contribute both for an objective evaluation of surgical skills, and for enabling the possibility of augmented reality features in laparoscopic surgery. Another important benefit is the improvement of studies for biomechanical characterization by reducing the complexity and cost of physical tracking instruments, which can be very bulky [3]. This is especially interesting for in-vivo studies in the laparoscopic operating theatre.

M. Harders and G. Székely (Eds.): ISBMS 2006, LNCS 4072, pp. 142 – 149, 2006.

Video-based tracking of laparoscopic tools can be decomposed in two main problems: (1) extraction of image parameters with segmentation techniques and (2) estimation of the 3D coordinates of the end of the tool with these parameters. There are some works already published in the literature about these two aspects.

Voros et al [4] have created an automatic and robust method to detect the edges of laparoscopic tools in real sequences. This method is based in a Hough transform fastened by the restriction of the localization of the trocar point. On the other hand, Tonet et al [5] have designed a method which is able to localize an instrument in 3D with regard to the position of the camera with an extra marker in the instrument. They propose an empirical estimation of the depth of the instrument based on the position and orientation of the instruments.

This work contributes with a geometrical optical model of the laparoscopic setting, which is the basis for two methods proposed for assessing the 3D position of tools' tips. This model is a conceptualization of the problem that encompasses former works [4,5].

2 Material and Methods

Two methods for assessing the 3D position of a tool tip are proposed. They depart from the knowledge of the edges of the instrument and its 2D localization of the tip, what would be the result of a video segmentation process. Moreover, the knowledge of the Field of View (FoV) of the camera permits to determine spatially the projective line (line from camera to the spatial point) for any image point.

2.1 Vanishing Point Based Method

This method gets the 3D coordinates of the tip of the tool through the information about the surgical setting (position of camera and trocar), the optical characteristic of the camera (field of view), and the 2D parameters extracted from the segmentation process of each frame (edges of tools and their tip point). It is developed from the geometrical model depicted in Fig. 1.

In this perspective, the point at infinity, also called vanishing point (V), is where the edge lines of the instrument meet in the image plane. The projective line of this point has the same direction vector of the tool. This is due to the fact that all 3D lines parallel to the tool ($\overline{TP}$) converges at this vanishing point in the image plane [6], and therefore the 3D line containing the camera point does ($\overline{CV}$, its projection is a single point). In this way, the instrument edges give information about the tool direction. ($\overline{CV}$ and $\overline{TP}$ are parallel vectors in Fig. 1)

Using the main idea, it is also built the optical ray to the tip of the tool, getting the director vector to the tip of the instrument from the camera. Along these lines, 2D localization of the tip and the FoV provide information about the direction vector camera-tip of the tool. (See $\overline{CP}'$ vector in Fig. 1)

Placing both vectors (director vector of the tool and the director vector camera-tip of the instrument) in the trocar and camera point (which are known), respectively, the position of the tip of the instrument is solved, like the intersection between both lines.

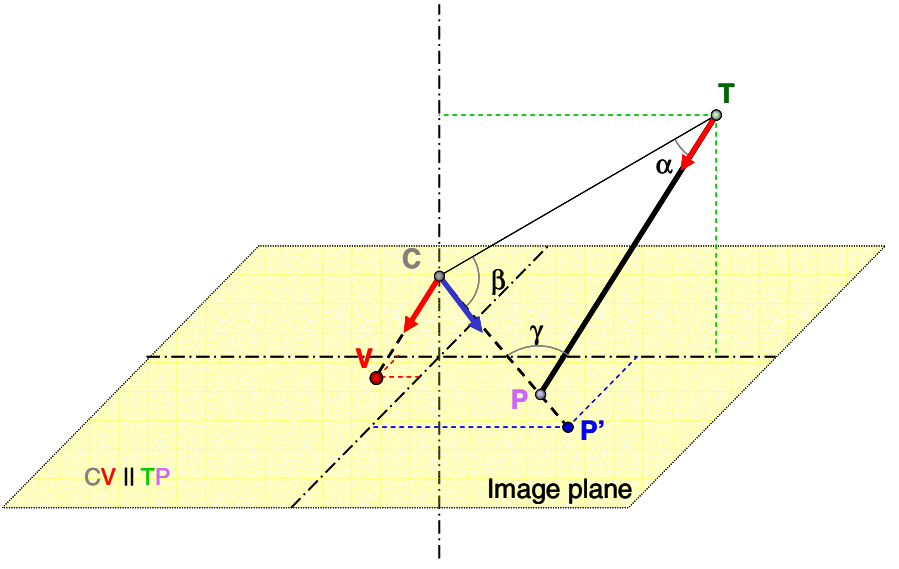


Fig. 1. Geometrical optical model of the working scene. C: optical center, TP vector: surgical tool, P: tool tip. P': projection of the tool tip. V: vanishing point.

Finally the vector $\overline{CP}$ can be expressed:

$$\overline{CP} = \left|\overline{CT}\right| \cdot \frac{\overline{CP'}}{\left|\overline{CP'}\right|} \cdot \frac{\sin(\alpha)}{\sin(\gamma)} = \left|\overline{CT}\right| \cdot \vec{u}_{CP'} \cdot \frac{\sin(\alpha)}{\sin(\gamma)} \tag{1}$$

2.2 Apparent Tool Size-Based Method

This method determines the tools' location according to the diameter size of the instrument in the image and the angle defined by the projective lines to two opposite edge points.

Knowing the real distance between these points (diameter of the instruments) and the angle defined by the camera as vertex and the two opposite edge points placed in the chosen position in the tool (α in the Fig. 2), it is possible to calculate the distance from this position to the camera (height of a isosceles triangle once known the basis). This is the module of the vector that joins the tip of the instrument (P) with the camera (C). Moreover, the projective line to this position is known, so it is possible to determine the coordinates of the tip of the vector.

Consequently the $\overline{CP}$ vector is calculated:

$$\overline{CP} = \left|\overline{CP}\right| \cdot \frac{\overline{CP'}}{\left|\overline{CP'}\right|} = \frac{Physical\,Tool\,Width\,(mm)}{Image\,Tool\,Width\,(pix)} \cdot \overline{CP'} \tag{2}$$

In conclusion, we can apply these methods thanks to the geometry of the endoscopic instruments. These instruments present a cylindrical shape which allows

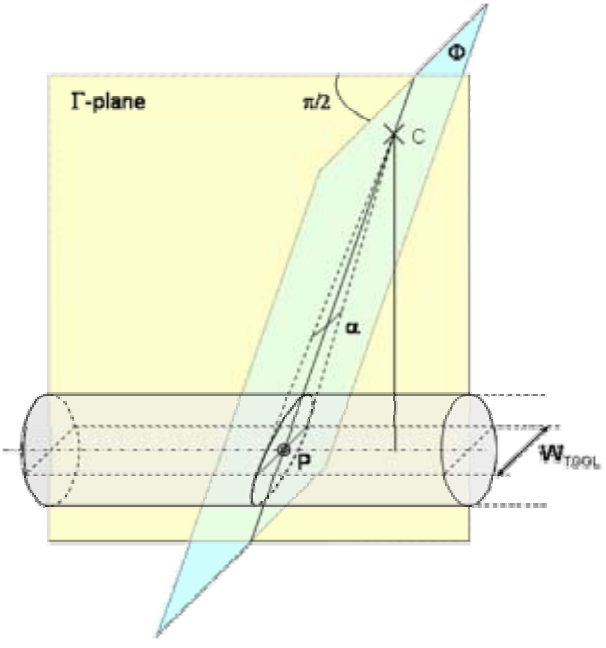


Fig. 2. Section of the instrument according to the projection line to the point P

knowing its diameter in the image plane in all situations and the perspective of its edge gives information about its direction.

2.3 Validation Process

The validation of both methods is done using two sequences of movements acquired in laboratory. This is necessary for the definition of a ground truth of the positions of the tools' tip in each frame without an external tracking device.

A laparoscopic setting is built, see Fig. 3, emulating the same conditions than a real surgery: relative positions between camera and trocar points are carefully respected. The key idea is to introduce a physical guide for the movements that relates the 2D coordinates with the depth. This physical guide is an inclined board (30º) placed at a given distance from the optical focus point (C point in the model). 3D trajectory of the tool keeping the tip alongside this board is defined with only the 2D information gathered from the segmentation process. Therefore a ground-truth of 3D coordinates from 2D information is defined and used to validate proposed methods. Two sequences are recorded (see Fig. 3): one with a constant depth (Y movement) and other with a variable depth (XZ movement). Each sequence has about 275 frames, around 10 seconds of recording.

Segmentation of each frame of the sequence is necessary for arriving into the estimation of the 3D coordinates. This is roughly done with digital derivate masks and temporal information gathered from former frames.

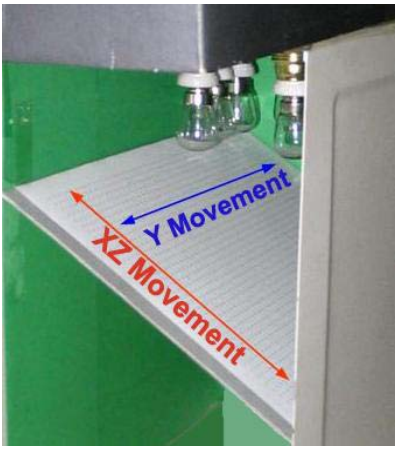


Fig. 3. Setting for recording lab sequences valid for a ground-truth definition

The recording was carried out with a home-videocamera and the sequence was analysed with Matlab program on a Pentium IV (3.00 GHz, 512 Mb RAM) computer.

3 Results

The two lab sequences have been analysed with the two proposed methods. Estimated 3D coordinates have been estimated in a total of 550 frames. Accuracy has been assessed comparing these estimations with the ground-truth trajectory by two means, regression lines for the variable-depth sequence (see Fig. 4) and error characterization for both sequences (mean absolute error and standard deviation, see Table 1 and Table 2).

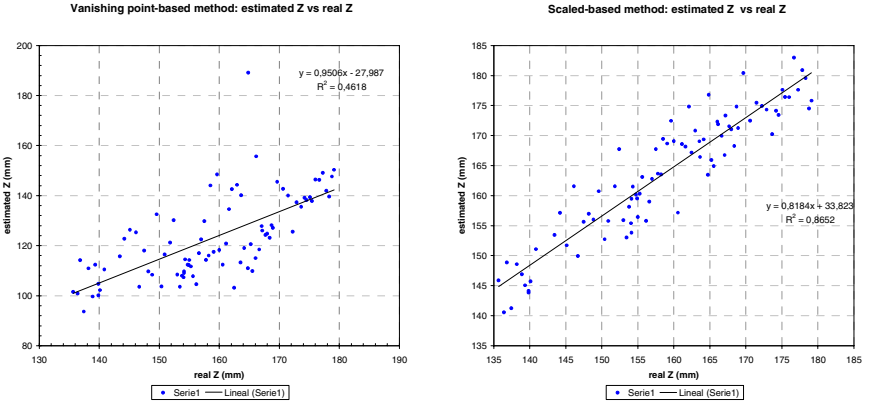


Fig. 4. Estimated Z coordinate versus ground-truth Z coordinate with both methods in variable-depth sequence

Table 1. Error characterization for the variable-depth sequence (SD: standard deviation)

	Vanishing point			Apparent size		
	|X error|	|Y error|	|Z error|	|X error|	|Y error|	|Z error|
mean (mm)	5.59	0.84	33.60	0.67	1.59	6.65
SD (mm)	3.55	0.97	9.26	0.56	0.52	3.85

Table 2. Error characterization for the constant-depth sequence (SD: standard deviation)

	Vanishing point			Apparent size		
	|X error|	|Y error|	|Z error|	|X error|	|Y error|	|Z error|
mean (mm)	2.20	7.26	43.69	0.68	1.90	3.00
SD (mm)	1.12	7.28	16.30	0,13	0.76	2.19

The "apparent tool size-based" method has shown a better behaviour than vanishing point-based one in all comparison metrics: R^2 regression coefficient, mean absolute error and standard deviation of absolute error (see Table 1 and Table 2). This method also shows a better behaviour in the constant-depth sequence compared to the variable-depth sequence. The vanishing point-based shows similar features in both sequences.

4 Discussion

Two methods have been proposed for assessing the 3D coordinates of the tip of laparoscopic tools. They are simply based in an analysis of the video sequences, and constitute a very interesting alternative to mechanical, optical or electromagnetic tracking devices.

Results have demonstrated the feasibility of these methods and of the video-based alternative, as recently concluded in a previous work [5]. Moreover, the developed geometrical optical model has provided an insight of the problem. This has enabled the development of the vanishing point-based method, which uses the information of the orientation of the lateral edges of the tool. Previous empirical results and formulas [5] are now understood from the right perspective provided by this model. Moreover, projection of the insertion point in this model is a crucial piece of information for restricting the Hough transform for segmentation purposes, as already addressed in [4].

The methods have been validated in two lab sequences and not in real laparoscopic ones. This has been required for the definition of a ground truth of the positions of the tools' tip in each frame of the sequences. This is a right alternative because the aim was to assess the feasibility of assessing the 3D coordinate with the 2D information extracted from a frame, and not to develop a robust segmentation technique for obtaining this 2D information from a laparoscopic frame.

The accuracy of proposed methods, which has been assessed in the two analysed sequences, could be qualified as moderate. Standard deviation of absolute errors has been lower than 16.3 mm and 3,85 mm in respectively the "vanishing-point-based" and "apparent tool size-based" methods. It has to be regarded that this accuracy depends on the result of the segmentation process, which could be a little rough in the implementation that has been used.

Both methods have showed a mean absolute error in the estimation of the 3D coordinates. The origin and behaviour of this bias is probably some uncertainty in the parameters of the models, like the relative position between of the camera focus point and the trocar point. There are also some corrections that need to be done in order to prevent some optical distortions.

The more noisy estimation of the Z coordinate of the "vanishing-point-based" method has been explained with its sensibility to errors in the slope of detected tools' edges that determine the vanishing point. This is more critic in those frames in which there is only a small portion of the tool depicted. However, the "apparent tool size-based" method is quite noisy in those situations in which the tool is far from the camera, when its size is small and the pixel quantification error is bigger. Therefore a combination of both methods, together with a robust image segmentation process, seems to be a good alternative to be studied in future work in order to get a good accuracy.

According to the results, both methods present a low accuracy in the present state of development. However, most of the laparoscopic surgery applications could need accuracy around 2-3 mm., for example in laparoscopic training programs. Nevertheless the required accuracy depends on the desired objective of the particular application, it rarely needs accuracy better than 0,5 mm, due to factors as tool movement magnification because of pivoting on the insertion point.

Other interesting issues to be addressed are the automatic determination of the 3D coordinates of the trocar and camera points, the tracking of camera movements, and the optimization of all processes in order to reach a real-time application. This will eventually lead to a complete system for tools tracking.

5 Conclusion

Laparoscopic tools' tracking is possible with an analysis of conventional video sequences. Developed geometrical optical model provides an insight of this problem. Preliminary results of the two proposed alternatives for assessing the 3D coordinates with 2D information are very promising.

Acknowledgement

This research work has been partially funded by the SINERGIA Thematic Network (G03/135) of Spanish Ministry of Health.

Authors would like to thank the valuable contributions of Patricia Sánchez and Samuel Rodríguez.

References

1. Kneebone,R.: "Simulation in surgical training: educational issues and practical implications". Medical Education, 37(3):267-277, 2003.
2. Rosen,J., J.D.Brown, L.Chang, M.N.Sinanan, B.Hannaford: "Generalized Approach for Modeling Minimally Invasive Surgery as a Stochastic Process Using a Discrete Markov Model". Biomedical Engineering, IEEE Transactions on, 53(3):399-413, 2006.
3. Immersion: Digiziting. www.immersion.com/digitizer/
4. Voros S., Orvain E., Cinquin P., 'Automatic detection of instruments in laparoscopic images: a first step towards high level command of robotized endoscopic holders'. BIOROB'06 1st IEEE/RAS-EMBS Int. Conf. On Biomedical Robotics and Biomechatronics, Pisa, Italy, February 2006.
5. Tonet O., Armes T.U., Magali G., Dario P. 'Tracking Endoscopic Instruments Without Localizer: Image Análisis-Based Approach' Stud. Health Technol. Inform., 119:544-549, 2005.
6. Casas J.'Óptica'. University of Zaragoza. Pons, 1994.

Mechanical Characterization of the Liver Capsule and Parenchyma

Marc Hollenstein[1], Alessandro Nava[1], Davide Valtorta[1], Jess G. Snedeker[2], and Edoardo Mazza[1]

[1] Institute of Mechanical Systems, ETH Zurich, CH-8092 Zurich, Switzerland
{mhollenstein, anava, davidev, emazza}@ethz.ch
http://www.zfm.ethz.ch/e/biomechanics
[2] Institute for Biomedical Engineering, ETH Zurich & University of Zurich, CH-8044 Zurich, Switzerland
snedeker@ethz.ch

Abstract. Internal organs are heterogeneous structures, both on the macro- and on the micro-scale. However, they are often modeled as homogeneous solids with uniform material properties. In this light, this work investigates the impact of the liver capsule on the integral behavior of the organ by means of in vitro tests and computer simulations. The stiffness of bovine liver obtained in tissue aspiration experiments differed by a factor of 2 to 3 when the capsule was removed. As a first step, the capsule was implemented as separate structure in a finite element model of the organ undergoing tissue aspiration. The finite element simulations are in good agreement with the experimental results.

1 Introduction

The identification of the mechanical properties of soft biological tissues is essential to the understanding of their functions, and therefore to a number of medical applications such as diagnosis, surgery planning and training of surgical procedures with virtual reality-based simulators (Picinbono et al. [1], Snedeker et al. [2] and Szekely [3]). Several experimental methodologies have been proposed for the mechanical testing of soft tissues (Brown et al. [4], Kalanovic et al. [5], Nava et al. [6], Nava et al. [7], Ottensmeyer [8] and Snedeker et al. [9]), but only few quantitative data are available for the in vivo behavior of human organs (Kauer et al. [10], Mazza et al. [11], Nava et al. [12] and Carter et al. [13]). Current methodologies describe the entire organ complex as a homogeneous material, neglecting the individual component tissues.

Internal organs essentially consist of a functional vascularized internal part (parenchyma), and an external capsule (stroma). The capsule is a thin but tough fibrous supporting connective framework of densely interwoven collagen fibers. The capsule primarily serves the structural integrity of the organ. Previous studies investigated the histology and quasi-static properties of the renal capsule (Yamada [14], Herbert et al. [15] and Farshad et al. [16]) and parenchyma (Yamada [14], Melvin et al. [17] and Farshad et al. [16]). Recently, Snedeker et al. [9] investigated the dynamic properties of the renal capsule at strain rates associated with blunt abdominal trauma. However, little data exist on the mechanical properties of the liver capsule (Yamada [14] and Arnold et al. [18]).

M. Harders and G. Székely (Eds.): ISBMS 2006, LNCS 4072, pp. 150 – 158, 2006.

The preliminary study presented in this paper has been undertaken in order to quantify the influence of the capsule on the mechanical behavior of internal organs. For this purpose, capsule and parenchyma of a bovine liver have been characterized individually and compared with their integral "parenchyma-capsule" response. These investigations are for example essential for the interpretation of in vivo aspiration experiments performed during open surgery (Mazza et al. [11], Mazza et al. [19] and Nava et al. [12]).

The capsule was tested in quasi-static uniaxial tension, whereas aspiration experiments were used to characterize the intact organ and the parenchyma. Constitutive equations were derived for implementation in numerical simulations. Finite element (FE) simulations of the liver undergoing aspiration, treating the capsule and parenchyma as separate structures, are presented.

2 Methods

The importance of the capsule in modeling the mechanical behavior of internal organs has been investigated using the example of a bovine liver. All experiments were conducted ex vivo and were performed in accordance with Swiss federal ethical research standards.

An intact bovine liver was obtained from the slaughterhouse immediately following animal euthanasia. The liver was transported on ice and kept moist wrapped in a physiological saline soaked surgical cloth at 4°C until the samples were prepared. The samples were tested within 8 hours of animal euthanization.

2.1 Uniaxial Tensile Testing of the Liver Capsule

For the specific characterization of the liver capsule, samples were extracted and tested in a standard uniaxial tensile test setup. The load-displacement diagrams were recorded with a Zwick 1456 universal uniaxial test machine (Zwick Inc., Ulm, Germany). A custom-made bio-chamber was used to simulate "in vivo" conditions. The vertical test setup is shown in Fig. 1A.

The capsule samples were prepared according to a method similar to Herbert et al. [15] and Snedeker et al. [9]. Samples were delicately excised using a surgical scalpel and a thin copper sheet template. The samples were approximately 15 mm in width and 50 mm in length. After incision, the trace of the scalpel in the underlying liver cortex was measured to verify a uniform sample width. The sample thickness was assumed equal within the same sample, and was assessed by averaging three measurements taken at random locations using a micrometer caliper. The capsule samples were then gently peeled off from the liver cortex using the fingers and surgical forceps. To facilitate the handling of the liberated samples, they were carefully placed and spread with the fingers on a sheet of ordinary paper, yielding a membrane sample with paper backing. The capsule tissue adheres to the paper due to surface tension. Once on the paper, the underside of the capsule was cleansed thoroughly of the parenchymal remainders by softly applying a gauze. Finally, the samples were immersed in physiological saline at room temperature until testing. The time between sample preparation and mechanical test never exceeded 1 hour.

For testing, the paper-backed samples were comfortably fixed outside the test setup on clamps equipped with sandpaper, leaving an initial gage length of at least 40 mm. The paper-backing was then carefully removed over the gage length with forceps. The integrity of the clamp interfaces, i.e. slip-free condition, was verified by Snedeker et al. [9] using video data. Immediately after clamping, the capsule samples were introduced into the bio-chamber. During testing, the samples remained entirely immersed in the physiological saline. A heating plate in combination with a thermocouple controlled the solution temperature of 37°C (bovine body temperature). The tensile force was measured with a 50 N load cell at a specified accuracy of ±0.25% of the effective force. The specified positioning accuracy of the clamps is ±3 μm. A xy-bench was installed on the load cell to align and prevent from shearing the sample during testing. A preload of 0.2 N was applied at the beginning of the test. Tests were carried out under displacement-controlled conditions, with a strain rate of 0.5% per second. In order to obtain a "preconditioned" state of the tissue samples, i.e. stable response to several loading-unloading cycles, the samples were cycled 10 times between 0 and 15% nominal strain prior to ultimate loading until tissue rupture.

In the present work, test results are reported as nominal stress and nominal strain, whereas the nominal stress is reported as the axial force divided by the original cross-sectional area of the undeformed sample (i.e. the undeformed sample width before extraction multiplied by the average thickness measured), and the nominal strain was calculated using measurements of the clamp displacement. Ideally the nominal quantities would be specified according to the "in vivo" condition of the tissue. This is assumed in this study, but was not verified.

