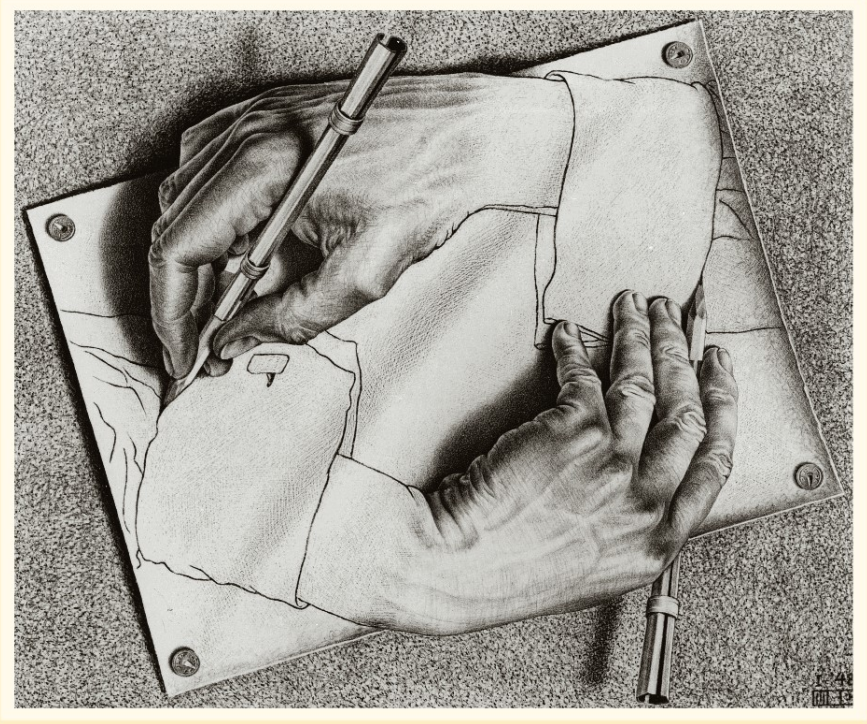
Ambient Intelligence
A Novel Paradigm

Paolo Remagnino
Gian Luca Foresti
Tim Ellis
(Eds.)



Springer

AMBIENT INTELLIGENCE

AMBIENT INTELLIGENCE
A Novel Paradigm

Edited by
PAOLO REMAGNINO
Digital Imaging Research Centre, Kingston University

GIAN LUCA FORESTI
DIMI, Universita' di Udine

TIM ELLIS
Digital Imaging Research Centre, Kingston University

Springer

eBook ISBN: 0-387-22991-4
Print ISBN: 0-387-22990-6

Created in the United States of America

Visit Springer's eBookstore at: http://ebooks.kluweronline.com
and the Springer Global Website Online at: http://www.springeronline.com

To a better and simpler future!

Contents

Preface

If the space around us could adapt to our needs and intentions, then our lives would be much simpler. We would have to spend less time on our daily chores, we would be more productive and, hopefully, we would live in a less worrisome and, most likely, more secure world.

Ideally, it would be great if we could live our lives in environments able to commune with us. In Star Trek, the generic user not only can dialogue with a computer, but can also shape the environment - the *holodeck* - to suit their needs. Undoubtedly we are still far from such a futuristic era, most likely we will not live long to witness a Star Trek world, but we can push forward science and technology. However, in order to progress we can not simply and blindly rely on technology, we must develop interdisciplinary methods encompassing all aspects of our life to develop intelligent solutions, user centric, able to understand us and our lifestyle and activities. We will most likely need to design new methods for learning, adapting to specific moments in time, environmental and personal conditions. We will need to find new ways to combine science, technology, art and, above all, cognitive and psychological studies, to devise complete solutions; not simply working ones, operating to better the space around us.

Current technological advances have made giant leaps forward, but they are still too intrusive and passive. The monitoring of public and private spaces usually impinges on our privacy and in some cases barely adheres to the local laws (some European countries have very strict privacy laws on video data and information, for instance). So, interdisciplinary solutions will have to be adopted to include legal and ethical issues to generate solutions commensurate to all of us, as citizens of a modern and democratic society.