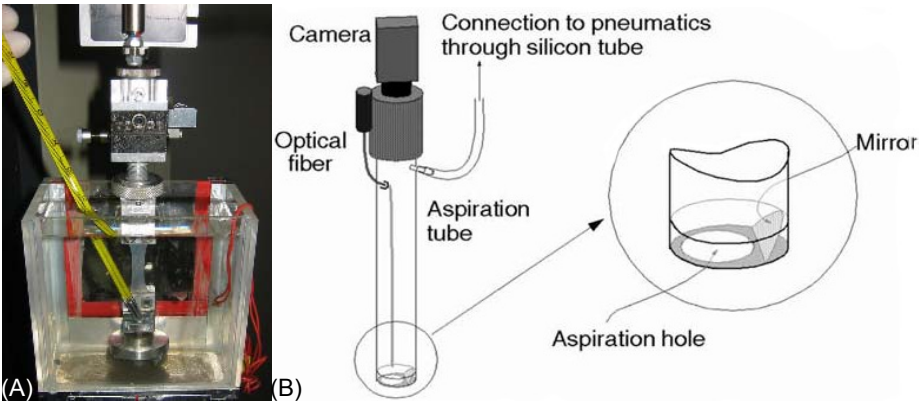


Fig. 1. (A) Capsule membrane sample exposed to uniaxial tensile testing inside the bio-chamber, and (B) aspiration device and principle of working

2.2 Constitutive Equations for the Capsule

With the simplifying assumptions of incompressibility, isotropy and non-dissipative behavior, the material response is characterized here by the so called reduced polynomial formulation of the strain energy potential U, shown in Eq. 1 (Yeoh [20]). The resulting relation between nominal uniaxial stress σ and stretch λ is given in Eq. 2.

$$U = \sum_{i=1}^{N} C_{i0} \left(\bar{I}_1 - 3 \right)^i \tag{1}$$

$$\sigma = 2\left(\lambda - \lambda^{-2} \right) \sum_{i=1}^{N} i C_{i0} \left(\lambda^2 + 2\lambda^{-1} - 3 \right)^{i-1} \tag{2}$$

C_{i0} are the material parameters and N is the order of the polynomial fit. $\bar{I}_1$ is the first deviatoric strain invariant. The parameters of the reduced polynomial form were determined for $N = 5$. Fitting was carried out with respect to the "preconditioned" final loading cycle. Constitutive equation parameters were defined after averaging of all data sets.

2.3 Aspiration Device

2.3.1 Setup and Experiments

The aspiration device shown in Fig. 1B has been developed by Vuskovic [21] and recently improved for new experiments, see Mazza et al. [11] and Nava et al. [6]. The device has been designed for in vivo applications. It consists of a tube in which the internal pressure can be controlled according to a desired pressure law. The experiment is performed by (i) gently pushing the tube against the tissue to ensure a good initial contact, and (ii) creating a time variable vacuum inside the tube so that the tissue is sucked in through the aspiration hole (diameter of 10 mm).

A complete description of the deformed tissue can be given by simply monitoring the side-view profile of the tissue during its deformation. The images of the side-view are reflected by a mirror and are captured by a digital camera. Time histories of measured pressure and deformation profiles are the input data used to evaluate the mechanical properties and to determine the constitutive model.

Two different sets of experiments were performed on the same bovine liver by applying the identical pressure history: (i) 4 experiments on different locations on the surface of the intact organ (in presence of the capsule), and (ii) 4 experiments on different locations after removal of the capsule. The organ was not perfused. The mechanical parameters were extracted from average displacement curves.

2.3.2 Mechanical Modeling

A phenomenological model is determined based on three-dimensional continuum mechanics analysis. The material is modeled as an incompressible, isotropic continuum with so called quasi-linear viscoelastic equations (Fung [22]). Viscoelasticity is taken into account by applying relaxation coefficients to the constants that define the energy function, Eq. 3. As for the capsule, the reduced polynomial form has been chosen for the strain energy potential U, Eq. 1.

$$C_{n0}(t) = C_{n0}^{\infty} \frac{\left(1 - \sum_{k=1}^{K} \bar{g}_k^P \left(1 - e^{-t/\tau_k} \right) \right)}{\sum_{k=1}^{K} \bar{g}_k^P} \tag{3}$$

In Eq. 3 the long term elastic module C_{n0}^{∞}, $\bar{g}_k^P$ and τ_k are material parameters to be determined from the experimental data. In the current implementation the second order

of the series expansion for the strain energy potential ($N = 2$) and at the second order for the Prony series ($K = 2$) are used.

The finite element method is applied to solve the so called ”inverse problem”: the experiment is simulated by a FE-model in which the time dependent aspiration pressure is imposed as kinetic boundary condition; the material constants are determined iteratively from the comparison of calculated and measured soft tissue deformation.

In order to evaluate the influence of the capsule on the mechanical behavior, two different models were applied: (i) a homogeneous model (Model A), and (ii) a bilayer model (Model B, see Fig. 4A), including as its top layer the organ capsule (thickness of 80 μm) and as the underlying layer the “homogeneous” parenchyma. Model A was used to analyze (i) the experimental curves of liver with capsule, and (ii) the data without capsule. The material parameters of the parenchyma were determined from (ii). Validation of the parameters (for capsule and parenchyma) is performed by simulating the aspiration experiment with Model B and comparing the displacement history to the experimental findings for liver with capsule.

3 Results and Discussion

The capsule samples were extracted from random locations of one bovine liver. A total of 5 samples were tested, and nominal stress-stretch data are reported for the preconditioned state of each sample. These preliminary results are not sufficient for proper statistical analysis, but allow a first insight on the mechanical properties of the liver capsule.

The tests yielded approximately bilinear nominal stress-stretch characteristics, Fig. 2. The curves are characterized by a relatively low initial stiffness, with a nearly linear stress-strain curve up to 3% strain. Considerable stiffening occurs between 5 and 10% strain, an effect typically attributed to the progressive fiber recruitment. Once a majority of fibers is engaged, the curves become nearly linear again, almost until tissue rupture.

The bilinear nature of the response is characterized with four scalar quantities: the low-strain elastic modulus E_1 and the high-strain elastic modulus E_2, the ultimate stress and strain denoted by σ_{max} and ε_{max}, respectively. A summary of the values is presented in Table 1.

Table 1. Results of the bovine liver capsule in tension (average and standard deviation are indicated)

Thickness [μm]	93 ± 6
E_1 [MPa]	1.1 ± 0.2
E_2 [MPa]	38.5 ± 4.9
σ_{max} [MPa]	9.2 ± 0.7
ε_{max} [%]	35.6 ± 5.2

The transition from low to high stiffness is quite consistent between samples, with the transition point, i.e. "locking stretch", occurring around 8% strain. The data show good repeatability. The evaluated low-strain elastic modulus E_1 varied less than 6.5% and the high-strain modulus E_2 less than 8%. The high-strain modulus agrees to a great extent with the published data of Snedeker et al. [9] and Yamada [14] for the similar tissue kidney capsule: Snedeker reported corresponding values of 41.5 MPa for human and 35.9 MPa for porcine kidney capsule, whilst Yamada obtained 38 MPa for cadaveric human kidney capsule. The average failure stress was calculated as 9.2 ± 0.7 MPa, which is in good agreement with findings of Snedeker, who reported a value of 9.0 ± 2.9 MPa. The measured thickness of the liver capsule samples varied as much as 5% within a given sample, and as much as 10% within the given liver, with an average value of 93 μm, what is almost double the values presented by Snedeker for the kidney capsule thickness.

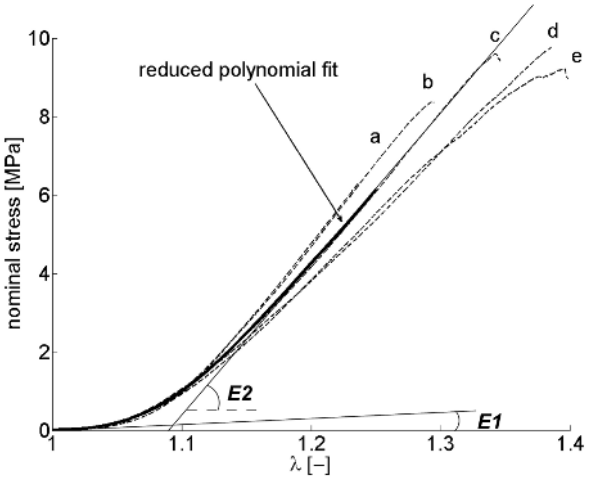


Fig. 2. Bovine liver capsule in uniaxial tension: the five samples *a* through *e* (*dashed*) and the resultant fit of a reduced polynomial hyperelastic material (*continuous bold*). The capsule exhibits a bilinear characteristic. E_1 was calculated as the slope of the best linear fit between 0 and 5% of the ultimate strain, and similarly E_2 was calculated between 60 and 80% of the ultimate strain. The parameters used in the reduced polynomial fit are [MPa]: C_{10} = 0.21, C_{20} = 33.7, C_{30} = -125.6, C_{40} = 235.6 and C_{50} = -58.2.

Through the inverse FE-characterization process of the aspiration tests, the mechanical parameters of the parenchyma and of the homogenized liver tissue (capsule and parenchyma considered as a single homogeneous tissue) were extracted. Fig. 3A shows the correspondence between simulated and measured displacement history in the aspiration test on the intact organ and the parenchyma. Fig. 3B shows the comparison of the corresponding uniaxial responses. Due to the influence of the capsule (with 3 orders of magnitude higher stiffness) the evaluation of the liver as homogenized tissue leads to an overestimation of the properties of the parenchyma by a factor of 2 to 3.

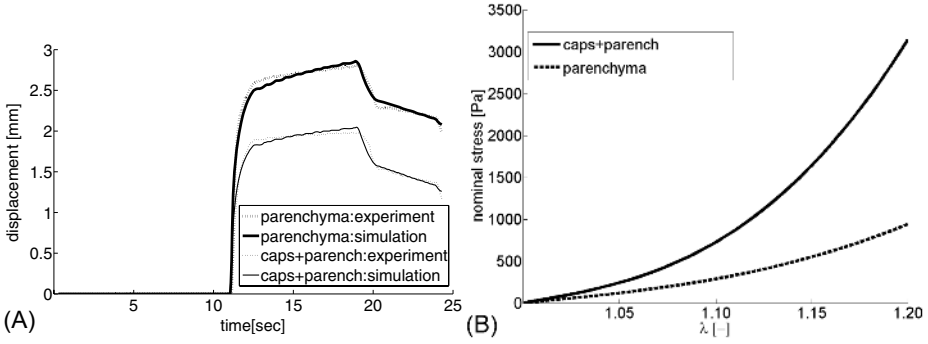


Fig. 3. Aspiration experiment on intact organ and parenchyma: (A) Inverse FE-characterization: simulation and experiments. (B) Corresponding uniaxial stress-strain behavior.

As a validation of the proposed models, the aspiration test on the intact liver has been simulated with the bilayer model. Fig. 4B shows the simulated and measured displacement history. The curve agrees to a good extent with the corresponding experimental data. The observed discrepancy might be related to the neglected time dependence of the mechanical response of the capsule.

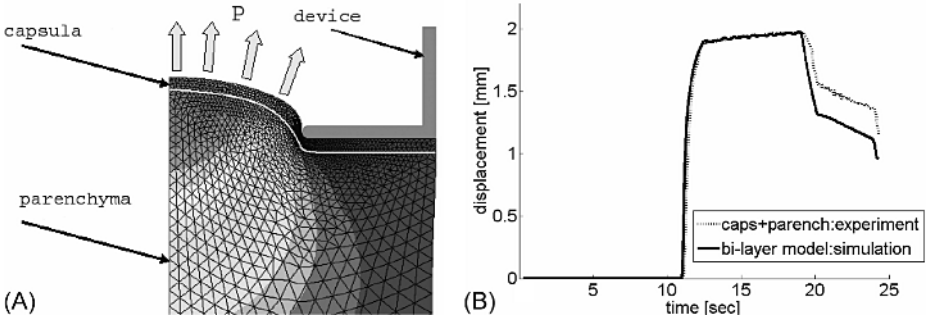


Fig. 4. (A) Bilayer FE-model. In the figure, the thickness of the capsule has been magnified by a factor of 10 to graphically better represent the model. (B) Comparison between the bilayer FE-simulation and the experiments.

4 Conclusions

The bovine liver capsule has been mechanically characterized in vitro in uniaxial tension in preconditioned state until tissue rupture. Findings agree well with published data for the human and the porcine kidney capsule (Snedeker et al. [9], Yamada [14]). The good repeatability of the tests confirms the suitability of the testing protocol and evaluation procedure. Accurate characterization of the "in vivo" loading conditions of the capsule and implementation of these as initial experimental condition in tensile tests will improve the reliability of the corresponding constitutive equations. In

addition, experiments with different loading rates will allow time dependence of the capsule response to be characterized.

Aspiration experiments were carried out on the intact organ and, having removed the capsule, on the parenchyma. The results show considerable differences in the corresponding mechanical response, and neglecting the influence of the capsule leads to a significant overestimation of the parenchymal properties. This demonstrates the importance of correct modeling of the capsule for (i) simulation of organ behavior, and (ii) interpretation of experimental data for diagnostic purposes.

To this end, a bilayer model of the organ has been implemented in the inverse FE-characterization process of the aspiration experiment. Using constitutive equations extracted from the individual measurements of capsule and parenchyma, FE-simulations yield good agreement with respect to the experimental displacement curves.

Acknowledgements

This work has been supported by the Swiss NSF project Computer Aided and Image Guided Medical Interventions (NCCR CO-ME).

References

1. Picinbono, G., Lombardo, J.C., Delingette, H., Ayache N.:Improving realism of a surgery simulator: linear anisotropic elasticity, complex interactions and force extrapolation. J. Visual. Comp. Animat., 13 (2002) 147-167
2. Snedeker, J. G., Bajka, M., Hug, J.M., Szekely, G., Niederer, P.:The creation of a high-fidelity finite element model of the kidney for use in trauma research. J. Visual. Comp. Animat., 13 (2002) 53-64
3. Szekely, G.:Surgical simulators,. Minimal. Invasive Ther. Allied Technol., 12 (2003) 14-18
4. Brown, J. D., Rosen, J., Kim, Y.S., Chang, L., Sinanan, M., Hannaford, B.:In vivo and in situ compressive properties of porcine abdominal soft tissues. Med. Meets Virtual Reality 11, (2003)
5. Kalanovic, D., Ottensmeyer, M.P., Gross, J., Gerhardt B., Dawson, S.I.:Independent testing of soft tissue viscoelasticity using indention and rotary shear deformation. Med. Meets Virtual Reality, 11 (2003)
6. Nava, A., Mazza, E., Kleinermann, F., Avis, N.J., McClure J., Bajka, M.,.Evaluation of the mechanical properties of human liver and kidney through aspiration experiments. Technol. Health Care, 12 (2004) 269-280
7. Nava, A., Mazza, E., Häfner, O., Bajka, M.:Experimental observation and modelling of preconditioning in soft biological tissues. Lecture Notes in Computer Science vol. 3078, 3078 (2004) 1-8
8. Ottensmeyer, M. P.:TeMPeST I-D: An instrument for measuring solid organ soft tissue properties. Experimental Techniques, 26 (2002) 48-50
9. Snedeker, J. G., Niederer, P., Schmidlin, F.R., Farshad, M., Demetropolous, C.K., Lee, J.B., Yang, K.H.:Strain-rate dependent material properties of the porcine and human kidney capsule. Journal of Biomechanics, 38 (5) (2005) 1011-1021
10. Kauer, M., Vuskovic, V., Dual, J., Szekely, G., Bajka, M.:Inverse finite element characterization of soft tissues. Med. Image Anal., 6 (2002) 275-287

11. Mazza, E., Nava, A., Hahnloser, D., Jochum, W., M. Bajka, M. "In vivo mechanical behavior of human liver,." *International Conference on the Ultrasonic Measurement and Imaging of Tissue Elasticity*, Austin, TX.
12. Nava, A., Mazza, E., Furrer, M., Villiger, P., Reinhart, W.H. "Mechanical response of human liver from in vivo experiments." *Proceedings of the First International Conference on Mechanics of Biomaterials & Tissues*, Waikoloa, Hawaii, USA.
13. Carter, F. J., Frank, T.G., Davies, P.J., McLean, D., Cuschieri, A. .Measurement and modelling of the compliance of human and porcine organs. Med. Image Anal., 5 (2001) 231-236
14. Yamada, H.:Strength of Biological Materials. edn. Williams & Wilkins, Huntingdon NY (1970)
15. Herbert, L.A., Chen, W.C., Hartmann, A., Garancis, J.C.:Mechanical properties of the dog renal capsule. Journal of Applied Physiology, 40 (1976) 164–170
16. Farshad, M., Barbezat, M., Flueler, P., Schmidlin, F.R., Graber, P., Niederer, P.:Material characterization of the pig kidney in relation with the biomechanical analysis of renal trauma. Journal of Biomechanics, 32 (1999) 417–425
17. Melvin, J. W., Stalnaker, R.L., Roberts, V.L., Trollope, M.L. "Impact injury mechanisms in abdominal organs." *17th Stapp Car Crash Conference*, 115-126.
18. Arnold, G., Gressner, A.M., Clahsen, H.:Experimental studies of the historheology of the liver capsule. Anatomischer Anzeiger, 142 (3) (1977) 180-191
19. Mazza, E., Nava, A., Bauer, M., Winter, R., Bajka, M., Holzapfel, G.A.:Mechanical properties of the human uterine cervix: An in vivo study. Med. Image Anal., 6 (2006) 125-136
20. Yeoh, O. H.:Some forms of the strain energy function for rubber. Rubber Chemistry and Technology, 66 (1993) 754-771
21. Vuskovic, V.:Device for in vivo measurement of mechanical properties of internal human soft tissues. Diss., ETH Zurich No. 14222.(2001)
22. Fung, Y. C.:Biomechanics: Mechanical Properties of Living Tissues. 2nd edn. Springer, New York (1993)

Parameter Optimisation of a Linear Tetrahedral Mass Tensor Model for a Maxillofacial Soft Tissue Simulator

Wouter Mollemans[1], Filip Schutyser[1], Nasser Nadjmi[2], Frederik Maes[1], and Paul Suetens[1]

[1] Medical Image Computing (Radiology - ESAT/PSI), University Hospital Gasthuisberg, Herestraat 49, B-3000 Leuven, Belgium
[2] Eeuwfeestkliniek, Harmoniestraat 68, B-2018 Antwerpen, Belgium

Abstract. We present an extensive validation of the prediction accuracy of our soft tissue simulator for maxillofacial surgery planning using a linear Tetrahedral Mass Tensor Model (MTM). Prediction accuracy is quantified by measuring distances between the predicted data and the actual post-operative CT data for a database containing 10 patients who underwent maxillofacial surgery. Two different setups are considered. First two important parameters of a homogeneous MTM are optimised, namely the material's Poisson Ratio ν and the number of tetrahedra contained by the mesh. Optimal results were achieved with $\nu \approx 0.46$ and $N_{tetra} \geq 30.000$. Moreover the average simulation time could be reduced to less than 2.5 seconds. In the second setup an inhomogeneous MTM that differentiate between biomechanical properties for fat and muscle tissue is introduced. Simulation results show to be independent of Young's Moduli and optimal results were achieved for $\nu_{fat} = 0.485$ and $\nu_{muscle} = 0.43$. Moreover it turned out that using such an inhomogeneous model doesn't improve simulation accuracy significantly when compared to the homogeneous model.

1 Introduction

Maxillofacial surgery treats abnormalities of the skeleton of the head. Skull remodelling implies osteotomies, bone fragment repositioning, restoration of bone defects and inserting implants. Since the human face plays a key role in interpersonal relationships, people are very sensitive to changes to their outlook. Therefore planning of the operation and reliable prediction of the facial changes are very important.

This simulation of the deformation of the facial soft tissues due to bone movement, demands a mathematical model that is able to imitate the behavior of the facial tissues. We presented in the past [1] the usage of a linear Mass Tensor Model (MTM) as biomechanical model. This model tries to combine the advantages of Mass Spring Models, that have an easy architecture and short simulation times, and Finite Element Models that are considered to be very biomechanically relevant, which results in accurate soft tissue predictions.

M. Harders and G. Székely (Eds.): ISBMS 2006, LNCS 4072, pp. 159–168, 2006.

The original MTM was introduced by Cotin *et al.* [2]. In the MTM the modelled object is discretized into a tetrahedral mesh. Inside every tetrahedron T_i, the displacement field is defined by a linear interpolation of the displacement vectors of the four vertices of T_i, as defined by the finite element theory. It is than shown that the total elastic force at vertex j, after displacement of some of the mesh vertices, is given by:

$$F_j = [K_{jj}]u_j + \sum_{k \in N(j)} [K_{jk}]u_k \tag{1}$$

where u_k is the displacement of vertex k, $N(j)$ is the collection of all vertices neighbouring to vertex j and $[K_{jk}]$ are the global stiffness tensors for vertex j.

These stiffness tensors are directly proportional to the material's Modulus of Young and are dependant on the material's Poisson Ratio ν and the initial mesh configuration [2]. When calculating the soft tissue deformations due to maxillofacial surgery we first displace a subset of soft tissue points over a predefined distance derived from the bone related planning. Next the new position of all the other soft tissue points is found by demanding that the total elastic force in each of these soft tissue points should be zero when the point is in rest. This equals to solving following equation for all these points:

$$F_j(u_j^{new}) = 0 \rightarrow [K_{jj}]u_j^{new} + \sum_{k \in N(j)} [K_{jk}]u_k^{new} = 0 \tag{2}$$

which is solved using an iterative local steepest gradient approach.

In this work we investigate the influence of different parameters of the MTM on the accuracy of the prediction result. Since visualisation of the new facial outlook is the goal of our simulator, we validate prediction accuracy by measuring distances between the predicted and actual post-operative data. We refer to [3] for more detail about this validation work flow.

Two different setups are studied. First the facial soft tissues are considered to be a homogeneous material. For 10 data sets, containing pre-operative and post-operative patient's CT data, we derive the optimal biomechanical constants, i.e. the Poisson Ratio and Young's Modulus, and the optimal mesh size. In the second setup the homogeneous MTM is extended to an inhomogeneous model, in which we differentiate between fat and muscle tissue. We briefly summarize how the model is built and investigate the effect of varying the value of the biomechanical constants for fat and muscle tissue. Optimal parameters are derived. Results are presented and discussed in the third and fourth section.

2 Material and Methods

2.1 The Homogeneous Tetrahedral Mass Tensor Model

In a first setup we approximate the facial tissues as a homogeneous linear elastic material. The same value of the biomechanical constants, i.e. the Poisson Ratio

and Young's Modulus, will be assigned to all the soft tissue points. Since the global stiffness constants are directly proportional to Young's Modulus in the MTM, it can easily be seen that we can substitute equation 2 by:

$$\frac{[K_{jj}]}{E}u_j + \sum_{k \in N(j)} \frac{[K_{jk}]}{E}u_k = 0 \rightarrow [K^*_{jj}]u_j + \sum_{k \in N(j)} [K^*_{jk}]u_k = 0 \qquad (3)$$

with E Young's Modulus and $[K^*_{jk}]$ the new global stiffness tensors, which are independent of the Young's Modulus. Because only calculation of the new rest position of the soft tissue points is relevant for our application,i.e. solving the above equation, just two parameters need to be further defined: the mesh topology (more specific the mesh size was tested) and the Poisson Ratio.

The Poisson Ratio. The Poisson Ratio is defined as the ratio of the contraction strain normal to the applied load to the extension strain in the direction of the applied load. This ratio can vary between 0.0 and 0.5, where a value of 0.5 corresponds to a perfectly incompressible material but is not achievable with a linear elastic model. We therefor we varied in our experiment the ratio between 0 and 0.495. We generated for each data set a tetrahedral mesh including all facial soft tissues and containing on average 80.000 tetrahedra. For each value assigned to the Poisson Ratio, we calculated the new facial outlook and measured distances between corresponding points of the predicted and post-operative data, using a validation framework as discussed in [3]. Next we calculated the 50%, 90% and 95% percentiles of the generated distance maps. These statistics give a good indication of the prediction accuracy.

The Mesh Size. In [3] we suggest to map deformations calculated on a volumetric tetrahedral mesh, including only the facial soft tissues that will deform during simulation, to a dense surface representation of the whole skin for computational efficiency and a nice visualisation result. When using this method we can easily investigate the influence of the mesh size on the final prediction result, since deformations calculated on different meshes are always mapped to the same skin surface. The tetrahedral mesh is built out of the pre-operative CT data. This meshing includes three steps. First the facial soft tissues are semi-automatically segmented using a levelset approach. Next a triangular mesh that envelops these segmented facial soft tissues, is constructed with the Amira software (Amira, TGS, France). Finally, starting from this triangular surface, we assemble a tetrahedral mesh using the Netgen package [4].

To control the number of tetrahedra used, we generate triangular meshes containing $[500, 2500, ..., 50000, 75000]$ triangles. For each triangular mesh a tetrahedral mesh is constructed and the new facial outlook is calculated. During these simulations the Poisson Ratio was set to 0.46. Afterwards the simulated deformations were mapped to the facial skin surface and distances between the deformed skin surface and the co-registered post-operative skin surface were calculated. The 50%, 90% and 95% percentiles of the distance distributions were determined.

2.2 The Inhomogeneous Tetrahedral Mass Tensor Model

To improve simulation results, we investigate the effect of assigning different biomechanical properties to each tissue type. We distinguish between two tissue types: fat tissue and muscle tissue.

Model Building. To distinguish between the different tissue types we look at the intensity values in the pre-operative CT data and use a so-called Gaussian mixture model in combination with the EM-algorithm [5]. This model tries to approximate the intensity histogram as a combination of Gaussian distributions, where each distribution corresponds to a specific tissue type (see figure 1).

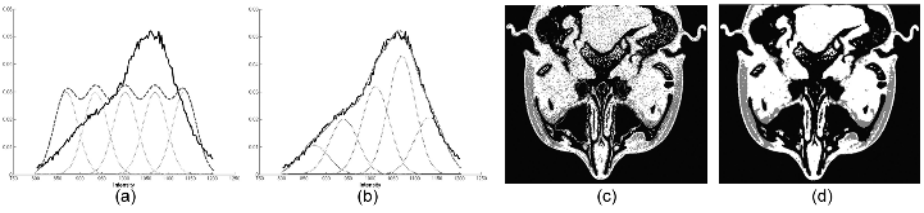


Fig. 1. (a) 5 Gaussian distributions are used to initialise the mixture model. The black line shows the intensity histogram of the CT slice, the initial Gaussians are drawn in grey dots and the dark grey dashed line indicates the initial approximation of the histogram. (b) Gaussian distributions after EM optimisation. (c)The segmented CT slice after application of the calculated tresholds. Grey labels correspond to fat tissue, while white labels correspond to muscle tissue. (d) The segmented CT slice after median filtering.

The Gaussian mixture model allows optimal threshold definition in every CT slice, i.e. the value for which the probability that a certain intensity value is labelled as fat, equals the probability to be labelled as muscle tissue. After applying these optimal tresholds for each CT slice (figure 1 (c)), we use a 5 by 5 median filter to remove noisy classifications. Figure 1 (d) shows the final segmentation map. The orange labels correspond to fat tissue, while white labels are defined as muscle tissue.

According to the Finite Element theory, a tetrahedral mesh should now be built where each of the tetrahedra contains only muscle or fat tissue. This approach can lead to a rather tedious and error-prone meshing step. Therefore we suggest to first build a very dense tetrahedral mesh, containing all facial tissues that will deform during simulation, based on the homogeneous segmented data. Typically such a tetrahedral mesh contains 150.000 tetrahedra (average tetrahedron volume smaller than 2.0 mm^3). Next we define for each tetrahedron the Young's Modulus and Poisson Ratio as the average Young's Modulus and Poisson Ratio over all voxels that lie inside this tetrahedron.

Parameters. Corresponding to the homogeneous model, since we only want to define the displacement of each soft tissue point after applying a fixed displacement to a subset of the points, we can again rewrite formula 2:

$$\frac{[K_{jj}]}{E_{fat}}u_j + \sum_{k \in N(j)} \frac{[K_{jk}]}{E_{fat}}u_k = 0 \rightarrow [K^{'}_{jj}]u_j + \sum_{k \in N(j)} [K^{'}_{jk}]u_k = 0 \qquad (4)$$

where E_{fat} is the Young's Modulus of fat tissue and $[K^{'}_{jk}]$ are the new global stiffness tensors. These tensors are now proportional to the ratio E_{muscle}/E_{fat}, ν_{muscle}, ν_{fat} and the initial mesh topology.

We investigate for this inhomogeneous model the influence of only the first 3 parameters on the final prediction result. ν_{muscle} and ν_{fat} were separately varied between 0.0 and 0.495. For E_{fat} and E_{muscle} many values have been reported in the literature [6,7,8]. Because of to the wide range of these reported values, we decided to vary E_{muscle}/E_{fat} between 1 and 64, imposing that muscle tissue must be stiffer than fat tissue. For all parameter settings distances between the simulated facial outlook and the actual post-operative outcome were measured as discussed in [3]. The 50%, 90% and 95% percentiles of the distance distributions were determined.

3 Results

We acquired a data set of 10 patients who underwent a maxillofacial procedure, including pre-operative and post-operative CT data. The average voxelsize of the CT data measured $0.35 \times 0.35 \times 0.5\ mm$. The patient group counts 2 Class III and 8 Class II patients [9]. For all patients the work flow as presented in [3], was used to generate the tetrahedral mesh and proper boundary conditions, that serve as input to the MTM.

3.1 Poisson Coefficient

Figure 2 summarizes the prediction error behavior in function of the Poisson Ratio for all 10 patients. On the left hand side the 90% percentile prediction errors for all 10 patients are shown. On each graph a cross indicates where the minimal prediction error was reached. We note that for different patients this minimum was achieved at different Poisson Ratio's. Moreover, it was noted for some patients that the recorded optimal Poisson Ratio, i.e. when the inspected error statistic becomes minimal, was dependent of this statistic. To find some sort of global mean over all data sets, we averaged the 50%, 90% and 95% percentile prediction error over all patients. As can be seen in figure 2, the prediction error was minimised for all three statistics when $0.45 \leq \nu \leq 0.46$. This value corresponds nicely to reported values on the Poisson Ratio of soft tissues [10].

We calculated for all 10 patients the difference in mm between the minimal 90% percentile and the 90% percentile prediction error, obtained when the Poisson Ratio was set to 0.46, which was found to be the global minimum. Only

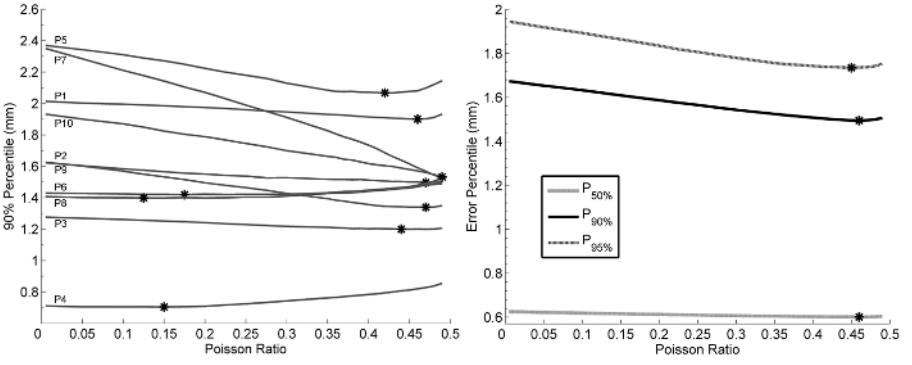


Fig. 2. (a) the 90% percentile prediction error (mm) in function of the Poisson Ratio of the homogeneous Mass Tensor Model for all 10 patients. (b) At the right hand side the average 50%, 90% and 95% percentiles are shown.

for patient 4 the difference measured more than 0.07 mm ($D^{P4}_{90\%} = 0.116\ mm$) Moreover the average difference between both percentiles over all data sets was found to be smaller than 0.05 mm. This difference is clearly negligible to the absolute accuracy of the predictions ($\approx 1.5\ mm$).

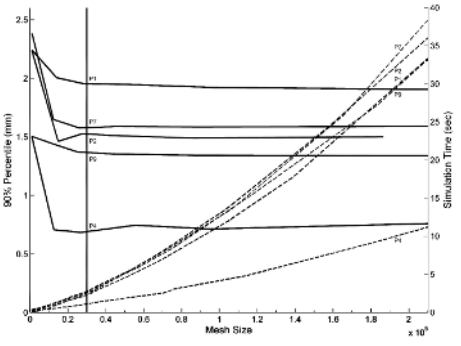
3.2 The Mesh Size

For 5 of the 10 patients we calculated the prediction accuracy in function of the mesh size. In figure 3 we present the results of these simulations. Accuracy becomes more or less constant when a 'sufficient' number of tetrahedra was used. For one data set, patient 4, even a slight decrease in accuracy was observed when the mesh size was increased. Probably this effect is induced by the volume/surface mapping technique [3] which includes some kind of Gaussian smoothing. When a less dense tetrahedral mesh is chosen, smoothing is larger and this may result in a better prediction of the new facial outlook.

When the mesh contains approximately 30.000 tetrahedra, the average difference between the 90% percentile achieved for this mesh and the result obtained whit a very dense mesh ($N_{tetra} > 200.000$), becomes smaller than $0.05mm$. This difference is negligible compared to the absolute accuracy of the predictions. Moreover figure 3 shows that simulation time varies more or less linear to the mesh size. As a consequence simulation time could be reduced to an average value of 2.2 *sec*. These very fast simulation times are a great benefit, when using the soft tissue simulator in daily clinical practice.

3.3 Inhomogeneous Model Parameters

We calculated the new facial outlook of all 10 patients, varying E_{muscle}/E_{fat} between 1 and 64 and ν_{muscle} and ν_{fat} between 0 and 0.495, as discussed in



Patient	N_{tetra}	T_{sim}	$D_{90\%}$
P1	31943	2.8	0.013
P2	27390	2.1	0.033
P4	31238	1.1	-0.068
P7	30094	2.7	-0.059
P9	28681	2.1	0.031
Average	29869	2.2	0.048

Fig. 3. On the left hand side the 90% percentiles (full lines) and simulation times (dotted lines) in function of the mesh size are shown. The black vertical line indicates when the mesh contains more than 30.000 tetrahedra. In the table on the right hand side, the third column shows the simulation time needed in seconds for a mesh containing approximately 30.000 tetrahedra. The differences in mm between the 90% percentile when using more than 200.000 and the 90% percentile obtained when using only 30.000 tetrahedra, are listed in the last column.

section 2.2. Figure 4 (a) lists simulation results for one typical data set, patient 9. For different ratio's of the Young's Moduli (E_{muscle}/E_{fat}) the minimal and maximal 50% and 90% percentiles over all possible values for ν_{fat} and ν_{muscle}, that were achieved, are calculated. As shown, the effect of the Young's Moduli is negligible compared to the absolute accuracy of the predictions ($\approx 1.5\ mm$), while the dependance on the Poisson Ratio's is in the order of $0.2\ mm$. This conclusion was enforced after processing the other data sets. For patient 9 the optimal prediction accuracy ($P_{90\%} = 1.382\ mm$) was achieved with $\nu_{fat} = 0.4$ and $\nu_{muscle} = 0.485$.

	Min		Max	
E_m/E_f	$P_{50\%}$	$P_{90\%}$	$P_{50\%}$	$P_{90\%}$
1	0.378	1.382	0.448	1.590
2	0.377	1.384	0.449	1.591
8	0.380	1.385	0.449	1.588
16	0.384	1.386	0.450	1.588
32	0.386	1.387	0.450	1.588
64	0.389	1.388	0.451	1.589

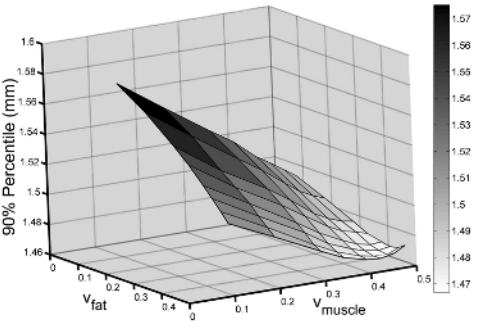


Fig. 4. The left table presents for data set 9, the minimum and maximum 50% and 90% percentiles over all possible values for ν_{fat} and ν_{muscle} for different ratio's of the Young's Moduli E_{muscle} and E_{fat}. At the right hand side, the average 90% percentile over all data sets is shown for $E_{muscle}/E_{fat} = 1$.

Similar to the homogeneous model, not all data sets reached a minimum for the same set of parameters. To find a global minimum, the 90% percentiles were averaged over all data sets and the minimum was defined. As expected these averaged values are independent of the ratio of Young's Moduli. The resulting graph for $E_{fat}/E_{muscle} = 1$ (figure 4) shows the prediction accuracy in function of the Poisson Ratio for fat and muscle. An optimal accuracy was reached for $\nu_{fat} = 0.485$ and $\nu_{muscle} = 0.43$.

4 Discussion and Conclusion

In this work we tried to optimise the biomechanical constants, i.e. the Young's Modulus and the Poisson Ratio, and the tetrahedral mesh size for a linear Mass Tensor Model (MTM) used to simulate the new facial outlook after maxillofacial surgery. Since the MTM is a variant of the Finite Element Model (FEM), results are applicable to all biomechanical FEM-based models. Two different setups were considered: a homogeneous and inhomogeneous model.

For the homogeneous setup, we showed that optimal simulation results are achieved with $0.45 \leq \nu \leq 0.46$. Moreover mesh size could be reduced to 30.000 tetrahedra, which lowered the average simulation time to 2.2 seconds, but did not affect the prediction accuracy. Since in our application no external force are calculated the influence of Young's Modulus is null. For the inhomogeneous model we showed that simulation results were almost independent of the ratio of Young's Moduli and highest accuracy was achieved with $\nu_{fat} \approx 0.485$ and $\nu_{muscle} \approx 0.43$. When comparing the homogeneous and inhomogeneous model, the simulation results were not found to be significantly better. Consequently we conclude that there is no net improvement when using an inhomogeneous tissue model.

Recently Zachow *et al.* [11], reported on a quantitative evaluation of 3D soft tissue predictions for maxillofacial surgery on a single data set. In this work they also investigated the influence of the Poisson Ratio and the usage of a inhomogeneous Finite Element Model, on the final prediction result of this one data set. They concluded that an optimal accuracy was achieved with $0.43 \leq \nu \leq 0.45$, but the observed variations in prediction accuracy were rather small ($\approx 0.02mm$). We however conclude, based on a more extensive validation as it includes 10 patients, that for some patients variations are indeed rather small (see figure 2), but for other data sets (patient 5,7,9 and 10) an appropriate choice of the Poisson Ratio ν clearly influences the final prediction result. For this last group, accuracy clearly improved when the Poisson Ratio was increased. Consequently the difference between the best prediction, i.e. when the 90% percentile was minimal, and the facial prediction when ν was set to 0.46, was smaller than 0.1 mm for al data sets. This difference is still negligible compared to the actual prediction accuracy ($P_{90\%} \approx 1.5\ mm$).

For some patients maximal accuracy is achieved in different facial regions for different parameter settings. This may result in a rather flat slope of the global prediction accuracy in function of the Poisson Ratio. To verify this hypothesis, we

defined for each facial tissue point the optimal Poisson Ratio, i.e. the value that minimises the 90% error percentile, and visualised these optimal values as a color code onto the facial skin surface. Figure 5 (a) shows the result for patient 2. When inspecting this image, one should also keep in mind the variation in accuracy that was obtained by setting ν to this optimal value. Therefore figure 5 (b) shows the measured standard deviation of the 90% percentile over all Poisson Ratio's, by means of a color code. The color code ranges from 0 mm to 0.3 mm. As can be seen for the cheek and the Labiomental Fold region, optimal accuracy is obtained when the Poisson Ratio is set to a high value ($\nu \approx 0.47$), while for the lip, tip of the nose and the Gonion region the most accurate result is obtained with $\nu \approx 0.1$. This difference probably causes the quite flat behavior of the global 90% error percentile in function of the Poisson Ratio. Similar conclusions could be made for the other data sets.

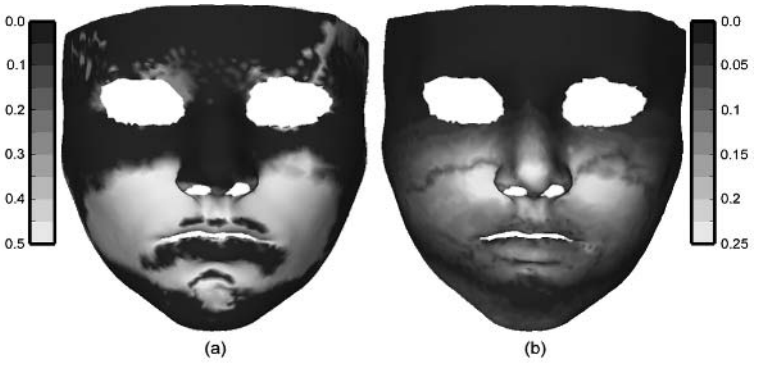


Fig. 5. For each facial tissue point the optimal Poisson Ratio (a) and standard deviation of the 90% error percentile in function of the Poisson Ratio (b), were calculated and visualised by means of a color code

Future research is required to define these different facial regions, based on the pre-operative CT data and planning data. These differences in optimal Poisson Ratio, may also arise from difference in lip posture between the pre-operative an post-operative acquisition or incorrectly defined boundary conditions like already suggested in [3]. In the near future, we hope to be able to define some correlation between the planned procedure and the biomechanical behavior of certain facial regions by extending and statistical analysing our validation database.

Acknowledgements

This work is part of the Flemish government IWT GBOU 020195 project on Realistic image-based facial modelling for forensic reconstruction and surgery simulation and K.U.Leuven/OF/GOA/2004/05.

References

1. Mollemans, W., Schutyser, F., Nadjmi, N., Suetens, P.: Very fast soft tissue predictions with mass tensor model for maxillofacial surgery planning systems. In: Proc. of CARS. (2005) 491–496
2. Cotin, S., Delignette, H., Ayache, N.: A hybrid elastic model allowing real-time cutting, deformations and force-feedback for surgery training and simulation. The Visual Computer **16**(8) (2000) 437–452
3. Mollemans, W., Schutyser, F., Nadjmi, N., Maes, F., Suetens, P.: 3D soft tissue predictions with a tetrahedral mass tensor model for a maxillofacial planning system: a quantitative validation study. In: Proc. of SPIE. (2006)
4. Schöberl, J.: An advancing front 2D/3D-mesh generator based on abstract rules. Computer Visual Science **1**(1) (1997) 41–52
5. Van Leemput, K., Maes, F., Vandermeulen, D., Suetens, P.: A unifying framework for partial volume segmentation of brain MR images. IEEE Transactions on medical imaging **22**(1) (2003) 105–119
6. Lorenzen, J., Sinkus, R.: MR elastography of the breast: Preliminary clinical results. Rofo-Fortschr Rontg **174**(7) (2002)
7. Levinson, S.F., Shinagawa, M., Sato, T.: Sonoelastic dermination of human skeletal muscle elasticity. Journal for Biomechanics **28**(10) (1995) 1145–1154
8. Duck, F.A.: Physical Properties of Tissue. Academic Press (1991)
9. Sarver, D.: Esthetic Orthodontics and Orthognathic Surgery. 1 edn. Mosby (1997)
10. Fung: Biomechanics: Mechanical Properties of Living Tissues. Springer (1993)
11. Zachow, S., Hierl, T., Erdmann, B.: A quantitative evaluation of 3D soft tissue prediction in maxillofacial surgery planning. In: Proc. of CURAC. (2004) 75–79

An Integrated Dynamic Jaw and Laryngeal Model Constructed from CT Data

Ian Stavness[1], Alan G. Hannam[2], John E. Lloyd[1], and Sidney Fels[1]

[1] Dept. of Electrical and Computer Engineering,
[2] Faculty of Dentistry,
University of British Columbia, Canada

Abstract. Computational modeling is an important tool for studying the structure and function of human anatomy in biomedicine. In this paper, a dynamic, anatomically accurate model of the human mandibular and laryngeal structures is presented along with a set of forward dynamic simulations that show consistency with previously published jaw modeling literature. Laryngeal motion during swallowing was also simulated and shows plausible upward displacement consistent with published recordings. A novel open source modeling platform, ArtiSynth, is described in the context of its use in the construction and simulation of the biomechanical jaw and larynx model.

1 Introduction

We have developed a high fidelity, dynamic, biomechanical model of the human jaw and laryngeal structures utilizing high resolution CT data for creating the model geometry and fast computational techniques for realtime simulation.

The jaw subsystem of our model is based upon previously published literature [1], in which a commercial mechanical simulation package, ADAMS, was used to construct a model of the human jaw. We have extended this work by incorporating the hyoid bone, cricothyroid complex, and associated muscles, which enables analysis of interactions between the jaw and laryngeal systems and other physiological tasks, such as swallowing [2] and vocal chord stretching [3]. Our model has been created using ArtiSynth [4], an open-source biomechanical modeling system which we have developed specifically for creating models of the human head and upper airway. ArtiSynth enables a modular approach to model construction, allows flexible incorporation of medical image data to inform model geometry, and provides significant improvements to usability and efficiency in creating forward dynamic simulations of the complex human mandibular system. It provides an extensible framework for connecting anatomical models, and we are integrating the jaw model with a 3D deformable tongue model in an effort to realize a complete, dynamic model of the human upper airway.

Computational modeling is an important tool for medical and scientific analysis of the human body. In particular, dynamic modeling allows for the study of the relationship between structure and function in physiological actions. The

M. Harders and G. Székely (Eds.): ISBMS 2006, LNCS 4072, pp. 169–177, 2006.

applications of craniomandibular modeling in biomedicine are significant and include the analysis of morphological and functional pathology, surgical planning and prediction, and the role of jaw posture during obstructive sleep apnea.

2 The Model

Our three-dimensional, dynamic jaw model (Fig. 1) is rooted in published literature on biomechanics models of the human jaw. As such, it shares a number of commonalities with previously published models; however, we have made innovations in both modeling and simulation techniques that further the state of the art.