Across the Ocean, back in the 90's, the Americans introduced the concept of smart rooms, the European Community responded a decade later with the paradigm of Ambient Intelligence. Ambient Intelligence (abbreviated AmI), is wider in scope, even though some of the ideas were inspired by the smart room concept. Smart Rooms were introduced to offer a new solution to the human-machine interface problem. AmI goes well beyond this, encouraging intelligent systems where they cease to exist, or gracefully disappears into the

background. AmI promotes pervasive, distributed technology, not intrusive, but always present. A clear analogy is electricity: it is ever-present and widely used, but we do not think about it, and most of the time we are not aware of it. AmI intends to provide design criteria for an intelligent infrastructure; intelligent, not only because it can interpret our actions and intentions, but also because it can, more or less interactively, change our environment to help us with transparent solutions.

The last decades have witnessed major advances in science and technology, sufficient to make part of the AmI dream possible. Digital communication and wireless communication are most likely the technologies that have changed our lives most profoundly. The Internet is the clearest example. Such technologies have indeed helped us but they have also generated new problems. For instance the Internet can be seen as a communing means, but it can also be interpreted as a bundle of technologies making more difficult, if not impossible, natural human interaction. AmI wishes to introduce new ways of communicating between humans and machines, removing hurdles created by interfaces, and introducing more direct and intuitive modes of communication.

Writing a book on AmI, including or even touching all the aspects that the paradigm entails would be impossible. We as editors, wish to give the perspective of Computer Vision practitioners who have been working in the field for at least a decade and who are excited by the prospect of entertaining new horizons, pushing the current state of the art of machine vision and combining commonly used methods with available and fascinating techniques borrowed from other disciplines. Whoever will read this collection might be a little disappointed, because it has a strong technological slant and it relies on some mathematical, statistical and artificial intelligence techniques - computer vision being one of them. To the disappointed we wish to convey that we have started down a new road, and we have to learn how to bridge the gap, but we are neither scared nor worried to explore it and this volume is meant to be only the first step towards a long journey that will help us to reach some important goals.

PAOLO REMAGNINO, GIAN LUCA FORESTI AND TIM ELLIS

Foreword

Ambient Intelligence technology presents us with a huge challenge in the 21st century: it offers the creation of an environment which is responsive to the activities and presence of people. What does this mean and what kind of world does this create? Designing for this type of environment needs a new approach: instead of thinking of technology as a resource inside a 'black box', the meaning of ambient intelligence is that a distributed network presents facilities to people wherever they are in multiple forms which are offered through old, new and hybrid interfaces. This technology therefore presents a number of challenges: - how can we represent the full functionality of a networked responsive system though a local interface? - how can we design the tangible interfaces in the 'real world' to be hybrid representatives of a virtual and physical 'border' condition? - how can we ensure the trustworthiness and security of a system in which people delegate responsibility to a diffused network?

The greatest challenge is therefore not in the technology itself but in the cultural manifestations and functions of the technology. The distribution of computation, communication and information across everyday environments such as the home, the workplace, the car and the shopping centre, means that a population of networked technical devices needs to be designed in such a way that they are useful, meaningful, and dependable for the people who use them. As they become connected to form communities of devices and ecologies of systems, accessing and sharing information, they have to remain understandable and dependable as they scale up and start to represent access to increasing amounts of functionality and content. And so the issue becomes one of complexity: exponential complexity in terms of the way that mediating objects are linked together: at the level of the user, the environment and the system.

The research challenges for Ambient Intelligence are therefore multiple: the programming of the system dynamics, the development of appropriate technological capabilities, the design of culturally relevant systems and the understanding of human aspirations. We need to ensure that Ambient Intelligence research includes design and cultural factors in the innovation process, because the problems and opportunities posed by such a technology are in fact, largely cultural and psychological: we need to know how to customise technology in

such a way that it is useful to many different types of people and not just those whom we can define as the 'technological elite' within western society: its 'lead users'. The ambition for this technology is that it is available in all environments - in which case, it has to be able to 'mould' itself to local conditions. This adaptivity can only be effected when we know how different people think and experience their world, how they want to spend their time, in what ways they may wish to work and why they may want to use this type of technology. In other words, we have to be able to customise from the level of global system to individual user, whatever their age or cultural background, This seemingly impossible requirement must be the starting point for the research on Ambient Intelligence, because otherwise the results will not be intelligent enough. Even worse, the results will be delivered ' preloaded' with inappropriate cultural assumptions and design forms - in which case, they will be useless.

In the European Commission's research programme 'Connected Community' I proposed that 'community as database' be a research theme for the projects which would be launched. The assumption was that human resources are the most intelligent on the planet, and it would therefore make sense to design technological systems to access 'real time' human resources rather than information in a computer database. We can develop Ambient Intelligence solutions that allow us to access the collective intelligence and experience offered as a human resource - whether they are retired teachers who can help a child with homework, or an unemployed person able to help with gardening, or a network of elderly people caring for one another - in all cases, the technology facilitates human intervention and action. This is the route that ensures that technology is designed with people in mind - that it is proposed as a 'tool' rather than a 'solution' and that it facilitates and enables human creativity rather than only automating functions and focussing exclusively on efficiency and productivity.

The worlds that we live in are very far from technological dreams such as 'the paperless office', the 'smart home' and the robot cities of fiction. We live in emotional, cluttered landscapes at home and at work, and we seem to like it this way. We construct environments which support our emotional and spiritual needs and not just our physiological needs. Ambient Intelligence technology must perform usefully within these environments, and be sophisticated enough to recognise and support local cultural values and individual needs at all levels of human aspiration. In my own work I have concentrated on the 'augmentation of the human' rather than the creation of autonomous technology. The ubiquity of the mobile telephone is perhaps the real forefront of distributed computing, and there we see the user as publisher - or 'micropublisher' - of texts and visual material. We can assume that this trend will continue and we will see the role of the user being transformed into one of producer, director, publisher and entrepreneur - facilitated by an Ambient Intelligence environment which promotes this level of empowerment.

I am proposing that for Ambient Intelligence Research to make a difference and effect a paradigm shift in the role of technology in our everyday lives, we need to form interdisciplinary teams of cultural anthropologists, psychologists, designers and research scientists working together to create 'open tools' that enable the individual and the collective to achieve their goals. Otherwise, we will add a new layer of unsophisticated complexity on top of the current 'digital divide' and will deliver technologically and culturally determined solutions to a disappointed and disenfranchised public. Instead, we can take up the challenge that Ambient Intelligence offers us and construct environments which synergise social and technological behaviours and contribute to human wellbeing. The 'palette' of technologies such as computer vision, speech recognition, artificial intelligence, robotics, embedded intelligence, machine learning, and distributed computing embrace a diversity of 'dynamic materials' that we can use to construct ecologies and communities of socio-technical systems. This book is a timely and valuable contribution to this discourse in the wider community of Ambient Intelligence researchers and practitioners.

Professor Irene McAra-McWilliam
Head of Department of Interaction Design
Royal College of Art
London

Chapter 1

AMBIENT INTELLIGENCE

A Gentle Introduction

P.Remagnino[1], H.Hagras[2], N.Monekosso[1], S.Velastin[1]

[1] *Digital Imaging Research Centre (DIRC), Kingston University, UK*
{p.remagnino, n.monekosso,s.velastin}@kingston.ac.uk

[2] *Department of Computer Science, University of Essex, UK*
hani@essex.ac.uk

Keywords: Ambient intelligence, people detection, machine learning, smart environments.

1. Introduction

This introductory chapter describes Ambient Intelligence *(AmI)* from the perspectives of researchers working in the field of Artificial Intelligence and Computer Vision. It is for the reader to get acquainted with some of the ideas that will be explored in greater detail in the following chapters.

Ambient Intelligence is a term that was introduced by the European community (see [9, 12]) to identify a paradigm to equip environments with advanced technology and computing to create an ergonomic space for the occupant user. Here the term ergonomic is used in a broad sense, encompassing both better living environment, secure space, but also an active, almost *living* space around us; capable of aiding us with daily chores and professional duties. Later on in the book the reader will be able to see examples of enhanced homes for the elderly, intelligent buildings, devices built for education and entertainment and conventional visual surveillance systems, easily portable to other domains of application, such as the training of professionals.