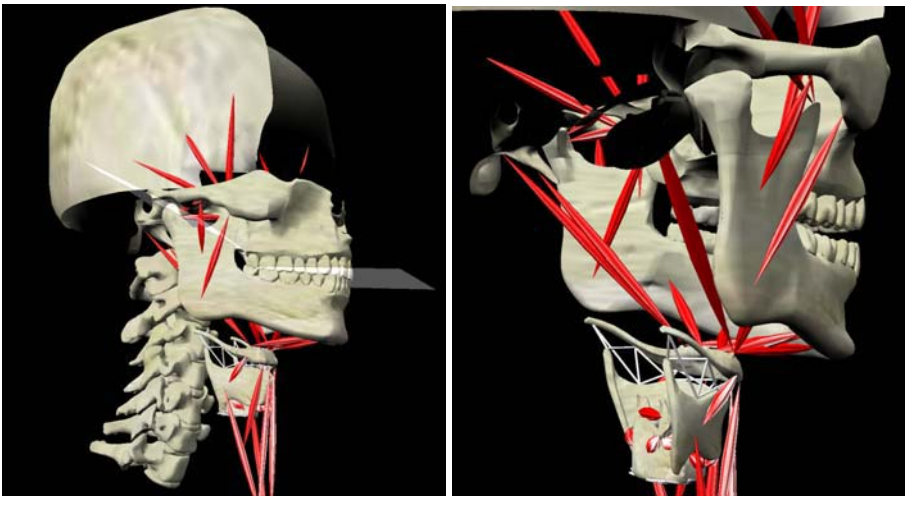


Fig. 1. The jaw model showing maxilla, mandible, laryngeal structures, and vertebrae, along with straight-line muscles and connective tissue

2.1 Related Work

Mathematical models have been used extensively to study the biomechanics of the human musculoskeletal system (see [5] for review).

Computational models of the human jaw have been used primarily to study jaw function in speech and mastication. Various aspects of human mastication have been analyzed with biomechanical models including joint dynamics [6], bite forces [7], muscle dynamics [8], and passive muscle forces [9] [10]. Jaw motion in speech utterances have also been extensively studied with models [11].

Physics-based models represent the mandible bone as a rigid body, the temporomandibular joint as a mechanical joint or contact surface, and the jaw muscles as tension-producing straight-line actuators. The Peck jaw model [1] was developed with a commercial computer aided engineering (CAE) package ADAMS,

and the Koolstra jaw models have been developed both with FORTRAN libraries [8] [12] and a commercial CAE package Madymo [6].

2.2 Model Construction

Our jaw and laryngeal model was constructed within ArtiSynth, a platform for three-dimensional biomechanical physics-based modeling and simulation targeted specifically at modeling orofacial and upper airway anatomy.

ArtiSynth provides a Java API that includes a set of base components (rigid bodies, particles, muscles, and finite element meshes) that can be extended and assembled to create complex biomechanical models. The jaw model was constructed with these model components given the appropriate geometric and dynamic properties described below.

Model Structure. The jaw substructure consists of a fixed rigid skull and a floating rigid mandible that are connected by two temporomandibular joints. These joints are modeled presently as rigid bilateral planar constraints, angled downward and forward, restricting the translation of the mandibular to the planar surface, which approximates the curvilinear condylar path.

Bite contact is achieved by an additional rigid unilateral planar constraint located at the upper mid-incisor point and angled to the occlusal plane that generates reaction forces at bite points normal to this dental plane. Our rigid body constraint formulation does allow for general mesh-on-mesh multi-point contact between the maxilla and mandible teeth. This would be potentially useful for detailed studies of three-dimensional bite forces using high resolution surface scans of dental casts. It also is computationally feasible, since it has recently been shown that contact simulation has an expected complexity of $O(n)$, in the number of contacts, for systems with a fixed number of degrees of freedom [13].

The larynx is a deformable organ that can be approximated as a series of rigid structures connected by deformable tissue. Our laryngeal substructure includes rigid bodies for the hyoid bone and laryngeal cartilages, as well as a fixed sternum to anchor hyoid and thyroid depressor muscles and a fixed vertebrae column as a landmark for anatomy registration. Stiff spring meshes are used to connect the hyoid and thyroid, representing the thyrohyoid membrane, as well as to anchor the cricothyroid complex, representing longitudinal compliance of the trachea (see Fig. 5). Although the laryngeal connective tissue is currently approximated by spring networks, ArtiSynth allows extension to more sophisticated finite-element tissue models.

The jaw and laryngeal model is actuated by a set of 45 straight-line Hill-type [14] muscles. Muscle properties for jaw muscles, including maximum force magnitude, fibre-to-tendon length ratios, and passive tension characteristics, are based on published values (see Table 1 in [1]). Properties of the laryngeal muscles are not well described in literature; therefore, we have chosen parameters that are arbitrarily scaled from mandibular muscles by approximate differences in muscle length and cross-sectional area. Muscle attachment sites have been taken from medical image data (see below). ArtiSynth allows for modular exchange of model

components, therefore simpler (linear spring) or more complex (multipennate muscle model) muscle components can be easily interchanged with the standard Hill-type model.

We have used a mass of 200 g for the mandible as reported in [9], and have chosen masses 10, 24, and 23 g for the hyoid, thyroid, and cricoid respectively. Center of mass and inertia properties have been derived from the body mass and mesh geometry.

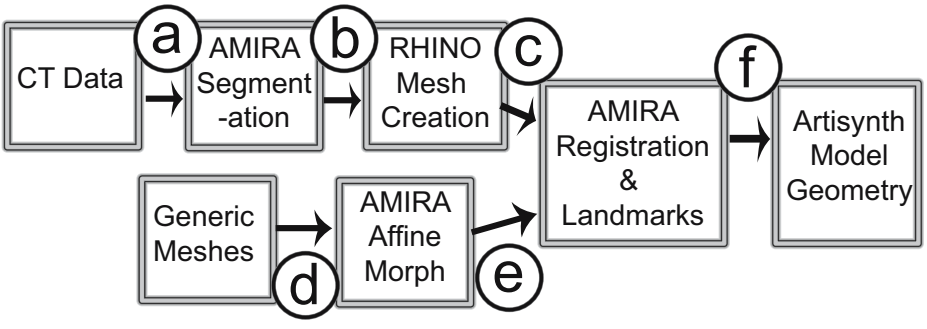


Fig. 2. Process diagram showing the creation of ArtiSynth Geometry from CT data: (a) raw voxel data, (b) segmented voxel mesh, (c) clean NURBS mesh, (d) generic mesh, (e) mesh shaped to specific anatomy, (f) registered meshes and muscle attachment location geometry

Geometry Extraction From CT Data. Several previous models have used published average values to create their model geometry. We used high-resolution (0.4 mm cubic voxel), cone-beam CT scans of a 35 year-old male with normal craniofacial and dental anatomy as an anatomical template.

The process of creating ArtiSynth geometry from the CT data is diagrammed in Fig. 2 and involved the use of two additional software applications: Amira [15], which was used to segment the raw CT data, and Rhino [16], which was used to fit NURBS surfaces to the segmented data. The created structures include a mandible, maxilla, tooth crowns, palate, pterygoid fossae, zygomatic arches, articular eminentia and fossae, infratemporal fossa, postglenoid region, and hyoid bone. Generic meshes for tooth crowns, temporal bones, thyroid cartilage, cricoid bone, arytenoids and cervical vertebrae were morphed and registered using Amira to complete the full set of craniomandibular and laryngeal structures.

We used conventional descriptors to set bilateral landmarks for each muscle attachment [17] [18]. These muscles included the anterior, middle and posterior temporalis, deep and superficial masseter, superior and inferior lateral pterygoid, medial pterygoid, anterior and posterior digastric, posterior mylohyoid, stylohyoid, geniohyoid, thyrohyoid, and cricothyroid muscles. Arbitrary locations were chosen for the inferior sternal attachments of the sternohyoid and sternothyroid, and the intermediate tendon of the omohyoid.

The file interface support built into ArtiSynth allows us to directly import the anatomical meshes and muscle attachment data into our model.

3 Model Validation: Forward Simulation

The complexities of the human mastication system, including both kinematic and actuator redundancy [19], are such that forward-dynamics simulation is the most common type of analysis used to study jaw motion with computer models. The trial and error process of tuning 18 muscle drive signals to achieve desired output motion was found to be tedious by previous authors because the input editing, simulation control, and output visualization tools were in separate programs [9]. In an attempt to create a more effective system for synthesizing jaw motion, ArtiSynth integrates both knobs and graphs into the simulation environment so that an experimenter can quickly tune inputs to achieve desired outcomes using a highly interactive direct manipulation interface. The input muscle parameters and simulation timing are controlled by the *ArtiSynth Simulation Timeline* shown in Fig. 3.

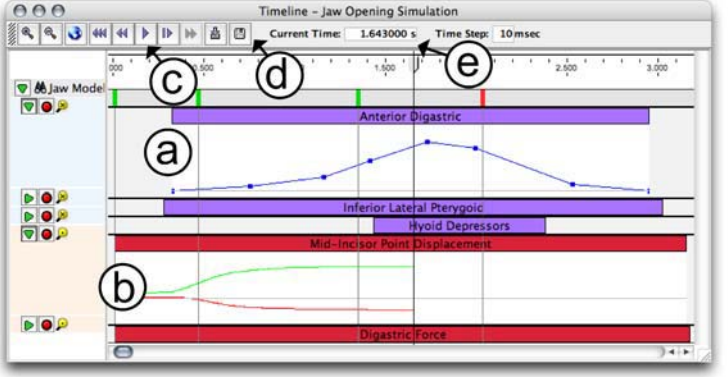


Fig. 3. The ArtiSynth Timeline controlling a jaw motion simulation: (a) input probe (i.e. muscle activation), (b) output probe (i.e. incisor position), (c) play controls, (d) save / load buttons, (e) current time cursor. Input probe data can be modified globally by stretching and translating probes along the timeline or locally by directly dragging data knot points within the probe display.

ArtiSynth Simulation Timeline. The ArtiSynth Timeline integrates input data manipulation, simulation control, and output data extraction in an intuitive interface that is designed on the metaphor of a timeline used in video editing (Fig. 3). Input data form *input probes* that drive the simulation while output data is viewed by *output probes*.

For jaw motion simulation, input probes provided piece-wise linear muscle drive trajectories to individual muscles and functional muscle groups. Output

probes recorded position, velocity, and force data from the model. Simulations were created by scaling and translating input probes along the timeline to coordinate the timing of muscle excitation consistent with putative muscle drive. Appropriate output probes are activated to record output data for the duration of the simulation.

3.1 Results

To validate our model, we activated it with muscle excitations to create a series of motions and verified that the resulting jaw trajectories and postures corresponded to those reported in the literature for similar sets of muscle excitation. The motions we chose to simulate include jaw rest posture, close, midline protrusion, and symmetrical opening. Predicted mid-incisor point position in the mid-saggital plane for the jaw motion trials is shown in Fig. 4.

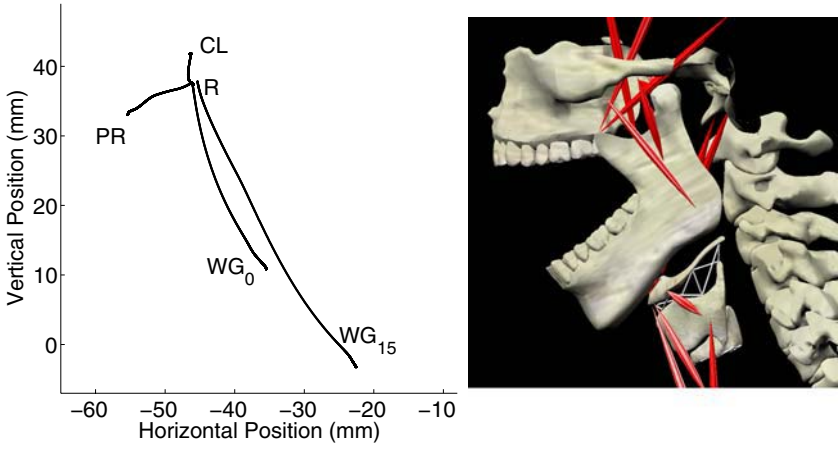


Fig. 4. (a) Plot of Mid-incisor point in mid-saggital plane for four jaw postural tasks: rest position (R), close (CL), maximum protrusion (PR), opening to 35 mm wide gape with no head rotation (WG_0), opening to maximum gape (51 mm) with 15° backward head rotation (WG_{15}). (b) Graphic image of jaw at maximum gape, showing alignment of hyoid bone with lower edge of mandible during head rotation.

Postural Rest Position. In order to achieve a plausible interincisal separation of 3-5 mm at rest under gravity, our model required a steady-state activity of 0.04% maximum activation in the closer muscles (temporalis, masseter, and medial pterygoid). This agrees closely with similar findings in [9]. Increased activation of the closer muscles moved the mandible upward until it stopped rigidly at dental intercuspal contact.

Wide Jaw Opening. Our model was driven to maximum jaw opening by full activation of the jaw opening muscles (anterior digastric, mylohyoid, and inferior lateral pterygoid) in two separate simulations. For comparison to [1] we used reduced passive muscle tension characteristics for the closers and achieved a maximum gape of 38 mm. For comparison to [20] we applied a 15° rotation to the head, and the model achieved a full wide gape opening of 50 mm. At wide gape, with a backward rotated head, the anterior tip of the hyoid body was aligned with the lower edge of the mandible, which is consistent with cephalometric measurements published in [21]. In both wide opening actions, the hyoid depressor muscles were activated to stabilize and slightly lower (1 mm) the laryngeal complex.

Protrusion. We simulated forward midline protrusion of the mandible by activating the inferior lateral pterygoid alone. The model achieved a self-limited forward protrusion consistent with normal behaviour.

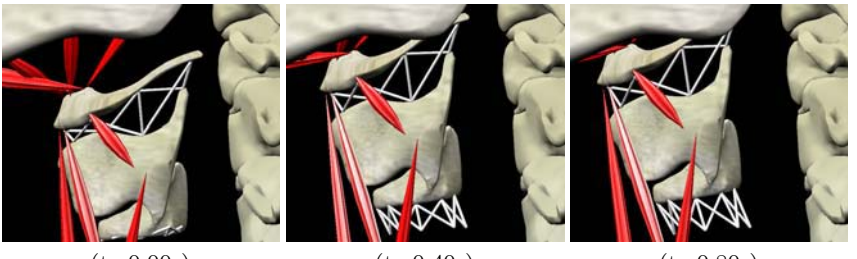


Fig. 5. Time-lapsed graphics of laryngeal complex elevation during simulated swallow

Swallowing. We initiated swallowing by activating the temporalis, masseter, and medial pterygoid muscles to elevate and stabilize the mandible. Upward motion of the larynx occurred following excitation of the digastric, mylohyoid, geniohyoid, and stylohyoid muscles. Predicted magnitude and duration of upward and forward translation of the laryngeal complex (see Fig. 5) is consistent with previous literature on laryngeal elevation [2].

4 Future Directions

Inverse Dynamics Muscle Prediction. As stated in [19], the prediction of muscle recruitment patterns for the human mastication system is the dominant challenge in jaw movement analysis. We are working on an inverse-dynamics scheme to automatically drive the jaw model to follow a desired motion trajectory for speech and mastication tasks. Parameters of muscle force magnitude, jaw stiffness, and task speed are being studied to solve the actuator redundancy problem.

Subject Specific Geometry Adaptation. We plan to build upon our existing technique for geometry extraction from medical image data to incorporate semi-automatic adaptation algorithms in order to more easily generate subject-specific geometry. Additionally, arbitrary geometry morphing will be useful for modeling specific morphological pathology.

Connection with Deformable 3D Tongue Model. We are also developing a muscle driven finite-element tongue model [22] and we are working toward dynamically interconnecting it with the rigid-body jaw-larynx model. Our preliminary model integration has a rudimentary implementation of this connection. We suggest this is an extremely rich direction for biomechanics modeling as the tongue body and extrinsic muscles have a major effect on laryngeal positioning, and the jaw-tongue-larynx system captures the predominant contributors to the fundamental human functions of feeding, breathing, and speaking.

5 Conclusion

We have presented our jaw and laryngeal modeling efforts within the context of our innovative biomechanical simulation software platform, ArtiSynth. We have presented preliminary results of jaw motion simulations that agree with the validation studies performed in previously published literature. Our model extends previous work in physics-based, muscle-driven jaw modeling through advancements in computational techniques and an increased anatomical complexity due to the addition of dynamic laryngeal structures. The medical applications of this computational model are numerous and we are currently working on integrating this model with soft tissue models of the tongue and larynx.

References

1. Peck, C.C., Langenbach, G.E.J., Hannam, A.G.: Dynamic simulation of muscle and articular properties during human wide jaw opening. Archives of Oral Biology **45**(11) (2000) 963–982
2. Fink, B.R.: The Human Larynx: A Functional Study. Raven Press, NY (1975)
3. Takano, S., Honda, K., Kinoshita, K.: Observation of cricothyroid joint motion using 3d high-resolution mri. In: Proc. Voice Phys. and Biomech. (2004) Online
4. Fels, S., Vogt, F., van den Doel, K., Lloyd, J., Stavness, I., Vatikiotis-Bateson, E.: Artisynth: A biomechanical simulation platform for the vocal tract and upper airway. Technical Report TR-2006-10, Computer Science Dept., University of British Columbia (2006)
5. Pandy, M.G.: Computer modeling and simulation of human movement. Annual Review of Biomedical Engineering **3**(1) (2001) 245–273
6. Koolstra, J.H., van Eijden, T.M.G.J.: Combined finite-element and rigid-body analysis of human jaw joint dynamics. J. Biomech. **38**(12) (2005) 2431–2439
7. Osborn, J.W.: Features of human jaw design which maximize the bite force. J. Biomech. **29**(5) (1996) 589–595
8. Koolstra, J.H., van Eijden, T.M.G.J.: Dynamics of the human masticatory muscles during a jaw open-close movement. J. Biomech. **30**(9) (1997) 883 –889

9. Langenbach, G.E.J., Hannam, A.G.: The role of passive muscle tensions in a three-dimensional dynamic model of the human jaw. Archives of Oral Biology **44**(7) (1999) 557–573
10. Peck, C.C., Sooch, A.S., Hannam, A.G.: Forces resisting jaw displacement in relaxed humans: a predominantly viscous phenomenon. Journal of Oral Rehabilitation **29**(2) (2002) 151–160
11. Shiller, D., Ostry, D., Gribble, P., Laboissiere, R.: Compensation for the effects of head acceleration on jaw movement in speech. J. Neuroscience **21** (2001) 6447–6456
12. Koolstra, J.H., van Eijden, T.M.G.J.: The jaw open-close movements predicted by biomechanical modelling. J. Biomech. **30**(9) (1997b) 943–950
13. Lloyd, J.E.: Fast implementation of Lemke's algorithm for rigid body contact simulation. In: Proc. IEEE Conf. on Robotics and Automation. (2005) 4549–4554
14. Zajac, F.E.: Muscle and tendon: properties, models, scaling, and application to biomechanics and motor control. Crit. Rev. Biomed. Eng. **17** (1989) 359–411
15. Amira: Mercury Computer Systems Inc. (2005)
16. Rhino: Robert Mcneel and Associates (2005)
17. Gray, H.: Anatomy of the Human Body. 20 edn. Lea Febiger, Philadelphia (1918)
18. Sicher, H.: Oral Anatomy Fourth Ed. Mosby, Saint Louis (1965)
19. Koolstra, J.H.: Dynamics of the human masticatory system. Crit. Rev. Oral. Biol. Med. **13**(4) (2002) 366–376
20. Koolstra, J.H., van Eijden, T.M.G.J.: Functional significance of the coupling between head and jaw movements. J. Biomech. **37**(9) (2004) 1387–1392
21. Muto, T., Kanazawa, M.: Positional change of the hyoid bone at maximal mouth opening. Oral Surgery, Oral Medicine and Oral Pathology **77**(5) (1994) 451 –455
22. Vogt, F., Lloyd, J.E., Buchaillard, S., Perrier, P., Chabanas, M., Payan, Y., Fels, S.S.: Efficient 3d finite element modeling of a muscle-activated tongue. 3rd International Symposium on Biomedical Simulation (Springer LNCS) (2006)

Simulation of Stomach Specimens Generation Based on Deformation of Preoperative CT Images

TrungDung Truong, Takayuki Kitasaka, Kensaku Mori, and Yasuhito Suenaga

Graduate School of Information Science, Nagoya University,
Furo-cho, Chikusa-ku, Nagoya 464-8603, Japan
ttdung@suenaga.m.is.nagoya-u.ac.jp,
{kitasaka, kensaku, suenaga}@is.nagoya-u.ac.jp

Abstract. This paper presents a novel method for generating virtually-unfolded views of the stomach by cutting and deforming CT images. Unfolded views are very useful for the diagnosis and treatment planning of stomach cancer, since they provide various information of the lumen which can only be obtained from a resected specimen. However, conventional methods cannot correctly reproduce luminal surfaces because elasticity for the shape model of the stomach is quite coarse defined. In this paper, we use Voigt elements for elasticity modeling, forces calculated from surface normals for directing the stomach to a flat shape, and the Newmark-β method for image deformation. We simulated deformation of a phantom dataset and compared the stability as well as computation time with the Euler method. Unfolded views from fifteen CT image datasets, corresponding virtual gastroscopic images, and resected specimens were used for comparison with conventional methods. Experimental results showed that our method can generate views faster in which concave regions of the stomach are better flattened and 99% of the luminal surface can be reproduced. Unfolded views from twelve datasets were presented for surgical planning. They were considered to have well reproduced lesions as well as fold patterns observed in virtual gastroscopic images and resected specimens.

1 Introduction

Virtual gastroscopy is now becoming an extremely gentle method for the early detection of stomach cancer as well as the follow-up examinations without putting too much strain on the patient [1]. However, because the stomach usually has a large cavity, virtual gastroscopy requires frequent change of viewpoints and view directions to confirm whether all regions of interest have been observed. It would be easier for observation if we could draw the entire lumen onto a planar surface. Mori et al. have shown that unfolded stomach views can be generated by cutting and deforming CT image data [2]. A number of approaches for unfolding the colon [3] or blood vessels [4] have also been developed.

Image deformation based unfolding requires both deforming the stomach to a flat shape and correct reconstruction of image data based on shape. Deformation

M. Harders and G. Székely (Eds.): ISBMS 2006, LNCS 4072, pp. 178–187, 2006.

of biological tissues can be accurately simulated by the finite element method (FEM)[5]. However, since the FEM requires remeshing for topological changes and is computationally expensive, it is not suitable for unfolding which definitely accompanies cutting and large deformation. Mori et al. [2] used a node-spring model to represent a stomach model whose deformation is computed as the movement of individual nodes under external forces. However, because elasticity for the model is quite coarse defined, their method cannot reproduce the luminal surface correctly. In this paper, we present a novel method that uses mass points and Voigt elements for elastic modeling, surface normals driven forces for directing the stomach to a flat shape, and the Newmark-β method [7] for image deformation. We believe our method could generate views in which the stomach wall is unfolded better and its luminal surface is reproduced more accurately. We will also show that our method can simulate stable deformation of soft tissues without modifying (or adding constraints to) mass point motion such as most conventional approaches using the Euler method usually do.

In Section 2, detailed procedures of our method are presented. Experimental results of simulating deformation of a phantom dataset and unfolding the stomach from preoperative CT images are shown in Section 3, where discussion is also stated.