The AmI paradigm can be realised only through a number of technologies, all involving modern computing hardware and software. In particular, an AmI system requires the use of distributed sensors and actuators to create a pervasive technological layer, able to interact transparently with a user, either passively by observing and trying to interpret what the user actions and intentions are, but

also actively, by learning the preferences of the user and adapting the system parameters (applied to sensors and actuators, for instance) to improve the quality of life and work of the occupant.

It must be born in mind that the AmI paradigm is not restricted to any type of environment. The idea of an *augmented* space surrounding a user could be an open or a close environment, constrained in a physical location, or spread across a large space. The most important concept is that the pervasive network is able to track the user preferences through space and time, improving the human-machine *relationship* that, in the AmI paradigm becomes very much anthropomorphic.

Section 2 presents the Essex approach to AmI, Section 3 introduces motion detection as a technique commonly used in Computer Vision and Video Surveillance and describes how it could be used as a presence detector. More advanced Computer Vision techniques will be introduced in later chapters, all aimed at identification, classification and tracking of individuals in a more or less cluttered scene. Concluding remarks appear in Section 4.

2. The Essex approach

The Department of Computer Science at Essex University carries out research in the field of AmI. Their approach is focused on the implementation of AmI as indoors smart environments. In particular, state of the art Artificial Intelligence techniques are employed in the implementation of a futuristic *Intelligent Dormitory* (iDorm). The following sections will describe their approach and some of the employed technology. The main idea here is to illustrate a concrete example of AmI put into practice with success. Later on in this chapter the reader will be able to understand how Computer Vision could enhance the iDorm and typical AmI *enabled* smart environments.

2.1 The iDorm - A Testbed for Ubiquitous Computing and Ambient Intelligence

The Essex intelligent Dormitory (iDorm), pictured in Figure 1.1 (left), is a demonstrator and test-bed for Ambient Intelligence and ubiquitous computing environments. Being an intelligent dormitory it is a multi-use space (i.e. contains areas with differing activities such as sleeping, working, entertaining, etc) and can be compared in function to other living or work environments, such as a one-room apartment for elderly or disabled people, an intelligent hotel room or an office. The iDorm contains the normal mix of furniture found in a study/bedroom allowing the user to live comfortably. The furniture (most of which are fitted with embedded sensors that provide data to the network for further processing) includes a bed, a work desk, a bedside cabinet, a wardrobe and a PC-based work and multimedia entertainment system. The PC contains most

office type programs to support work and the entertainment support includes audio entertainment (e.g. playing music CDs and radio stations using Dolby 5.1 surround sound) as well as video services (e.g. television, DVDs, etc). In order to make the iDorm as responsive as possible to the needs of the occupant it is fitted with an array of embedded sensors (e.g. temperature, occupancy, humidity, light level sensors, etc) and effectors (e.g. door actuators, heaters, blinds, etc). Amongst the many interfaces, we have produced a virtual reality system (VRML) shown in Figure 1.1 (right) that marries the Virtual Reality Modeling Language with a Java interface controlling the iDorm. It provides the user with a visualization tool showing the current state of the iDorm and allows direct control of the various effectors in the room. Although the iDorm

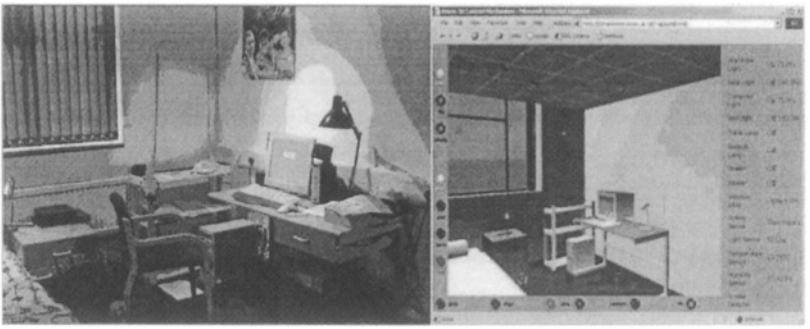


Figure 1.1. left) Photo of iDorm. right) The iDorm VRML interface.

looks like any other room, above the ceiling and behind the walls hide a multitude of different networks and networked embedded devices. In building the iDorm, we have installed devices that reside on several different types of networks. As such access to the devices needs to be managed, gateways between the different networks can be regarded as critical components in such systems, combining appropriate granularity with security. Currently the iDorm is based around three networks: Lonworks, 1-Wire (TINI) and IP, providing the diverse infrastructure present in ubiquitous computing environments and allowing the development of network independent solutions.

Lonworks is Echelon's proprietary network and encompasses a protocol for building automation. There are many commercially available sensors and actuators for this system. The physical network installed in the iDorm is the Lonworks TP/FP10 network. The gateway to the IP network is provided by Echelon's iLON 1000 web server. This allows the states and values of sensors and actuators to be read or altered via a standard web browser using HTML forms. The majority of the sensors and effectors inside the iDorm are connected via a Lonworks network.

The 1-Wire network, developed by Dallas semiconductor was designed for simple devices to be connected over short distances. It offers a wide range of commercial devices including small temperature sensors, weather stations, ID buttons and switches. The 1-Wire network is connected to a Tiny Internet Interface board (TINI board), which runs an embedded web server serving the status of the networked devices using a Java servlet. The servlet collects data from the devices on the network and responds to HTTP requests.

The IP network forms a backbone to interconnect all networks and other devices like the Multi-media PC (MMPC). The MMPC will be the main focus for work and entertainment in the iDorm. Again the MMPC uses the HTTP protocol to display its information as a web page.

The iDorm's gateway server is a practical implementation of a HTTP server acting as a gateway to each of the room's sub networks. This illustrates the concept that by using a hierarchy of gateways it would be possible to create a scalable architecture across such heterogeneous networks in ubiquitous computing environments. The iDorm gateway server allows a standard interface to all of the room's sub networks by exchanging XML formatted queries with the entire principal computing components, which overcomes many of the practical problems of mixing networks. This gateway server will allow the system to operate over any standard network such as EIBus, Bluetooth and Lonworks and could readily be developed to include 'Plug N Play' allowing devices to be automatically discovered and configured using intelligent mechanisms. In addition, it is clear such a gateway is an ideal point to implement security and data mining associated with the sub network. Figure 1.2 shows a logical network infrastructure in the iDorm.

2.2 The iDorm Embedded Computational Artifacts

The iDorm has three types of embedded computational artifacts connected to the network infrastructure. Some of these devices contain agents.

The first type is a physically static computational artifact closely associated with the building. In our case this artifact contains an agent and thus is termed the iDorm Embedded Agent. The iDorm agent receives the iDorm sensor values through the network infrastructure and contains intelligent learning mechanisms to learn the user's behaviour and compute the appropriate control actions and send them to iDorm effectors across the network. The iDorm Embedded Agent is shown in Figure 1.3(left) and is based on 68000 Motorola processor with 4 Mbytes of RAM, an Ethernet network connection and runs VxWorks Real Time Operating System (RTOS). The sensors and actuators available to the iDorm Agent are as follows: The agent accesses eleven environmental parameters (some, such as entertainment, being parameters on multi-function appliances):