2 Method

Unfolded views can be generated by deforming the stomach wall region in a CT image dataset and then visualizing the deformed dataset by volume rendering. Our method consists of six steps : (1) segmentation of the stomach wall, (2) generating a stomach model, (3) cutting and stretching the model, (4) flattening the model to a flat shape, (5) reconstructing image data, and (6) visualizing the reconstructed data. Steps (1), (5) and (6) are performed by exactly applying the conventional method [2], thus we do not describe them in this paper.

2.1 Our Stomach Model

Our stomach model consists of a set of hexahedra wrapping the stomach wall region in a CT image dataset. Each hexahedron is located so that its center coincides with the location of a voxel. The lengths of hexahedron edges in the x, y, and z directions are d, d, and d' ($d' = d \times \mathit{slice\ interval}/\mathit{pixel\ pitch}$), respectively, where d is an interger. Mass points are allocated at vertices of the hexahedra. Voigt elements are allocated along each edge and across the diagonals of each face (Fig. 1). Two hexahedra allocated at two adjacent voxels share one face including four mass points and four Voigt elements.

The equation of motion for a mass point i at time t can be represented by

$$m_i \mathbf{a}_i(t) = \mathbf{Fe}_i(t) + \sum_{j \in S_i} \mathbf{Fs}_{ij}(t), \tag{1}$$

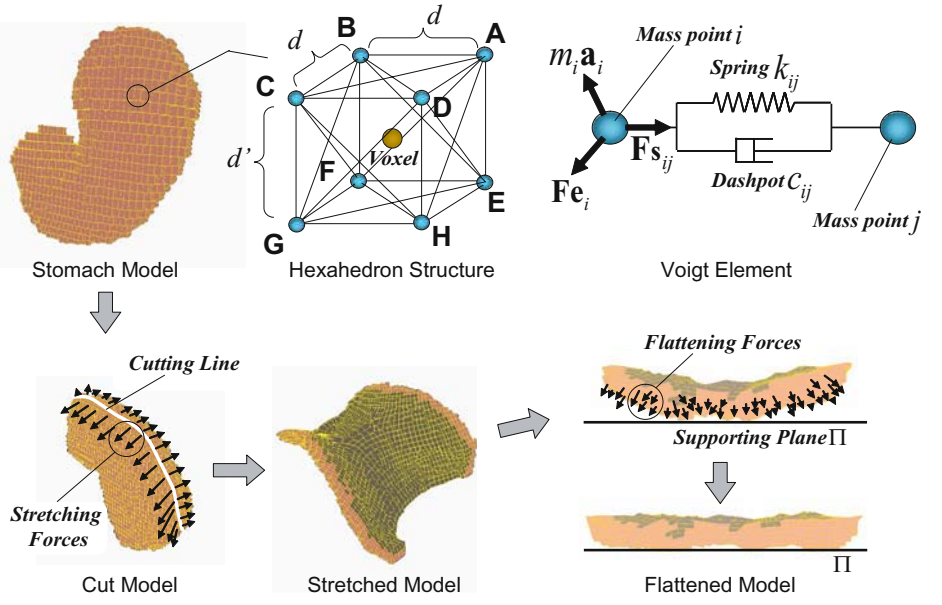


Fig. 1. Cutting, stretching and flattening the stomach model

where m_i and $\mathbf{a}_i(t)$ denote the mass and the acceleration of the mass point. $\mathbf{Fe}_i(t)$ is the external force working on the mass point. S_i denotes a set of mass points connected to i by Voigt elements. $\mathbf{Fs}_{ij}(t)$, the internal force by the Voigt element connecting mass points i and j, is formulated as

$$\mathbf{Fs}_{ij}(t) = k_{ij}\left(1 - \frac{\|\mathbf{r}_{ij}(0)\|}{\|\mathbf{r}_{ij}(t)\|}\right)\mathbf{r}_{ij}(t) + c_{ij}\mathbf{v}_{ij}(t), \tag{2}$$

where $\mathbf{r}_{ij}(t)$ and $\mathbf{v}_{ij}(t)$ represent the position and the velocity of j relative to i. k_{ij} and c_{ij} are the spring and damping constants.

Motion of the stomach is achieved by discretizing Eq. 1 using the Newmark-β method and then computing the acceleration, the velocity, and the position of mass points. By using a time step Δt, at time $t^{(n+1)} = (n+1)\Delta t$ $(n = 0, 1, 2, ...)$, acceleration $\mathbf{a}_i^{(n+1)}$, velocity $\mathbf{v}_i^{(n+1)}$, and position $\mathbf{r}_i^{(n+1)}$ of mass point i can be computed as

$$\mathbf{a}_i^{(n+1)} = \frac{\sum_{j\in S_i}\left[k_{ij}\left(1 - \frac{\|\mathbf{r'}_{ij}^{(0)}\|}{\|\mathbf{r'}_{ij}^{(n)}\|}\right)\mathbf{r'}_{ij}^{(n)} + c_{ij}\mathbf{v'}_{ij}^{(n)}\right] + \mathbf{Fe}_i\left(t^{(n+1)}\right)}{m_i + \sum_{j\in S_i}\left(c_{ij}\delta\Delta t + k_{ij}\beta\Delta t^2\right)}, \tag{3}$$

$$\mathbf{v}_i^{(n+1)} = \mathbf{v}_i^{(n)} + (1-\delta)\mathbf{a}_i^{(n)}\Delta t + \delta\mathbf{a}_i^{(n+1)}\Delta t, \tag{4}$$

$$\mathbf{r}_i^{(n+1)} = \mathbf{r}_i^{(n)} + \mathbf{v}_i^{(n)}\Delta t + (0.5-\beta)\mathbf{a}_i^{(n)}\Delta t^2 + \beta\mathbf{a}_i^{(n+1)}\Delta t^2, \tag{5}$$

where δ and β are parameters for controlling acceleration changes between times $t^{(n)}$ and $t^{(n+1)}$. $\mathbf{r'}_{ij}^{(n)}$ and $\mathbf{v'}_{ij}^{(n)}$ can be expressed using acceleration $\mathbf{a}_{ij}^{(n)}$ of j relative to i

$$\mathbf{r'}_{ij}^{(n)} = \mathbf{r}_{ij}^{(n)} + \mathbf{v}_{ij}^{(n)}\Delta t + (0.5-\beta)\mathbf{a}_{ij}^{(n)}\Delta t^2, \tag{6}$$

$$\mathbf{v'}_{ij}^{(n)} = \mathbf{v}_{ij}^{(n)} + (1-\delta)\mathbf{a}_{ij}^{(n)}\Delta t. \tag{7}$$

Computation accuracy and stability depend on parameters δ and β [7]. In this paper, we choose $\delta = 1/2$ and $\beta = 1/4$ to enable unconditionally stable computation.

2.2 Cutting and Opening the Stomach

The stomach model is cut and stretched until its inner surface becomes observable. First, we manually input a cutting line on the model. Any parts of the stomach can be cut and removed, however, an usual cutting line is based on the rule defined in [6]. This rule requires two cuts that remove the cardia and pylorus of the stomach and a cut along its greater curvature. Cutting is implemented as the removal of mass points and Voigt elements in the hexahedra that coincide with the cutting line. Next, we apply the following stretching force to every mass point i near the cutting line

$$\mathbf{Fo}_i(t) = \alpha\big(\mathbf{Po}_i - \mathbf{r}_i(t)\big) \;/\; \|\mathbf{Po}_i - \mathbf{r}_i(t)\|, \tag{8}$$

where $\mathbf{r}_i(t)$ is the position of the mass point, $\mathbf{Po}_i$ is a control point specified in advance, and α is a constant representing the magnitude. Finally, deformation of the model is achieved by iteratively computing Eqs. 3, 4, and 5 using $\mathbf{Fo}_i(t)$ as $\mathbf{Fe}_i(t)$.

2.3 Flattening the Stomach Wall Using Surface Normals

Flattening forces driven by surface normals are introduced to direct the opened stomach onto a plane. First, we generate this so-called supporting plane using the coordinates and normals of the mass points allocated at vertices of hexahedron faces forming the stomach outer surface. Let us denote a set of these mass points as S_V. For each mass point i ($i \in S_\mathrm{V}$) a normal $\mathbf{K}_i$ is computed as a normalized sum of the normals of the faces that form the outer surface and share mass point i. The basepoint $\mathbf{B}_\Pi$ and the normal $\mathbf{N}_\Pi$ of the plane Π (defined by $(\mathbf{x} - \mathbf{B}_\Pi)\cdot\mathbf{N}_\Pi = 0$) are determined as follows :

$$\mathbf{N}_\Pi = \sum_{i\in S_\mathrm{V}} \mathbf{K}_i \;/\; \|\sum_{i\in S_\mathrm{V}} \mathbf{K}_i\| \quad , \quad \mathbf{B}_\Pi = \arg\max_{\mathbf{r}_i,\ i\in S_\mathrm{V}} (\mathbf{r}_i \cdot \mathbf{N}_\Pi)\,. \tag{9}$$

Next, we apply the following flattening force to every mass point i ($i \in S_\mathrm{V}$) to direct it onto the supporting plane :

$$\mathbf{Fu}_i(t) = \begin{cases} 0 & (\lambda_i \le 0) \\ \mu f_\mathrm{b}\mathbf{K}_i(t) & (0 < \lambda_i \le \lambda_\mathrm{b}), \\ \mu\|\mathbf{H}_i(t)\|\mathbf{K}_i(t) & (\text{otherwise}) \end{cases} \tag{10}$$

where $\lambda_i \equiv \mathbf{K}_i(t) \cdot \mathbf{N}_{\Pi}$. $\mathbf{H}_i(t)$ is a vector heading from $\mathbf{r}_i(t)$ in direction $\mathbf{K}_i(t)$ to its intersection point with Π. μ and f_{b} are two parameters for the magnitude control of $\mathbf{Fu}_i(t)$, while λ_{b} is a threshold for the magnitude of $\mathbf{Fu}_i(t)$. Deformation of the model is achieved by iteratively computing Eqs. 3, 4, and 5 using $\mathbf{Fu}_i(t)$ as $\mathbf{Fe}_i(t)$, until $\sum_{i \in S_{\mathrm{V}}} \|\mathbf{Fu}_i(t)\| \ / \ |S_{\mathrm{V}}| < \epsilon_{\mathrm{b}}$ is satisfied, where ϵ_{b} is a threshold and $|S_{\mathrm{V}}|$ is the size of S_{V}. To keep the model lying on the supporting plane, we calculate collision detection and response for any mass point that collides with the plane.

3 Results and Discussion

3.1 Comparison of Computation Stability with the Euler Method

We simulated deformation of a phantom model to compare the stability and computation time with the Euler method. Our phantom model consists of 1,386 mass points and 10,435 Voigt elements which were generated from a brick-like dataset containing $10 \times 20 \times 5$ voxels with $d = 1$, $m = 1$, $k = 9.87$, and $c = 0.13$. We applied streching forces of same magnitude to 4 mass points at 4 corners of the model until $t = 1.0$ then released. Five snapshots of the model simulated with $\Delta t = 0.02$ are shown in the upper row of Fig. 2. From $t = 16.0$, the model computed by the Euler method (darkly depicted) was collapsed, while the model computed by the Newmark-β method (brightly depicted) mostly preserved its shape. To gain the same stability by the Euler method, we had to choose $\Delta t = 0.0005$ (the lower row of Fig. 2). The computation time then became about 35 times that by the Newmark-β method with $\Delta t = 0.02$. These results shows that the Newmark-β method is very effective for simulating fast and accurate image deformation.

3.2 Unfolding the Stomach from Preoperative CT Images

We generated unfolded views from fifteen cases of CT image datasets and compared with those generated by the conventional method [2] from several aspects such as flatness of the unfolded stomach walls, reproduction rate of their luminal surface, virtual gastroscopic images, and resected specimens. Stomach walls were modeled with $d = 8$, $m = 1$, $k = 39.48$, and $c = 3.14$. Parameters and thresholds for computation are : $\alpha = 94.75$, $\Delta t = 0.1$, $\mu = 0.25$, $f_{\mathrm{b}} = 80.0$, $\lambda_{\mathrm{b}} = 0.5$, $\epsilon_{\mathrm{b}} = 5.0$. The *flatness* d_{Π} of the unfolded stomach wall and the *reproduction rate* R_s of its luminal surface are computed as

$$d_{\Pi} = \max_i \{(\mathbf{p}_i - \mathbf{B}_{\Pi}) \cdot \mathbf{N}_{\Pi}\} - \min_i \{(\mathbf{p}_i - \mathbf{B}_{\Pi}) \cdot \mathbf{N}_{\Pi}\}, \tag{11}$$

$$R_s = (1 - N_{\mathrm{CS}}/N_{\mathrm{S}}) \times 100, \tag{12}$$

where $\mathbf{p}_i$ is the position of voxel i in the stomach wall region, N_{S} is the number of voxels belonging to the luminal surface region, and N_{CS} denotes the number

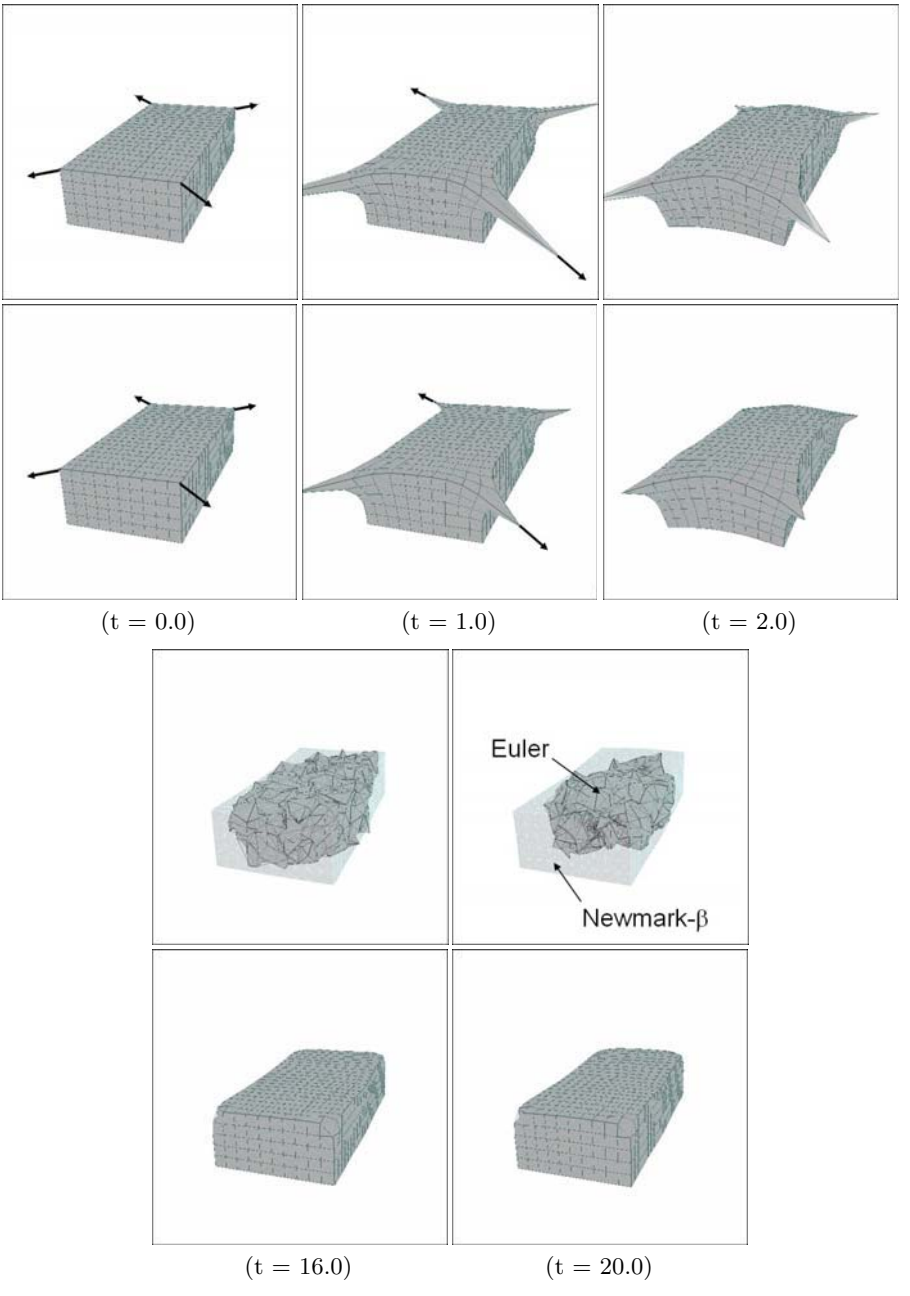


Fig. 2. Comparison of computation stability in simulating deformation of a phantom model using the same force magnitude and time step. Upper row : $\Delta t = 0.02$, lower row: $\Delta t = 0.0005$. Arrows illustrate stretching forces.

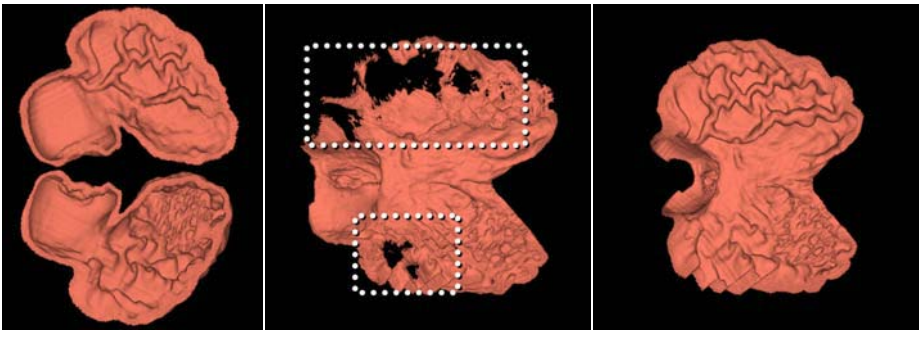


Fig. 3. Unfolding results under the same force magnitude ($\alpha = 15.0$) and time step ($\Delta t = 0.1$). (a): divided view, (b): unfolded view by Mori et al.'s method [2], (c): unfolded view by our method. Dashed rectangles indicate areas in which luminal surface was incorrectly reproduced due to collapse of hexahedra.

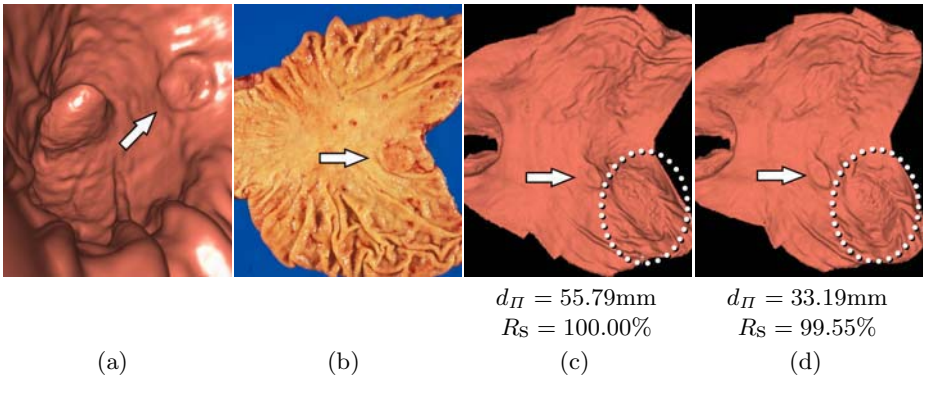


Fig. 4. Unfolding results with/without the proposed flattening process. (a): virtual gastroscopic view, (b): resected specimen, (c): unfolded view without flattening, (d): unfolded view with flattening. Dashed ellipses indicate concave areas in which unfolding is improved by the flattening process. Arrows indicate a cancer region.

of voxels that belongs to both the luminal surface region and the hexahedra collapsed during deformation.

An unfolded view generated by our method (without the flattening process) and that by the conventional method [2] using the same force magnitude ($\alpha = 15.0$) and time step ($\Delta t = 0.1$) are shown in Fig. 3. It can be seen from the unfolded view generated by the conventional method that two regions containing 49% of the luminal surface were not correctly reproduced (Fig. 3-(b)), while only

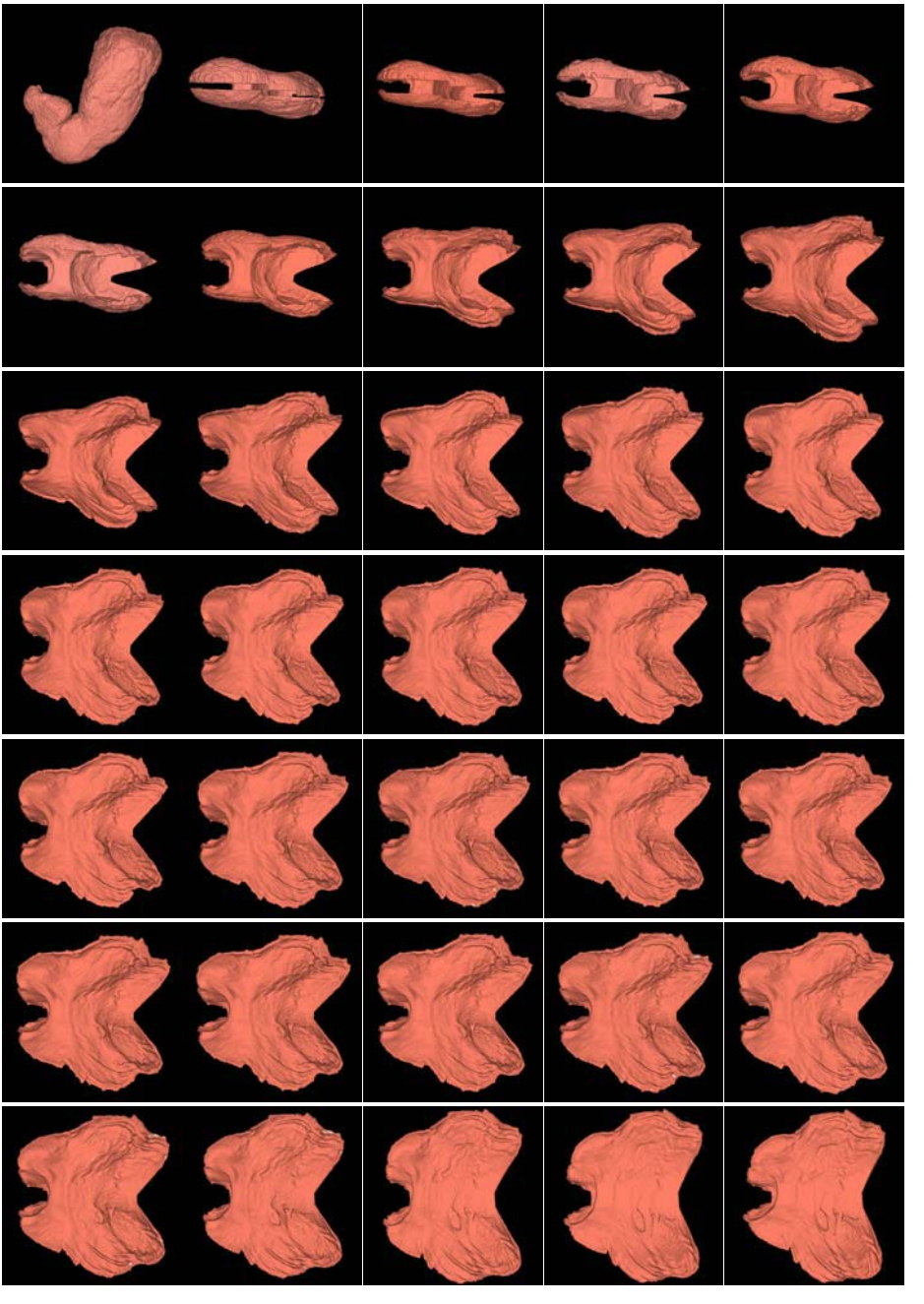


Fig. 5. Some views obtained in time sequence during simulation. (1): initial view, (2): cut view, (3)-(30): stretched views, (31)-(35): flattened views. All views except (1) were generated from the same viewpoint and view direction.

1% of the luminal surface was not reproduced by our method (Fig. 3-(c)). We also found that the conventional method gained 99% reproduction rate only with $\alpha \leq 0.15$. The computation time then became about 5 times that by our method.

Another example of unfolded views generated by our method with and without the flattening process is shown in Fig. 4. In the unfolded view generated without the flattening process, there are some concave areas in which the stomach was not unfolded to a flat shape (Fig. 4-(c)). By introducing the flattening process, the flatness of the stomach wall decreased from 55.79 mm to 33.19 mm, and about 1% of the luminal surface was not correctly reproduced. Some snapshots taken during the unfolding process are shown in Fig. 5.

By using Voigt elements for the elastic modeling of the stomach and the Newmark-β method for image deformation, our method can reconstruct image data more accurately. The unreproduced luminal surface region is 1%, which is small enough not to be detected on the unfolded views by human eyes. We presented the unfolded views from twelve datasets for surgical planning. They were considered to have well reproduced lesions as well as fold patterns observed in virtual gastroscopic images and disected specimens (Fig. 4). This is very meaningful for image-guided diagnosis in which the status of the lumen provides various information. By introducing flattening forces determined from surface normals, we can take into account the relationship between the stomach and the supporting plane for shape deformation. Thus, concave parts of the stomach, which are difficult to unfold by the conventional method, can be flattened, generating views of more precise information of the luminal surface.

4 Conclusion and Future Work

We have presented an improved method for generating unfolded stomach views based on flattening forces and stable volumetric image deformation. Experimental results showed that our method can generate unfolded views in which stomach walls are flattened better and their luminal surface is reproduced more accurately. We believe that our method can be used as one basic visualization technique for intuitive observation of the stomach from preoperative CT images. Our future work includes further evaluation of unfolded views and application to a larger number of datasets. Especially, evaluation of geometric distortions of the luminal surface, investigation of the relationship between the quality of unfolded views and the resolution of CT images, as well as quantitative comparison between unfolded views and resected stomach specimens are good issues to challenge.

Acknowledgments

We would like to thank Dr. Shigeru Nawano of the National Cancer Center East, Japan and Dr. Kunihisa Miyagawa of the National Cancer Center for providing CT images and useful comments from a medical viewpoint. We also thank

Dr. Michitaka Fujihara of the Faculty of Medicine, Nagoya University and Dr. Kazunari Misawa of Graduate School of Medicine, Nagoya University for evaluating a number of unfolded views presented in this paper and their supervision. Special thanks to the reviewers of this paper for their useful comments and suggestions. Parts of this research were supported by the Grant-In-Aid for Scientific Research from the Ministry of Education, the 21st century COE program, the Grant-In-Aid for Scientific Research from Ministry of Education, Culture, Sports, Science and Technology, Japan Society for Promotion of Science of Japanese Government, and the Grand-In-Aid for Cancer Research from the Ministry of Health and Welfare of Japanese Government. We are also supported by the Kayamori Foundation of Informational Science Advancement.

References

1. J. H. Kim, S. H. Park, H. S. Hong, and Y. H. Auh, "CT gastrography," Abdominal Imaging, Vol. 30, No. 5, pp. 549–517, 2005.
2. K. Mori, H. Oka, T. Kitasaka, Y. Suenaga, and J. Toriwaki, "Virtual Unfolding of the Stomach Based on Volumetric Image Deformation," MICCAI 2004, LNCS 3217, pp. 389–396, 2004.
3. G. Wang, E. G. McFarland, B. P. Brown, and M. W. Vannier, "GI Tract Unraveling with Curved Cross Section," IEEE Trans. Med. Imaging, Vol. 17, No. 2, pp. 318–322, 1998.
4. L. Zhu, S. Haker, and A. Tannenbaum, "Flattening Maps for the Visualization of Multibranched Vessels," IEEE Trans. Med. Imaging, Vol. 24, No. 2, pp. 191–198, 2005.
5. S. Cotin, H. Delingette, and N. Ayache, "Real-time elastic deformations of soft tissues for surgery simulation," IEEE Trans. Vis. Comput. Graphics, Vol. 5, No. 1, pp. 62-73, 1999.
6. Japanese Gastric Cancer Association ed., "Japanese Classification of Gastric Carcinom, 13th ed.," Kanehara Publishing Inc., Tokyo, Japan, 1999.
7. N. M. Newmark, "A Method of Computation for Structural Dynamics," Proc. ASCE, Vol. 85, No. EM3, pp. 67-94, 1959.

3D Cylindrical B-Spline Segmentation of Carotid Arteries from MRI Images

P. Makowski, P.J.H. de Koning, E. Angelie, J.J.M. Westenberg, R.J. van der Geest, and J.H.C. Reiber

Leiden University Medical Center, Department of Radiology,
Division of Image Processing, Albinusdreef 2, 2300RC Leiden, The Netherlands
Tel.: +31 71 526 1116; Fax: +31 71 526 6801
p.s.makowski@lumc.nl

Abstract. The paper presents a segmentation method based on 3D cylindrical B-spline model. Proposed method was applied to 5 patient MRA studies of carotid arteries and 1 phantom dataset. Carotid bifurcation was segmented as two independent, overlapping branches. The presented method was evaluated against observer drawn contours and phantom model. Statistical assessment of vessel lumen area showed 10.4% systematic underestimation and good precision (SD = 9.2 vs. $SD_{interobs}$ = 11.8) of presented method in comparison to the observer's results.

1 Introduction

1.1 Purpose of the Work, Segmentation Requirements

Many medical applications require 3D models of the luminal surface of blood vessels, which can be obtained as a result segmentation of 3D MRA or CTA data sets. Example applications of such models are the assessment of the severity of stenoses (i) and the simulation of blood flow using computational fluid dynamics (CFD) (ii). In this paper a segmentation method is presented involving a 3D cylindrical B-spline [1] with a novel approach for fitting this model to image data. We use a numerical method, previously described in [2], for fitting an elliptical model to a circular shape. The presented method required introduction of internal contracting/stiffness forces [3][4] to provide model relocation to the segmented features. We applied the method to obtain models of carotid bifurcations from 5 MRI patient studies and 1 phantom MRI. Method evaluation was performed to prove its usefulness for the quantitative assessment of vessel lumen area (i) and the accuracy of an entire 3D vessel shape representation (ii).

1.2 Alternative Segmentation Methods for 3D Models

Many segmentation methods have been proposed for generating 3D surface models of blood vessels. One of the oldest approaches is using a combination of 2D contour delineation in image cross-sections and triangulation, [5] contains a number of references to such works.

M. Harders and G. Székely (Eds.): ISBMS 2006, LNCS 4072, pp. 188 – 196, 2006.

Another example of 3D segmentation is a method combining isosurface detection with fitting the surface to the image data [6]. The first phase involves the marching cubes method [7], which produces the triangle mesh roughly representing the object surface. The triangle mesh is then matched to the voxel data by means of an algorithm involving image and smoothing forces.

One of the main drawbacks of the above-mentioned methods is the sensitivity to image artefacts and noise in the final surface model. These problems can be seen as unexpected local relocations of the final surface (slice translation, knots/protrusions or holes/concavities). Such variations can disqualify the resulting mesh if it is to be used for measuring the size of anatomical features or to define the geometrical boundary conditions for CFD application. One way to avoid this is by using an approximation of the assumed shape (in case of vessel cross-sections – an elliptic shape). In [8] and [9] ellipses were fitted to the vessel wall in planes perpendicular to the vessel axis. The obtained 2D objects were converted to 3D ellipsoids, which formed a binary object representing the vessel interior. The vessel surface was obtained then by means of isosurface triangulation and smoothing. This approach reduces the influence of image artefacts, but in many cases ellipses or even deformed ellipses [10] are too simple shapes to represent diseased blood vessels. Therefore use of the 3D cylindrical B-spline surface seems to be the most natural approach.

$$S(i,j)=\sum_{u=0}^{Mu}\sum_{v=0}^{Mv} B_{ul}(i)B_{vm}(j)\,p_{uv} \tag{1}$$

Eq. (1) defines the tube surface by means of basis functions B and control points' coordinates $\boldsymbol{p}$. As defined in [1] $\boldsymbol{p_{uv}}$ are control points, B_{ul} is the u-th B-spline periodic basis function of order l and $i\in [0,2\pi)$, B_{vm} is the v-th B-spline non-periodic basis function of order m and $j\in [0,1]$. Eq. (1) shows that control points of the tubular B-spline surface can be organized as a set of consecutive rings, located in planes perpendicular to the tube/vessel axis.

1.3 Alternative B-Spline Segmentation Methods

In [11] an example of the use of deformable B-splines for segmentation of simple tubular objects has been presented. The spline surface is defined as a number of closed curves, located in parallel planes. The surface fitting process is based on energy minimization. The total energy E is defined as a sum of internal surface energy and external energy, connected with the image gradient or the Chamfer distance. Using the conjugate gradient method the location of control points is updated in consecutive planes to minimize E.

The conjugate gradient method has also been used in [1] for 3D and in [12] for 2D B-splines, where a comparison with the steepest descent method has been presented.

Another technique to fit a 2D B-spline model has been presented in [13]. In the first phase local image minima are found. In the second phase control points are relocated to minimize the distance between the curve and local minima.

2 Image Material

Five patient examinations were used, 4 men and 1 woman at age ranging from 67 to 78 years. The data were acquired on a 1.5T MR system (SIGNA EXCITE, GE Medical Systems). For each patient two datasets were obtained. 1) After administration of a contrast agent (Gadodiamide - Omniscan, GE Amersham, Princeton, NJ) a 3D MRA gradient echo (flip angle 29, TE=1.6ms, TR=4.4ms) sequence was performed. In addition, a pre-contrast image was taken and subtracted from the contrast enhanced data set, to increase the contrast between the vessel lumen and the background. The MRA dataset reconstruction matrix was 256x256x56 resulting in a voxel size of 0.94x0.94x2mm.

2) The MRA dataset was used to plan the slice locations of the second dataset slices. This was a multi-slice T1-weighted, ECG gated, black-blood sequence acquired in planes perpendicular to the vessel axis, focusing on the carotid bifurcation. Matrix size was 256x224 with 0.54 mm pixel size in sixteen 2 mm slices. In this vessel wall MR sequence both the lumen and outer contours of the vessel wall were visualized at high resolution.

3) For a quantitative reference a third dataset of a custom made plexiglas phantom was taken using 3D contrast-enhanced MRA acquisition (3D T1-weighted spoiled gradient-echo with flip angle 40, TE=1.7ms and TR=5.7ms), on 1.5 T MR system (Intera Philips Medical Systems). Reconstruction matrix was 256x256x40 and voxel size was 1.02x1.02x2mm was used. The phantom consisted of 3 equiaxial cylinders with lumen cross-section area equal to 15.973, 32.979, 57.146 mm^2. Contrast agent (Magnevist - Berlex Pharmaceuticals, Richmond, CA) was applied in the constant flow.

3 Proposed Segmentation Method

3.1 Initialization of the Tube Center

The initial location of the cylindrical vessel model is very important for the result of the final segmentation. The initialization phase should actually choose the vessel to be segmented. We have used a previously implemented and described [14] algorithm, based on the fast marching level set method (FMLSM). It is required for the user to manually select points in the beginning and end of each vessel branch to be segmented. Then, a curve located inside the vessel, between the inserted points can be found automatically and used as the model axis.

3.2 Initialization of the Surface Location

Having the tube model with its center defined, the next initialization step is to define the location of surface. In the simplest way it can be done by setting a fixed radius for all spline segments. The initialization phase has significant influence on the decision what algorithm should be chosen to fit the tube surface to image data. If the surface is initialised too far from the desired image features it will not be possible to find them

due to a lack of influence, or influence of spurious features. Increasing the area, where image forces "emanating" from a specific feature can act on the model surface, can solve the lack of influence problem. One of the approaches is to use Gauss image filtering, resulting in blurring the image features [3]. Another, more advanced approach is use of GVF method [15] to extend the influence of the image gradient to every point in the image space. An alternative approach has been presented in the balloon model [4]. Instead of increasing the area of edge influence the fitting algorithm can be modified by adding pressure/contraction forces, which is an implicit way to define the area in image space, where the edge under segmentation can be found. This balloon approach is also used in our method, so its description will be presented in the next section.

3.3 Surface Fitting – Model Definition

In many publications concerning deformable models [3] and deformable templates [2] the Euler-Lagrange (EL) equation is used to find a minimum of the functional defining total model energy. For surface fitting we use the energy minimizing method, which involves iterative solving of the EL equation as presented in [2]. However, we restrain from defining the energy functional explicitly. Instead, we prefer to use a more intuitive description of the EL equation, which can actually be interpreted as a set of forces acting on the model. Therefore the solution of the EL equation will be equivalent with finding the location of the surface providing the equilibrium of forces acting on control points.

In [2] a method description has been provided which we find most suitable as a base for our modifications. The equation describing the dynamic template model in [2] can be presented as (2). For the clarity of description, in the beginning we will use a 2D model.

$$\gamma \frac{d\mathbf{p}}{dt} + \sum_{i=0}^{Ni-1} \left(\frac{\partial P(x_i, y_i)}{\partial x} \frac{\partial S_x}{\partial \mathbf{p}} + \frac{\partial P(x_i, y_i)}{\partial y} \frac{\partial S_y}{\partial \mathbf{p}} \right) = \mathbf{0} \tag{2}$$

In [2] **p** is a vector of ellipse parameters. In our case model parameters are space coordinates of control points. Therefore **p** is a vector of control points coordinates. P is the potential function derived from the image gradient as in eq. (3). N_i is the number of sampling points (tessellation points), index *i* will be used to indicate a sampling point in the 2D case. In [2] S_x, S_y are explicitly defined as ellipse equations, in our case S is defined by eq. (1). Therefore **S** derivatives with respect to vector **p** can be computed as a product of basis functions connected with a chosen control point coefficient p_u ($p_{u,v}$).

$$P(x, y) = -\left|\nabla I(x, y)\right| \qquad \text{or for 3D:} \quad P(x, y, z) = -\left|\nabla I(x, y, z)\right| \tag{3}$$

Our first modification of the model eq. (2) is the introduction of the pressure/shrinking forces. In the 2D model of a closed curve this force can be defined by eq. (4), where, as in [3], the stiffness force is defined. However we apply the force directly to the control point instead to the curve sampling points.

$$\mathbf{f}_u = \mathbf{v}_{u-1} - 2\mathbf{v}_u + \mathbf{v}_{u+1} \tag{4}$$

Where: $\mathbf{v}_u = [x_u, y_u]$ is a control point position and u is used to denote the control point index. Eq. (4) can be written for each control point creating a vector **f**. Eq. (4) defines the forces for a curve in the 2D case. We will keep the 2D version of this equation also in the 3D model. Eq. (4) is defined in a number of 2D planes perpendicular to the vessel axis. We also introduce constrains for the 3D model, preserving movement of control points within this planes. Eq.(2) defines a dynamic iterative process, which complemented with eq. (4) can be written as eq. (5).

$$\mathbf{p}_{uv}^{t+1} = \mathbf{p}_{uv}^{t} + \mathbf{f}_{uv} - \frac{\Delta t}{\gamma} \sum_{i=0}^{Ni-1} \sum_{j=0}^{Nj-1} \left(\frac{\partial P(x_{ij}, y_{ij}, z_{ij})}{\partial x} \frac{\partial S_x}{\partial \mathbf{p}} + \frac{\partial P(x_{ij}, y_{ij}, z_{ij})}{\partial y} \frac{\partial S_y}{\partial \mathbf{p}} + \frac{\partial P(x_{ij}, y_{ij}, z_{ij})}{\partial z} \frac{\partial S_z}{\partial \mathbf{p}} \right) \tag{5}$$

Finally we propose the model of B-spline surface with image and stiffness forces acting directly on the control points. The dynamic tube model defined by eq. 5, initialised with a certain radius outside the vessel wall, successively contracts around the vessel axis driven by vector **f.** When the image forces are strong enough to balance vector **f** the surface finally stops close to the segmented image feature. There exists a drawback in the model (eq. 5) with a quantitative influence on the results of segmentation. Due to the presence of **f** forces, the resulting mesh is slightly relocated from the position of maximum gradient towards the centre of the tube. To avoid this problem we introduce a two-phase segmentation, similar to the one described in [16]. In the first phase we use the model from eq. 5. In the second, when the surface is very close to the segmented edge, we remove contraction forces by setting **f=0**. This results in relocation of the surface to the maximum image gradient. Each phase should be stopped when the control points' movements are lower than a certain small threshold value.

4 Results

Obtained surfaces of the carotid arteries of all 5 study subjects are shown in Fig. 1. Each vessel branch was segmented separately, so the final structure consists of two overlapping tubular structures. In all patient studies a set of 7 control points per ring in 30 rings was used. Only for the phantom (Fig. 1f) we used 11 points per ring. For all presented cases a fixed initial radius of 4.0 mm was used. Typical segmentation time for the right branch (internal carotid artery) in structure (Fig. 1e) was approximately 3s for detection of the centerline and 6s for the segmentation (on a Pentium 4 – 3GHz computer).

For the evaluation we have compared the segmented structure with the manually drawn contours traced in axial cross-sectional images, based on a Multi-Plane Reformat (MPR) dataset. Axial MPR images were generated with a 1 mm slice separation. The same planes were used to obtain the cross-sections of segmented surfaces, resulting in closed contours. For cross-sections located in the bifurcation region it was necessary to combine the contours obtained from both branches, which was accomplished using methods from [17].

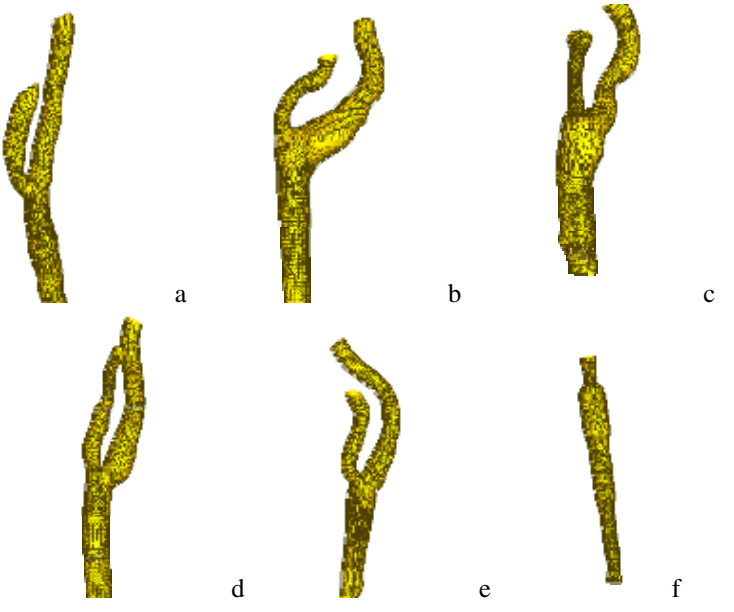


Fig. 1. Reconstructed internal surface of carotid arteries (a-e) and a phantom (f). Tessellation/sampling points are located in vertices of presented triangles.

We obtained reference contours from two observers (A, B). Contours were drawn using an in-house developed software package (VesselMASS) [18] used for quantitative analysis of vascular MR images. Observers drew contours in the MRA images (1st dataset) according to typical MR contrast enhanced examination properties. For quantitative assessment of the vessel wall location observers compared contour size with the vessel size in the corresponding vessel wall images (2nd dataset).

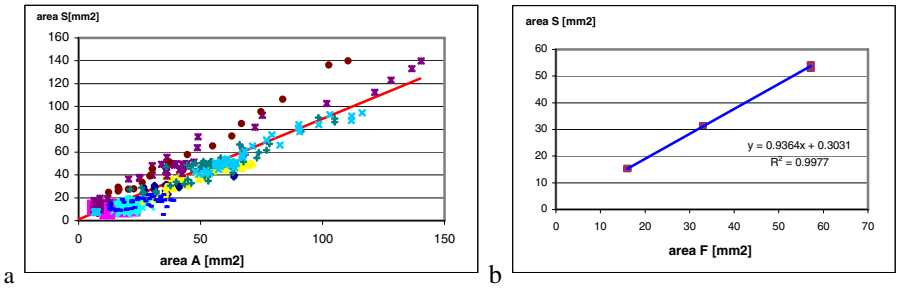
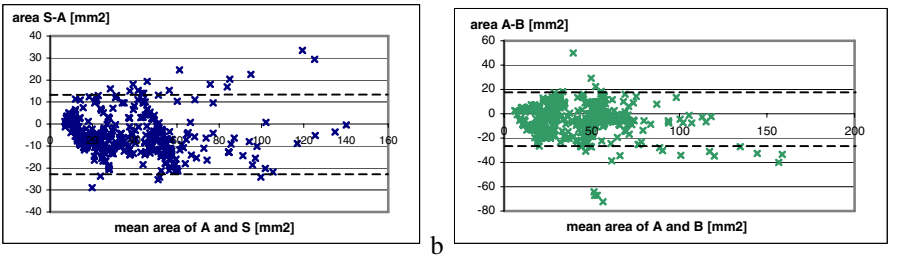


Fig. 2. (a) Correlation between the lumen area of manually drawn contours (observer A) and segmented surface, (b) between physical phantom lumen area and segmented surface

Quantitative comparison of the contour area was done for more than 350 contour pairs for each observer. We found a systematic underestimation of the vessel

area - the regression line resulting from Fig. 2a implies the value of *Ca=1.104* as the area correction coefficient. A similar conclusion was obtained also for the phantom data, but the correction coefficient was equal to *Cf=1.068* (Fig. 2b). Fig. 3a presents the same data as Fig. 2a in a form of Bland-Altman [19] graph, average value for observer A: *AVs-a= -4.7453*, standard deviation for observer A: *SDs-a=9.2326*. Inter-observer variability comparison is presented in Fig.3b, *AVa-b = -4.809*, SD_{a-b} *=11.818*. A graphical assessment of the data variations for contour areas is presented in Fig.3a, b.



a b

Fig. 3. (a) Differences between area of segmented contours and drawn by observer A against their mean [19]. (b) Inter-observer differences between contour areas of observer A and observer B against their mean. Agreement intervals (+/- 2SD) denoted by dashed lines.

5 Conclusions and Future Work

Finally a conclusion can be presented, based on SD (agreement intervals) comparison, that the used segmentation method and manual contour drawing method provide similar precision for area measurements, but for better accuracy a correction coefficient *Ca* can be used. The correction coefficient *Cf* should not be used here directly, due to differences in image acquisition conditions for in-vivo and in-vitro study. However *Cf* can be used to confirm the systematic area underestimation in the described method.