- Time of the day measured by a clock connected to the 1 - Wire network

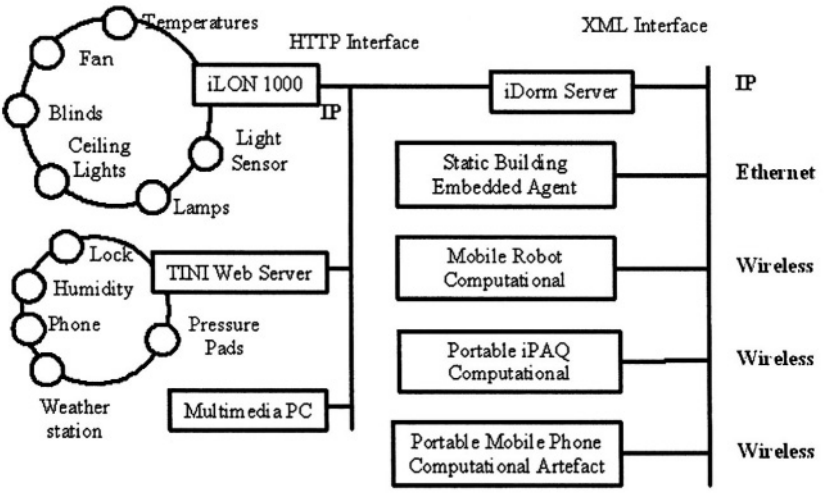


Figure 1.2. The logical network infrastructure in the iDorm.

- Inside room light level measured by indoor light sensor connected to the Lonworks network
- Outside outdoor lighting level measured by an external weather station connected to the 1-Wire network
- Inside room temperature measured by sensors connected to the Lonworks and the 1-Wire networks
- Outside outdoor room temperature measured by external weather station connected to the 1- wire network
- Whether the user is using his audio entertainment on the computer - sensed by custom code publishing the activity on the IP network
- Whether the user is lying or sitting on the bed or not, measured by pressure pads connected to the 1-Wire network
- Whether the user is sitting on the desk chair or not, measured by a pressure pad connected via a low power wireless connection to the 1-Wire network
- Whether the window is opened or closed measured by a reed switch connected to the 1-Wire network

- Whether the user is working or not, sensed by custom code publishing the activity on the IP network
- Whether the user is using video entertainment on the computer - either a TV program (via WinTV) or a DVD using the Winamp program sensed by custom code publishing the activity on the IP network

The agent controls nine effectors, which are attached to the Lonworks network:

- Fan Heater
- Fan Cooler
- A dimmable spot light above the Door
- A dimmable spot light above the Wardrobe
- A dimmable spot light above the Computer
- A dimmable spot light above the Bed
- A Desk Lamp
- A Bedside Lamp
- Automatic blind status (i.e. open/closed and angle)

The room is also equipped with other sensors such as a smoke detector, a humidity sensor, activity sensors and telephone sensor (to sense whether the phone is on or off the hook) as well as a camera to be able to monitor what happens inside the iDorm. It is possible to follow (and control) the activities inside the iDorm, via a live video link over the Internet.

The second type of the embedded computational artifacts is a physically mobile computational artifact. This takes the form of a service robot and contains an agent and thus termed robotic agent. The robotic agent can learn and adapt, online, the robot navigation behaviors (which is different to iDorm embedded agent that seeks to realize ambient intelligence). The robot prototype used in the iDorm is shown in Figure 1.3(middle left). The robot can be regarded as a servant-gadget with the aim of delivering various objects of interest to the user of the iDorm such as food, drink and medicine. The mobile robotic agent has a rich set of sensors (9 ultrasound, 2 bumpers and an IR beacon receiver) and actuators (wheels). It uses 68040 Motorola processors and runs VxWorks Real Time Operating System (RTOS). The robot is equipped with essential behaviors for navigation, such as obstacle avoidance, goal-seeking and edge-following. These behaviors are combined and co-ordinated with a fuzzy coordination module so the robot can reach a desired location whilst avoiding obstacles. The robot's location is passed to and processed as an additional input

by the static iDorm embedded agent that controls the iDorm. In the experimental set up we use a simplified system in which the robot can go to two locations identified by infrared beacons to pick up objects. After picking up an object the robot can deliver it to the user and then go to its charging station, which is identified by another infrared beacon. The robotic agent sends information about its location to the iDorm agent and it takes destination instructions from that agent depending on the user's previous behavior. For example the robot might have learned to go and fetch a newspaper from a specific location whenever it is delivered in the morning.

The communication between the static iDorm embedded agent and the mobile robotic agent is implemented via a wireless link. Communication is established by initiating a request from the iDorm embedded agent to the mobile robotic agent server. Once the request has been sent the server passes it to the robotic agent to carry out the task and informs the iDorm embedded agent of the robot's current status. If the task is in progress or not completely finished then the server sends a message indicating that the job is not complete. Every time the iDorm embedded agent wants to send out a new request, it waits until the previously requested job has been successfully completed.

The third type of the embedded computational artifacts is a physically portable computational artifact. Typically these take the form of wearable technology that can monitor and control the iDorm wirelessly. The handheld iPAQ shown in Figure 1.3(middle right) contains a standard Java process that can access and control the iDorm directly, this forms a type of "remote control" interface that would be particularly suitable to elderly and disabled users. Because the iPAQ supports Bluetooth wireless networking, it was possible to adjust the environment from anywhere inside and nearby outside the room. It is also possible to interact with the iDorm through mobile phones as the iDorm central server can also support the WML language. Figure 1.3(right) shows the mobile phone WAP interface which is a simple extension of the web interface. It is possible for such portable devices to contain agents but this remains one of our longer-term aims.

The learning mechanisms within the iDorm embedded agent are designed to learn behaviors relating to different individuals. In order to achieve this the embedded agent needs to be able to distinguish between users of the environment. This is achieved by using an active lock, designed and built within the University of Essex, based on Dallas Semiconductors 1-Wire protocol. Each user of the environment is given an electronic key, about the size of a penny. This is mounted onto a key fob and contains a unique identification number inside its 2-kilobyte memory. The Unique ID Number of the user is passed to the iDorm embedded agent so that it may retrieve and update previous rules learnt about that user. We have tried various learning mechanisms to realize

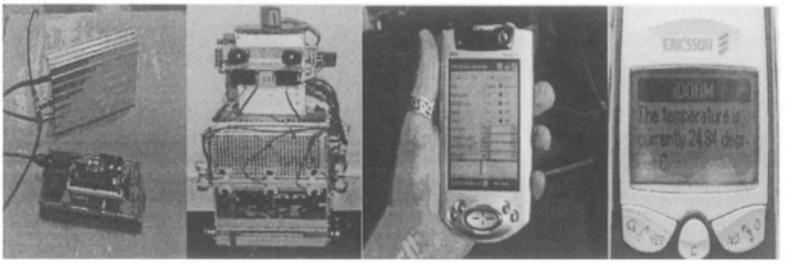


Figure 1.3. The iDorm embedded computational artefacts left) Static iDorm Embedded Agent, middle left) Mobile Service Robot, middle right) Portable iPAQ interface, right) Portable mobile phone interface.

the ambient intelligence vision in ubiquitous computing environments. More details can be found in [6], [4], [5], [2], [3].

3. Integrating Computer Vision

Computer Vision used to exist as one of the many incarnations of Artificial Intelligence [11][15]. At present and for a few decades now, Computer Vision has become an independent research field in its own right. The main goal of Computer Vision research is to interpret images and sequences of images, by learning models of stationary and moving people and objects captured by a more or less large network of sensors. Sensors employed in Computer Vision are 2D cameras. In general a Computer Vision system deploys heterogeneous networks of cameras, including fixed cameras mounted with casing or on tripods, motorized cameras (pan-tilt-zoom, also abbreviated as PTZ) and omni directional cameras, capable of a 360 degree field of view, typically using a semi-hemispherical mirror lens.

Describing Computer Vision is not possible in a Chapter, the interested reader is referred to [13][1] as textbooks on the subject. The aim of this chapter is to illustrate one of the many techniques and describe how it can be integrated in an AmI system. Vision is what we would call a *fragile* technology, because, even though it has been around for quite sometime its results are mainly at academic level and only a small portion of Computer Vision techniques can be deemed sufficiently robust to be marketed and employed in real-life applications robustly and reliably. Typically, what is usually called low level vision - sometimes called image processing - is more robust and it forms the springboard for camera based machine vision applications. The next section illustrates a well known technique used to detect moving objects in the scene. The idea is very simple, a visual process learns the background of the scene - that is whatever is stationary

- and a statistical model is built. Moving objects in the scene - represented by those pixel not conforming to the statistical model - are then extracted and subsequently merged to form $2D$ models of objects and people. Such models are then usually further analyzed, extracting their main characteristics, including chromatic and geometric information.