The proposed method, as many others, delineates surface/contours in the location of maximum image gradient. In case of MRA images this location does not correspond to the vessel wall location and finally causes vessel size underestimation. In case of vessel diameter measurements application, the above-mentioned correction due to statistically observed systematic error could solve the problem. This is of course not the case in modeling the geometry for CFD calculations, therefore in future work we are going to transform the image data to move the location of maximum gradient closer to the vessel wall. For CFD applications it will be also necessary to convert the vessel model to the form accepted by simulation software. We are going to modify and use methods previously described in [9] [8] and [10]. We also plan to use information obtained from FMLSM concerning the vessel diameter to initialize the model.

Acknowledgements

The work was supported by a grant from the Dutch Technology Foundation (STW grant LGT 6454).

Authors wish to thank Bruce A. Wasserman, MD, from Neuroradiology Division, Johns Hopkins Hospital, Baltimore, USA, for kindly provided image data.

References

[1] A.F.Frangi, W.J.Niessen, R.M.Hoogeveen, T. van Walsum, M.A. Viergever "Model - Based Quantitation of 3-D Magnetic Resonance Angiographic Images" IEEE Trans. on Medical Imaging vol.18,no.10, October 1999

[2] A.Scaggiante, M.Zampato, R.Frezza "Identifying and Tracking Ellipses: A Technique Based on Elliptical Deformable Templates" 10th International Conference on Image Analysis and Processing (ICIAP'99) pp. 582-587

[3] M. Kass,A.Witkin,D.Terzopoulos "Snakes: active contour models" First international conference on computer vision; 1987. pp. 259-68

[4] L.D.Cohen "On active contour models and balloons" CVGIP: Image Understanding V. 53, 2, Mar1991, pp.211 - 218

[5] T.Sorensen, S.Therkildsen, P.Makowski, J.Knudsen, E.Pedersen "A New Virtual Reality Approach for Planning of Cardiac Interventions", Artificial Intelligence in Medicine, 22/2001, 2000, pp. 193-214

[6] P.J. Yim, G.B.C. Vasbinder, V.B.Ho, P.L.Choyke "Isosurfaces as Deformable Models for Magnetic Resonance Angiography" IEEE Trans. on Medical Imaging vol.22, no.7, July 2003

[7] W.E.Lorensen, H.E.Cline **"**Marching cubes: A high resolution 3D surface construction algorithm" SIGGRAPH 87 v.21 pp.163-170

[8] E. T. Fruend , C. Baltes , P. Makowski , M. K. Kolandavel , P. Boesiger , P. G. Walker , E. M. Pedersen , S. Ringgaard "Realistic 3D numerical simulations of blood flow patterns and Wall Shear Stress in the Right Coronary Artery: A new approach combining MRI and Computational Fluid Dynamics" . Preliminary results 2003 ISMRM Toronto, Canada

[9] P. Makowski, S. Ringgaard, E. T. Fruend, E. M. Pedersen "A Novel Approach to Segmentation of Coronary Arteries in MR Images for Computational Fluid Dynamics (CFD) Simulations" 2004 ISMRM, Kyoto, Japan

[10] M.Podyma, I.Zbicinski, J.Walecki, M.L.Nowicki, P.Andziak, P.Makowski, L.Stefanczyk "Numerical Analisys of Blood Flow in Human Abdominal Aorta" Biomedicine 2005, Bologna, Italy

[11] J.Huang, A.Amini "Anatomical Object Volumes from Deformable B-spline Surface Models" Int. Conf. Image Processing, Chicago, October 98

[12] P. Brigger, J.Hoeg, M. Unser, "B-spline snakes: a flexible tool for parametric contour detection," *IEEE Trans. Image Processing*, vol. 9, pp. 1484-1496, 2000

[13] B.Levienaise-Obadia, A.Gee "Adaptive Segmentation of Ultrasound Images" Image Vision Comput. 17(8), 1999, pp.583-588

[14] P.J.H. de Koning,J.A. Schaap, J.P. Janssen, J.J.M. Westenberg, R.J. van der Geest, J.H.C. Reiber "Automated Segmentation and Analysis of Vascular Structures in Magnetic Resonance Angiographic Images" Magnetic Resonance in Medicine 50:1189–1198 (2003)

[15] C. Xu, J. L. Prince "Snakes, shapes, and gradient vector flow" IEEE Trans. on Image Processing vol. 7, no. 3 March 1998 pp359-369
[16] P.Makowski, T S Sorensen, SV Therkildsen, A Materka, H Stodkilde-Jorgensen, EM Pedersen "Two-phase active contour method for semiautomatic segmentation of the heart and blood vessels from MRI images for 3D visualization " Computerized Medical Imaging and Graphics, Elsevier No 26/2002 pp9-17
[17] B.R.Vatti “A Generic Solution to Polygon Clipping” Communications of the ACM 35(7), July 1992, pp.56-63
[18] Adame IM, van der Geest RJ, Wasserman BA, Mohamed M, Reiber JHC, Lelieveldt BPF. “Automatic segmentation and plaque characterization in atherosclerotic carotid artery MR images” MAGMA (Magnetic Resonance Materials in Physics, Biology and Medicine) 2004;16 (5): 227-234.
[19] J.M.Bland, D.G.Altman “Measuring agreement in method comparison studies” Statistical Methods in Medical Research 1999, 8 pp: 135-160

Simulation of Soft-Tissue Deformations for Breast Augmentation Planning

Liesbet Roose[1], Wim De Maerteleire[2], Wouter Mollemans[1], Frederik Maes[1], and Paul Suetens[1]

[1] Medical Image Computing (Radiology - ESAT/PSI), Faculties of Medicine and Engineering, University Hospital, Gasthuisberg, Herestraat 49, B-3000 Leuven, Belgium
liesbet.roose@uz.kuleuven.ac.be
[2] 3D Medical BV, Keizer Karel V singel 14, 5615 PE Eindhoven, The Netherlands

Abstract. Virtual surgery simulation plays an increasingly important role as a planning aid for the surgeon. A reliable simulation method to predict the surgical outcome of breast reconstruction and breast augmentation procedures would be useful for ensuring a symmetrical and naturally looking result and for communication between the surgeon and the patient. In this paper, we extend our previously developed basic framework to simulate subglandular breast implantation with a more realistic interaction model for implant and tissue. We model both the breast tissue and the implant using Mass Tensor models, based on continuum mechanics of linear elastic materials. Appropriate boundary constraints are defined to mimic the interaction between the breast and the implant model, including sliding contacts. We illustrate our approach with a preliminary validation study on 4 patients, yielding a mean error between the simulated and the true post-operative breast geometry below 4 mm and maximal error below 9 mm, which is found to be sufficiently accurate for visual assessment in clinical practice.

1 Introduction

Breast cancer is the most common cancer in women worldwide, accounting for 1 million new cases annually [1]. An operative mastectomy in which the breast is amputated will frequently be part of the treatment, but afterwards, women often choose for reconstruction with a breast implant. Besides this reconstruction surgery, there is also a growing request for breast augmentation because of aesthetic improvement, for instance in the case of congenital asymmetry of the breasts. For these augmenting and reconstructive procedures, a symmetrical and naturally looking result is of utmost importance. Therefore, a wide variety of implant sizes and shapes is available, and preoperative implant selection has become a major issue [2].

A reliable simulation method to predict the surgical outcome of breast reconstruction and breast augmentation procedures would be useful for the surgeon to ensure a symmetrical and naturally looking result, and could be a practical means of communication with the patient. Our goal is the development of a 3D

M. Harders and G. Székely (Eds.): ISBMS 2006, LNCS 4072, pp. 197–205, 2006.

interactive planning environment to enable such simulations, using a biomechanically based model of the breast.

Modeling and predicting female breast tissue deformation has gained considerable interest in various application fields during the last decade, including image registration, surgery simulation and biopsy planning. For registration purposes, most often free form deformation models are used [3], which lack the biomechanical relevance needed for our application. More appropriate are the classical Mass Spring models (MSM) or Finite Element models (FEM). MSM are attractive because of their conceptual and computational simplicity, but due to the discretization of the tissue in mass points and springs, and the cumbersome specification of system parameters [4,5], the biomechanical relevance of these models is lower than for FEM. In previous work [6], we showed that the use of MSM can cause non-physical local minima in case of large deformations, as occurring in breast augmentation. Samani et al. [7] and Azar et al. [8] proposed a FEM-based description of the breast to model the deformation of the soft breast tissue during a biopsy, and Ruiter [9] uses a similar model to register volumetric MRI-data with X-ray mammograms. These models are specifically designed for tumor detection, which requires a much higher accuracy than the application we are concentrating on.

In this paper, we use a breast model based on the Mass Tensor method (MTM) as introduced by Cotin et al. [10]. This model combines the advantages from MSM and FEM, i.e. our method is physically based, is conceptually simple and offers the possibility to compute very rapidly an estimate of the exact solution. Since the implant deforms under influence of the enclosing tissue, we use separate, but coupled, MTM for both the breast and the implant.

2 Method

Fig. 1 shows the different steps in our general framework, from data acquisition to simulation. In this section, we will discuss some aspects of this framework into more detail.

2.1 Data Acquisition and Mesh Building

The starting point for the simulation pipeline is the generation of a geometrical model of the individual anatomy of the patient. The developed framework focuses on subglandular placement only, where the implant is placed on top of the pectoralis major muscle. Since muscular tissue is very stiff compared to the breast tissue and the silicone or saline implant, the muscle is assumed to maintain its original shape. Therefore tissue deformation is computed only for the tissue volume enclosed between the outer muscle surface and the skin.

In previous work [11], we presented a method to construct a volumetric model based on pictures taken with a 3D camera and tissue thickness estimates. Compared to classical MR- or CT-based methods, our method is cheap and fast, and enables us to acquire images in standing position. The volumetric breast model

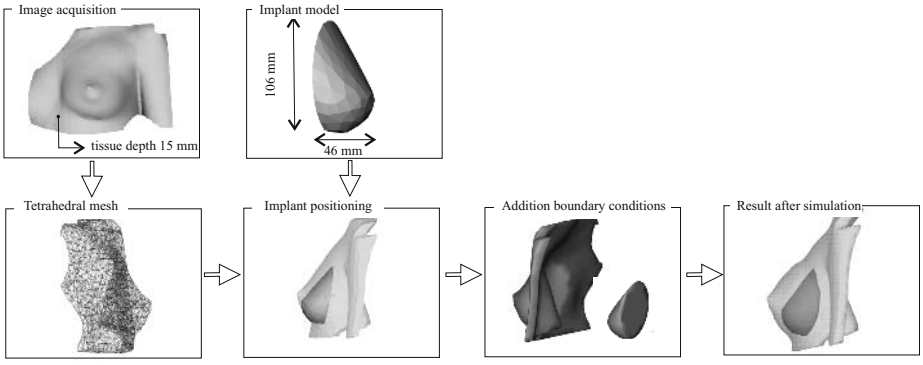


Fig. 1. Flowchart illustrating the course of model building and simulation of breast augmentation.

is represented with an unstructured tetrahedral mesh. According to conclusions from previous research [6,12], where various elasticity models were investigated, the behavior of the breast tissue is modeled as a homogeneous, isotropic, linear elastic material with a Young's modulus of 48 kPa and a Poisson coefficient of 0.45 [13].

Different shapes of implants are available for the surgeon to create a suitable form of the breast [14]. Hence, we constructed a library of tetrahedral mesh representations for a set of frequently used silicone breast implants, based on the specifications of the manufacturers.

2.2 Boundary Conditions

During the surgical intervention, the breast tissue is separated from the muscle surface with an incision plane called the pocket plane. The implant is placed with its ground plane on this pocket plane, between the muscle and the breast tissue. Evidently, the insertion of the implant causes the breast tissue to move upwards, while at the same time the stretched breast tissue exerts a compressive force on the implant. To simulate the deformation of the implant and the breast tissue, we assign boundary conditions to both meshes in order to mimic their effect on each other.

First a set of boundary conditions is applied to the implant to simulate its alignment with the muscle surface. Therefore all points on the ground plane of the implant become sliding points: they remain connected to this muscle surface but are able to slide over it. Then the points on the muscle-fat interface which are lying underneath the implant are marked as sliding points, connected to the upper surface of the implant. The other points on the muscle-fat interface are designated as immoveable points since they are supposed to be connected to the muscle, except for a small ring of points surrounding the set of sliding points. These points remain free points without constraints on their movement,

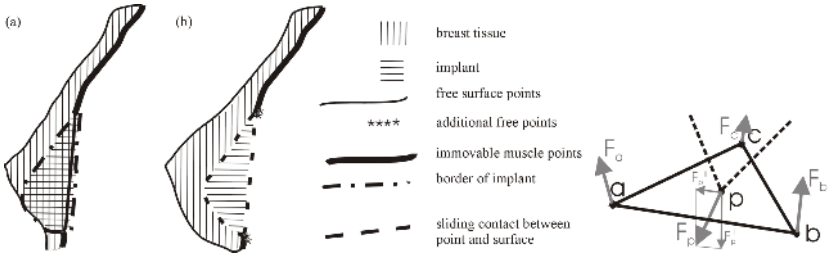


Fig. 2. Left: Sagittal cross-section of the breast to illustrate the boundary conditions imposed: (a) pre-operative view of the tissue, with the implant superposed; (b) post-operative view. Right: Sketch of a sliding point p and its connected surface abc.

to mimic the incision of the pocket plane, which is typically slightly larger than the implant ground plane.

2.3 Force Computation

To compute the evolution of the tissue after application of the boundary constraints, we need to determine the forces in the meshpoints, due to elastic deformation and the interaction between the two meshes.

Elastic Forces. We follow the general approach proposed by Cotin et al. [10] to compute the internal elastic forces at each vertex of the mesh, resulting from the deformation of the mesh. This deformation is characterized by the displacements $\mathbf{U}_i$ in the individual vertices. The elastic force $\mathbf{F}_i$ applied on vertex i, corresponds to the derivative of the elastic energy with respect to the position $\mathbf{P}_i$:

$$\mathbf{F}_i = \frac{\delta W_{elastic}}{\delta \mathbf{P}_i} = \frac{\delta \left[\frac{\lambda}{2}(trE)^2 + \mu trE^2\right]}{\delta \mathbf{P}_i} \tag{1}$$

with E the Green-St.Venant strain tensor and λ and μ the Lamé coefficients characterizing the elasticity and compressibility of the material. Using the tetrahedral mesh structure, this force can be computed as

$$\mathbf{F}_i = [K_{ii}]\mathbf{U}_i + \sum_{j \in N(i)} [K_{ij}]\mathbf{U}_j \tag{2}$$

where $N(i)$ is the set of points connected to i and $[K_{ij}]$ are tensors which are only depending on material characteristics and the rest position of the tetrahedra.

Sliding Contact Forces. To simulate the interaction between tissue and implant mesh, we add contact forces to the sliding points and to its connected surface. Suppose a sliding point p with position $\mathbf{P}_p = \alpha\mathbf{P}_a + \beta\mathbf{P}_b + \gamma\mathbf{P}_c$ connected to triangle abc (fig. 2). We decompose the elastic force acting on a point in a component $\mathbf{F}^{\perp}$ orthogonal to its connected surface and a component $\mathbf{F}^{\parallel}$

parallel to this surface. Since we assume ideal sliding contact, only the orthogonal component of the elastic force on p will exert an effect to surface abc. To preserve conservation of force and moment, contact forces $\alpha\mathbf{F}_p^{\perp}$, $\beta\mathbf{F}_p^{\perp}$ and $\gamma\mathbf{F}_p^{\perp}$ are added to respectively point a, b and c. To compute the contact force on point p, exerted by surface abc, the derivative of the elastic energy in the tetrahedra with a, b or c as vertex, is computed relative to a displacement of abc along it normal.

2.4 Iterative Solution Strategy

To find a new equilibrium configuration for the mesh after application of the boundary conditions, we search the configuration in which the forces in the free points and the parallel components of the forces in the sliding points are minimal.

$$\mathbf{U} = \arg\min \left(\sum_{i \in P_{free}} \mathbf{F}_i(\mathbf{U}) + \sum_{i \in P_{sliding}} \mathbf{F}_i^{\parallel}(\mathbf{U}) \right) \tag{3}$$

We solve this optimization problem with a quasi-static approach: the position of a point at time $t+1$ is the solution of the static problem with boundary conditions given at time t to let the force in point p be zero. From equation (2) the necessary displacement $\mathbf{U}_i^{t+1}$ in iteration $t+1$ for vertex i can be derived:

$$[K_{ii}]^{-1}(\mathbf{F}_i^t - \sum_{j \in N(i)} [K_{ij}]\mathbf{U}_j^t) = \mathbf{U}_i^{t+1} \tag{4}$$

The advantage of this approach, compared to solution schemes that directly solve the matrix representing the FEM equations, is that the iterations can be stopped as soon as sufficient accuracy is reached, instead of computing the exact solution. This results in an substantial time gain. Moreover, this solution scheme is conceptually simple and has good convergence properties.

The iterations are stopped at iteration t where the maximal movement in one iteration $||\mathbf{U}_i^t - \mathbf{U}_i^{t-1}||$ over all points drops below a threshold D_{stop}:

$$\max_i(||\mathbf{U}_i^t - \mathbf{U}_i^{t-1}||) < D_{stop} \tag{5}$$

This corresponds to Jacobi relaxation as described in [15].

3 Validation

3.1 Required Accuracy

An interesting issue is the accuracy that is clinically required for this application. We strive for a realistically looking result, and we have to investigate which simulation error is acceptable to obtain a convincing simulation result, compared to the real result after surgery, assessed by visual inspection. The required accuracy was estimated by comparing the skin surface of left and (mirrored) right

breast in post-operative images of 5 patients whose breasts were qualified as being symmetric by both the patient and the surgeon. The mean error μ_d between two breasts of the same patient was in the range between 2.7 mm and 3.5 mm and the maximal distance d_{max} was found to be between 5.2 mm and 9.4 mm.

3.2 Validation Strategy

To investigate whether we reach this accuracy, we validated our simulation method on 4 patient cases (8 breasts). We have pre- and postoperative 3D pictures for each of the patients at our disposal, for both the left and the right breast. The post-operative pictures are taken approximately three months after surgery, to ensure that the tissue had recovered from the surgical intervention and that swelling due to the surgery has disappeared. All images are taken with the Rainbow 3D camera (Genex Technologies, Kensington, USA). Implant sizes used range from 245 ml to 295 ml. To evaluate our methods, the simulation results are compared with the real post-operative image data.

First, the pre- and post-operative images have to be registered based on common rigid structures in the images, such as the skin surrounding the breast, parts of the belly and the skin around the clavicle, as far as they are visible in the images. The common parts of the skin are isolated in the post-operative image (see fig. 3), and this surface is registered with the pre-operative surface, using the well-known Iterative Closest Point (ICP) algorithm [16]. The resulting transformation is then applied to the original post-operative surface.

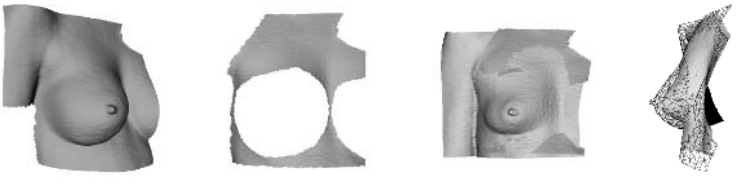


Fig. 3. (a) Post-operative picture. (b) Selection of useful structures in post-operative surface. (c) Registration of post-operative picture with the pre-operative picture. (d) Combination of post-operative (wireframe) and pre-operative picture.

However, it can be seen in Fig. 3 that the alignment of the two surfaces is far from perfect. There are different reasons for this poor quality of registration. First, the skin surface of the patient can change considerably during the three months betweens the two acquisitions, for example when the patient gains or loses weight. Second, there are slight deviations in the patients' position, mostly due to breathing and the positioning of the arm. This makes the region of the shoulder not suitable to use for the registration. The remaining area to be used for the registration is thus limited and lacks salient features for registration.

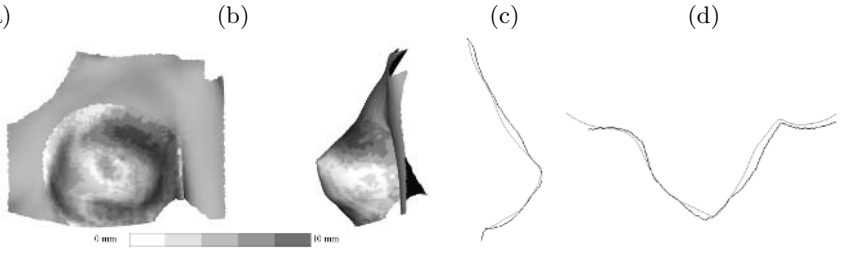


Fig. 4. Visualization of the simulation error: (a) and (b) show the error distribution on a frontal and lateral view; (c) and (d) show breast contours in a horizontal and sagittal plane (grey for simulated surface, black for real post-operative surface)

As a consequence, it is important to start the ICP computation with a manual initialization. However, for some cases, the resulting registration is clearly incorrect, which can for instance be seen if the position of the nipple is manifestly different in the pre-operative and registered post-operative picture. This was the case for patient 3 and patient 4, and for these cases we had to register the two surfaces manually.

After the registration, a volumetric model is built based upon the pre-operative picture and after insertion of the implant, the simulated post-operative breast surface is computed. Since it was not possible to acquire a post-operative CT- of MR-scan to know the exact position of the implant, this position has to estimated from the post-operative 3D picture.

3.3 Results

To qualify the difference between a simulation and the real post-operative result we measure the maximum distance d_{max} and the mean distance μ_d between both surfaces, computed over the breast surface. Fig. 4 shows a visualization of the simulation error for one case. Mean and maximal error for the 4 patients are listed in table 1.

Table 1. Quantitative evaluation of our simulation results. This table lists the maximal and mean difference between our simulation results and the real post-operative picture.

	left breast		right breast	
	max (mm)	mean (mm)	max (mm)	mean(mm)
patient 1	8.1	2.5	6.9	2.4
patient 2	8.3	3.8	9.0	3.4
patient 3	8.5	3.8	8.8	3.9
patient 4	8.9	2.6	8.1	3.6

4 Discussion and Future Works

We developed a prototype planning system to predict the outcome of a breast augmentation, which only requires a 3D picture of the breast and an estimate of the tissue thickness as input and used an adaptation of the Mass Tensor method to compute the deformation of the implant and the breast tissue. In this paper, we presented an extension of our previous work by introducing simultaneous deformation of the implant and the tissue, combining both with a sliding contact for a more realistic mimicking of the interaction between them. A preliminary validation study on this approach shows a maximal error of 9 mm and a mean error below 4 mm. From comparison of these values with the differences measured between symmetrical appearing breast, we can conclude that our simulations already give a reasonably good result for visual assessment in clinical practice.

Tetrahedral meshes have been used, but to ensure good convergence of our simulation method, initial meshes of high quality are needed. Therefore we will explore the use of hexahedral meshes which are better suited for large deformations. Further future works include the addition of gravity forces to our model and the extension to other implantation methods, such as subpectoral implantation, where the implant is placed partly under the muscle.

Acknowledgements

This work is part of the Flemish government IWT GBOU 020195 project on Realistic image-based facial modelling for forensic reconstruction and surgery simulation and K.U.Leuven/OF/GOA/2004/05.

The author is Research Assistant of the Fund for Scientific Research - Flanders (Belgium)(FWO-Vlaanderen).

References

1. Stewart, B., Kleihues, P.: World cancer report. Oxford University Press (2003)
2. Hudson, D.: Factors determining shape and symmetry in immediate breast reconstruction. Annals of Plastic surgery **52**(1) (2004) 15–21
3. Rueckert, D., Sonoda, L., Hayes, C., Hill, D., Leach, M., Hawkes, D.: Nonrigid registration using free-form deformations: application to breast MR images. IEEE Trans. Medical Imaging (1999)
4. Bianchi, G., Solenthaler, B., Szekely, G., Harders, M.: Simultaneous topology and stiffness indentification for mass-spring models based on fem reference deformations. Proceedings 7th international conference on medical image computing and computer-assisted intervention - MICCAI 2004 (2004)
5. Gelder, A.V.: Approximate simulation of elastic membranes by triangulated spring meshes. Journal of graphics tools **3** (1998) 21–42
6. Roose, L., Maerteleire, W.D., Mollemans, W., Suetens, P.: Validation of different soft tissue simulation methods for breast augmentation. In Lemke, H.U., ed.: Computer assisted radiology and surgery. Volume 1281 of ICS. (2005)

7. Samani, A., J.Bishop, Yaffe, M., Plewes, D.: Biomechanical 3-D finite element modeling of the human breast for MR/X-ray using MRI data. IEEE Trans. Medical Imaging **20**(4) (2001) 271–279
8. Azar, F.S., Metaxas, D.N., Schall, M.D.: Methods for modeling and predicting mechanical deformations of the breast under external pertubations. Med. Image Anal. **6**(1) (2002) 127
9. Ruiter, N.: Registration of X-ray mammograms and MR-volumes of the female breast based on simulated mammographic deformation. PhD thesis, Universitaet Mannheim (2003)
10. Cotin, S., Delingette, H., Ayache, N.: A hybrid elastic model allowing real-time cutting, deformations and force-feedback for surgery training and simulation. The Visual Computer **16**(8) (2000) 437–452
11. Roose, L., Maerteleire, W.D., Mollemans, W., Maes, F., Suetens, P.: Pre-operative simulation and post-operative validation of soft-tissue deformations for breast implantation planning. In: Proceedings SPIE medical imaging: visualization, image-guided procedures, and display. (2006)
12. Tanner, C., Degenhard, A., Hayes, C., Sonoda, L., Leach, M., Hose, D., Hill, D., Hawkes, D.: Comparison of biomechanical breast models: A case study. In: Medical Imaging: Image Processing. Volume 4683 of Proc. SPIE. (2002) 716–731
13. Bakic, P.: Breast Tissue Description and Modeling in Mammography. PhD thesis, Lehigh University, USA (2000)
14. Spear, S.L., Bulan, E.J., Venturi, M.L.: Breast augmentation. Plastic and Reconstructive surgery **114**(5) (2004)
15. Delingette, H., Ayache, N.: Soft tissue modeling for surgery simulation. In Ayache, N., ed.: Computational Models for the Human Body. Handbook of Numerical Analysis (Ed : Ph. Ciarlet). Elsevier (2004) 453–550
16. Besl, P.J., McKay, N.D.: A method for registration of 3-D shapes. IEEE Trans. Pat. Anal. and Mach. Intel. **14**(2) (1992) 239–256

Local Mesh Adaptation for Soft Tissue Simulation

Céline Paloc[1,2], Alessandro Faraci[1], and Fernando Bello[1]

[1] Imperial College London, UK
[2] VICOMTech, San Sebastian, Spain

Abstract. This paper addresses the problem of graphically modelling the realistic behaviour of deformable tissue that can undergo structural modifications. Building on traditional modelling methods, we propose the online remeshing of a volumetric deformable model for locally adapting the underlying mesh and thus optimising the computational workload. Our technique overcomes limitations of previous methods that made it difficult to modify the topology of the mesh online. The performance of our methodology is demonstrated for each of the two main tissue modelling methods: mass-spring systems and finite element methods.

1 Introduction

Cutting, tearing, and carving are some of the most common tasks performed by surgeons in open and minimally invasive surgery. While the ability to simulate the accurate deformation of soft tissue is important, a model without the capability to modify the simulated tissue has limited utility.

Considerable efforts have been made to improve on the modelling of soft tissue, focusing on two main approaches, the mass-spring system (MSS) and the finite element model (FEM), both being relatively unsatisfactory. MSS have limited accuracy but require few computations, while FEMs compute accurate and complex deformation where calculation speed is not a paramount concern.

Both MSS and FEM require a discretisation of the soft tissue object with a two-dimensional or three-dimensional mesh. The accuracy of a mesh in representing a given object is related to the mesh resolution. For both MSS and FEM, a high resolution or high number of cells of small size is required in order to produce accurate physical behaviour, but the computational load increases with the resolution. The local adaptation concept uses the fact that the highest possible accuracy is not always required in every part of an object. Sufficiently high accuracy for a specific task can be achieved by locally adapting the resolution of a mesh in different parts, thus reducing processing costs and memory space.

There have been several attempts at applying local mesh adaptation to tissue modelling ([1], [2], [3]). However these models rely on a comprehensive structure built in an expensive pre-processing phase and strongly dependent on the geometry of the object, thus simulating dynamic structural modification becomes

M. Harders and G. Székely (Eds.): ISBMS 2006, LNCS 4072, pp. 206–214, 2006.

difficult or impossible. The work described in this paper proposes a methodology to locally adapt the resolution of the mesh by re-tesselating it dynamically in and around the region of interest. Our methodology is applied to both MSS and FEM, and does efficiently balance execution speed and physical accuracy of traditional tissue models.

2 Adaptive Remeshing of Tetrahedral Representation

2.1 Local Mesh Refinement

The method described in this paper assumes an initial coarse Delaunay tetrahedralisation of the tissue model. The local refinement is based on the incremental insertion algorithm, commonly used in mesh generation [4]. It maintains a Delaunay tetrahedralisation into which vertices are inserted one at a time. For each vertex v to be inserted into the existing Delaunay tetrahedralisation:

1. v is located by searching the adjacent tetrahedra.
2. Depending on the location of v, the adjacent tetrahedra are split according to the configurations shown on Figure 1.
3. The Delaunay property is restored by using face or edge flips (Figure 2).

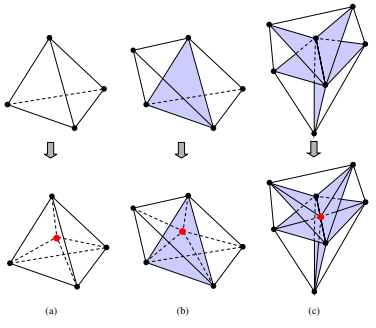


Fig. 1. Incremental insertion of a new vertex in a tetrahedral mesh. (a) In a tetrahedron. (b) On an (inner) face. (c) On an (inner) edge.

Despite the good performance of the incremental insertion algorithm, the generation of poor quality elements (such as flat tetrahedra or *slivers*) can not be avoided. However it is particularly important to preserve the quality of the elements, because elements with large or small angles can degrade the numerical solution. This problem is solved by inserting an additional vertex at the circumcenter of a triangle or tetrahedron of poor quality [4].

Although this method has been proved to always produce a result, there is no guarantee that all the generated new elements are larger than a specific value. This is critical in simulation where the integration time step varies with the size of the mesh elements. Decreasing the time step can drastically slow down the

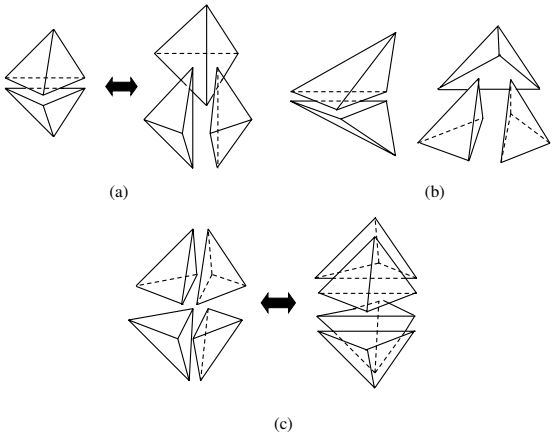


Fig. 2. Tetrahedral flip. (a) 2-3 or 3-2 flip. (b) Unflippable case. (c) 4-4 flip.

simulation. A trade-off between size and quality of the mesh elements is therefore required. Our proposed solution is to add constraints on the quality refinement by controlling the size of the generated elements. It is particularly important when the mesh size and/or topology vary over the simulation, which is the case for any adaptive models, but also when structural modifications are performed on the model. For this reason the insertion of vertices whose local feature size falls below a threshold is rejected.

2.2 Local Mesh Decimation

A mesh region refined during the simulation by the insertion of new vertices might not require the additional elements later in the simulation. The implementation must thus allow to re-simplify the mesh region previously refined.

Because the refinement method described previously inserts new vertices in the tetrahedralisation, we focused on algorithms which simplify the tetrahedralisation by removing these same vertices from the mesh. We used the *vertex decimation* algorithm proposed in [5]. A general tetrahedralisation algorithm attempts to fill the hole created by the removal of a vertex or a set of adjacent vertices (Figure 3). If the algorithm is unable to tetrahedralise the hole, the candidate decimation vertex can not be removed and the vertex is added to the end of the current decimation list for later reconsideration.

2.3 Regions of Adaptation

The mesh should be refined where it is too coarse to exhibit a realistic behaviour, while it should be simplified where a high accuracy is no longer required.

During a simulation, different regions might require a higher accuracy and should be refined for a more realistic behaviour. They have been identified as regions of high deformation; regions of high curvature; regions of contact.

In surgical simulation, tissue being torn is an example of high deformation. The regions subject to the highest deformation should be highly refined in order

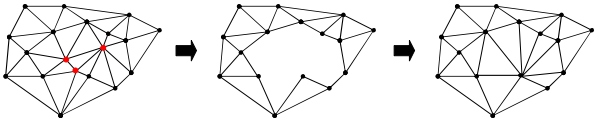


Fig. 3. Local mesh decimation based on vertex removal (2D)

to exhibit a realistic modification of the tissue structure. The area of an organ being modified by the environment or by its own deformation should be refined to model a plausible deformation at the flexion point. Finally, contact can occur between soft tissues, soft tissue and surgical instrument, and soft tissue and other anatomical structures such as bones. The soft tissue model should be refined in the area of contact to model the collision accurately.

3 Application to Mass-Spring System

In [6], we presented a volumetric MSS which stays self-consistent despite changes in resolution. Smooth behaviour of the model at different and combined resolutions is achieved by carefully defining the system parameters (masses, springs stiffness and damping). Building on our model, we now introduce the adaptive remeshing described in the Section 2.

The adaptive remeshing is applied to the mesh at its resting configuration which can be refined by inserting new vertices. The attributes of a newly inserted vertex (position and velocity in the mesh under deformation) are not known *a-priori* and must be computed. We consider the local position of the vertex in the mesh at its resting configuration by computing the barycentric coordinates α_i of the vertex relatively to its neighbours. We then infer the vertex attribute a by linear interpolation of the neighbours attributes a_i (Figure 4).

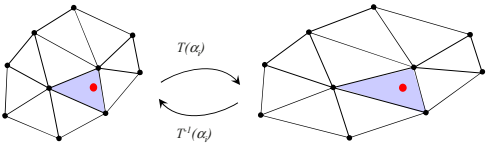


Fig. 4. The local position of a new vertex is defined relatively to its neighbours in the mesh at its resting configuration. Its position in the mesh under deformation is then computed by linear interpolation.

Let n being the number of neighbours, with $n = 4$ if the vertex is located in a tetrahedron, 3 if on a face, 2 if on an edge:

$$a = \sum_{i=0}^{n} \alpha_i a_i \tag{1}$$

The interpolation function allows the computation of the position of new vertices in a mesh under deformation while preserving consistency. For some attributes, the assumption of linearity might seem very restrictive. In practice the method is highly satisfactory as no visual or physical artifacts are observed when new vertices are inserted into the mesh.

Besides the vertex attributes, other physical parameters of the model, such as the point masses and the spring values (stiffness and damping) which are strongly dependent on the topology of the mesh, must be updated after each refinement or simplification. Those parameters are computed online according to the formulas given in [6] which were chosen to guarantee the most consistent behavior of the MSS despite changes of the underlying mesh. Moreover, the remeshing procedures being local, only the masses and springs whose neighbourhood have been modified by the adaptive procedure need to be updated, which allows to keep the computational load low.

Quantitative and qualitative experiments were carried out to evaluate the resulting model. The same simulation was run three times, using a low-resolution mesh, a high-resolution mesh and an adaptive mesh, dynamically refined in regions of interest and simplified in previously refined distant regions. Each set of experiments was repeated for several tissue models of different complexity. The computational time during one of the experiments is given for 100 iterations in Figure 5.

For each tissue model we observed that the high-resolution and the adaptive meshes exhibit similar behaviour, while the low-resolution meshes suffer from a lack of accuracy in the region of interaction. Moreover, due to the small number of points and springs, the low-resolution mesh is more subject to inverted tetrahedra which can lead in extreme cases to an inaccurate rest configuration. The high resolution mesh tends to prohibit interactive simulation frame rates, whereas the adaptive mesh preserves a low computational time despite the fact that the mesh needs to be updated in the region of interest.

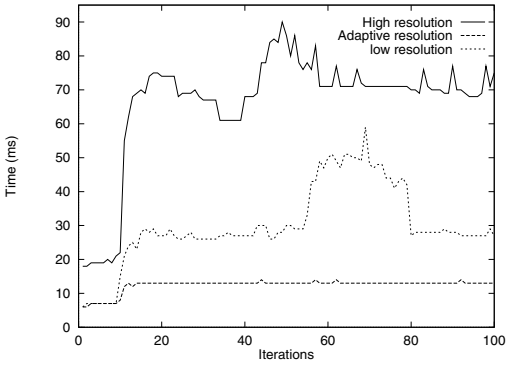


Fig. 5. Computational time during probing simulations of a MMS for different resolutions: *low* (91 tetrahedra), *high* (415 tetrahedra), *adaptive* (91-415 tetrahedra). The simulations were run on a single processor of an SGI Onyx2 (195MHz MIPS R10K).

The system allows interacting in real time (30Hz) with an initial mesh of about 100 mass points by setting the elastic modulus to $10kPa$, the density to $10^3 kg.m^{-3}$ and the minimum edge length to a quarter of the initial minimum edge length. This performance is similar to the multiresolution model of [2]. However, this model suffers from troublesome popping artifacts during changes of resolution, whereas the adaptation of our model resolution is totally transparent to the user. In addition, the model in [2] relies on pre-computed meshes and the capability to perform structural modifications has not been demonstrated, unlike the model described in this paper which allows such modifications.

4 Application to a Nonlinear Finite Element Model

In [7], we presented a volumetric nonlinear FEM allowing real-time interaction through haptic feedback using the concept of active tetrahedra.

We define an active node as a point of the mesh whose internal force vector ensures a change in the configuration of neighbouring tetrahedra. Also, a node becomes active when an external force is applied to it. We define a tetrahedron as active if at least one of its four nodes is active. If at a certain time step all the nodes of an active tetrahedron present very small displacements, velocities and accelerations, the tetrahedron becomes inactive and the values of displacements, velocities and accelerations of its nodes are set to zero. A smaller number of active tetrahedra are used during simulation, by disregarding those tetrahedra that deform very little, i.e. whose nodes are moving less than a given threshold and accelerations and velocities are also below certain values.

We now insert new nodes in the region of contact between our model and the surgical instrument using the refinement procedure described in Section 2. Figure 6 shows the new internal nodes (indicated by the two arrows) which have been added in real time inside the mesh of a liver, mainly added close to the region of contact. More internal nodes further away from the contact region will be added if the deformation spreads further.

During deformation, adding new nodes means that mechanical properties of nodes and tetrahedra have to be updated. Every time new nodes are inserted in the mesh, the quantities of each tetrahedral element are updated: current displacement, previous displacement and mass vectors.

For each new tetrahedron Ts_i $(i = 1..4)$ we calculate the entries corresponding to the newly inserted node $\mathbf{P}$ for the element displacement at the previous time step $\mathbf{u}^e_{t-\Delta t}$, the element displacement at current time step $\mathbf{u}^e_t$ and the element mass vectors $\mathbf{M}^e$ as follows:

$$\mathbf{u}^e_{t-\Delta t}(\mathrm{P}) = \frac{\mathbf{u}_{t-\Delta t}(\mathrm{P})}{4} \qquad \mathbf{u}^e_t(\mathrm{P}) = \frac{\mathbf{u}_t(\mathrm{P})}{4} \qquad \mathbf{M}^e(\mathrm{P}) = \frac{\mathbf{M}(\mathrm{P})}{4} \quad (2)$$

where $\mathbf{u}_{t-\Delta t}$, $\mathbf{u}_t$ and $\mathbf{M}$ are vectors for the overall mesh.

The above expressions ensure consistency between the element and general quantities since the following equalities have to be satisfied:

$$\sum_{e=1}^{4} \mathbf{u}_{t-\Delta t}^{Ts_i}(\mathrm{P}) = \mathbf{u}_{t-\Delta t}(\mathrm{P}) \qquad \sum_{e=1}^{4} \mathbf{u}_{t}^{Ts_i}(\mathrm{P}) = \mathbf{u}_{t}(\mathrm{P}) \tag{3}$$

$$\sum_{e=1}^{4} \mathbf{M}^{Ts_i}(\mathrm{P}) = \mathbf{M}(\mathrm{P}) \tag{4}$$

Element stiffness and damping matrices are updated for each new tetrahedron.

We carried out evaluation on a tetrahedral mesh made of: 1019 external nodes; 557 internal nodes (1576 total nodes in the mesh); 5983 tetrahedra. Elastic modulus, Poissons ratio, damping, density and time step were set respectively to $4KPa$, 0.475, 200, $1050Kg/m^3$, 0.0005 sec. During the simulation, an average of 150 tetrahedra were simultaneously active in the vicinity of the surgical tool. Refinement was performed in that same active region.

The frame rate per second (fps) was higher than the requested value of 30 for an acceptable visual rendering and a haptic frame rate per second (hps) maintained at 1000 throughout the simulation, ensuring the required rate necessary for an effective haptic feedback. Since the simulation involved the integration of haptic feedback, a double processor PC was used to test our model. The simulation could achieve real-time performance because not all the nearly 6000 tetrahedra were employed by the FEM simulation, but only a small proportion (2.5%) of them, corresponding to the 150 active tetrahedra close to the region of contact.

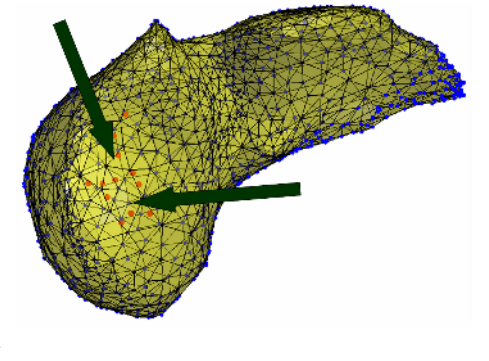


Fig. 6. Initial surface nodes of a FEM liver model. Internal nodes (indicated by the arrows) are inserted locally in real time.

5 Demonstration of Cutting

The adaptive remeshing of the underlying mesh permits dynamic structural modifications of the model undergoing deformation. This ability directly leads to a

new concept for performing tasks involving structural modifications, such as cutting. Building on our adaptive model, we propose a new approach to the cutting problem using online remeshing in the region of the cut. The mesh is refined and accommodated along the incision by inserting new elements (vertices and external faces). The approach was integrated and tested in our MSS implementation (Figure 7) [8].

Our proposed method allows the simulation to start with a coarse mesh while existing methods require a high resolution mesh to simulate a cut of the same accuracy [9]. Furthermore, serious drawbacks of previous methods, such as the generation of small and degenerate elements, are overcome thanks to our quality and size control. We have also shown that the computational cost of our approach is suitable for real-time simulation. Those advantages contribute to a major improvement of the overall efficiency during cutting simulation.

Fig. 7. Grasping and cutting a MSS liver model based on online remeshing

6 Conclusion

This paper described a methodology to locally adapt a soft tissue model by modifying dynamically its mesh topology, using Delaunay refinement, and adapting the algorithms to perform local refinement and simplification of MSS and FEM.

For a mesh of moderate size, the adaptive approach offers the same accuracy as a fine mesh while providing higher frame rate. Moreover, unlike previous models based on multiresolution, the change of the model resolution is totally transparent to the user (without any popping artifacts).

The main advantage of the tetrahedral Delaunay refinement proposed in this paper is to allow arbitrary structural modifications by insertion of new mesh points. Moreover, the main characteristic of Delaunay refinement is preserving the quality of the mesh while ensuring stability of the numerical solution.

Despite a significant reduction on the computational cost due to the adaptive framework, the mesh complexity compatible with real time requirements is still low. Simulating a complex and realistic tissue behaviour in real time

remains challenging. However, we have demonstrated the promising potential of the adaptive approach. While hardware acceleration will improve, dynamic adaptivity can form the foundation of real-time deformable modelling entailing complex interactions, making possible the modelling of high-fidelity simulation.

References

1. Debunne, G., Desbrun, M., Barr, A., Cani, M.P.: Interactive Multiresolution Animation of Deformable Models. In: Eurographics Workshop on Computer Animation and Simulation. (1999)
2. Debunne, G., Desbrun, M., Cani, M.P., Barr, A.H.: Dynamic Real-Time Deformations using Space and Time Adaptive Sampling. In: Computer Graphics Proceedings. Annual Conference Series (2001)
3. Jerabkova, L., Kuhlen, T., Wolter, T., Pallua, N.: A Voxel Based Multiresolution Technique for Soft Tissue Deformation. In: Proceedings of ACM Symposium on Virtual Reality Software and Technology (VRST). (2004) 158–161
4. Shewchuk, J.: Tetrahedral Mesh Generation by Delaunay Refinement. In: Symposium on Computational Geometry. (1998) 86–95
5. Renze, K., Oliver, J.: Generalized Unstructured Decimation. IEEE Computational Geometry and Applications **16**(6) (1996) 24–32
6. Paloc, C., Bello, F., Kitney, R., Darzi, A.: Multiresolution Volumetric Mass Spring System for Real Time Soft Tissue Deformation. In: Proceedings of MICCAI. (2002) 219 – 226
7. Faraci, A., Bello, F., Darzi, A.: Soft Tissue Deformation using a Nonlinear Hierarchical Finite Element Model with Real-Time Online Refinement. In: Proceedings of Medicine Meets Virtual Reality. (2005)
8. Paloc, C.: Adaptive Deformable Model (allowing Topological Modifications) for Surgical Simulation. Phd thesis, Imperial College London (2003)
9. Mor, A.: Progressive Cutting with Minimal New Element Creation of Soft Tissue Models for Interactive Surgical Simulation. Phd thesis, Robotics Institute, Carnegie Mellon University (2001)

Author Index