Computer Vision is essential for an AmI system: it gives the possibility to monitor an environment and report on visual information, which is commonly the most straightforward and human-like way of describing an event, a person, an object, interactions and actions of achieving robustness the scene. What is not straightforward is to make robust a suite of techniques that are intrinsically fragile. One way is by learning from raw data and information, rather than creating precompiled models of ideal objects, people or events. The technique chosen as one of the most representative in Computer Vision, does exactly this; it builds a model of what is stationary in the scene by learning from raw visual data.

3.1 User Detection

An AmI system should be able to detect the presence of a user, and machine vision processes can do this. One could imagine building the expected model of the user given the current field of view of the monitoring camera, but this would most certainly fail to address the simple problem of identifying whether someone has entered their office, they have decided to sit down or they are now at the keyboard typing away a document or coding a piece of software.

Visual detection in this chapter is performed by employing a learning rule on a pixel basis. Consider the intensity of a pixel as a stochastic process, clearly non stationary because of continuous variations in ambient light, either due to natural or artificial lighting conditions. The idea here follows one of the many background modeling techniques where for each pixel a number of modes is learnt (peaks on the pixel probability density function) in terms of a mixture of Gaussian models. In mathematical terms a pixel p_i will be modeled in terms of a set of Normal distributions $N(\mu_i, \sigma_i)$. Both μ and σ can be learnt by using the algorithm in Figure 1.4. The variable β plays the role of the learning coefficient,

At time $t+1$, for pixel p_{t+1}

$$\begin{aligned} \mu_{t+1} &= \mu_t + \beta_t * (\mu_t - p_{t+1}) \\ \sigma^2_{t+1} &= \sigma^2_t + \beta_t * (\sigma^2_t - (\mu_{t+1} - p_{t+1}) * (\mu_{t+1} - p_{t+1})) \end{aligned}$$

where $\beta_{t+1} = \beta_t - \epsilon$

Figure 1.4. Background updating equations.

usually initialized to a value close to 1 and slowly decremented. This means that at the beginning of the learning phase the current pixel values are considered

greatly and their contribution slowly decays as β reaches its lower threshold, set to a small value, typically 0.2, to allow a contribution from newly read pixel values.

A user can then be detected as a compact ensemble of foreground pixels by estimating whether a newly read pixel value statistically falls within acceptable boundaries. The described method is a slight variation of the method proposed by Stauffer [16]. A pixels is classified using the Mahalanobis distance, which is tested for all modes of the pixel distribution $\frac{(\mu_t - p_t)^2}{\sigma^2} < th$.

The following series of frames (Figure 1.4 and 1.5) illustrate two examples of how Computer Vision could be used to detect the presence of a user and maintain it.

Figure 1.5. User reaching a working station: (first row) original, (second row) extracted foreground.

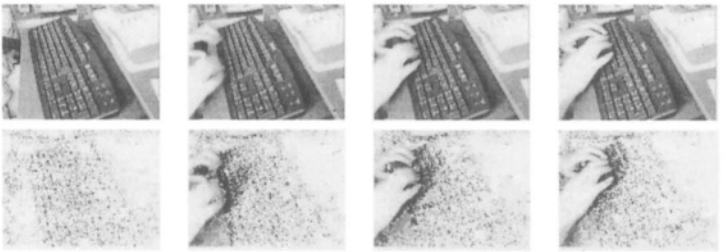


Figure 1.6. User at working station: (first row) original, (second row) extracted foreground.

Both series of frames illustrates the identification phase, which follows a training phase during which the background statistical model is built. This technique has its advantages and disadvantages. The training phase can be relatively short, no more than 100 frames are required to build a stable background model. The main problem is that the faster the background models adapts to changes in the scene, the quicker it does adapt to the presence of a person, if this person stops moving, as they do when sitting at a working station or reading a book. There are therefore two basic solutions to this problem: on one hand the

visual processes could keep a log of a person having entered the environment, for instance, and then log out the person when she moves out again (detected by the motion process). On the other hand one can think of not being bothered of whether the person actually stands still and accept that he or she will in fact not move at times, counting on the fact that every once in a while the person will indeed move and be detected again. This is what happens when a person is at the keyboard.

3.2 Estimating reliability of detection

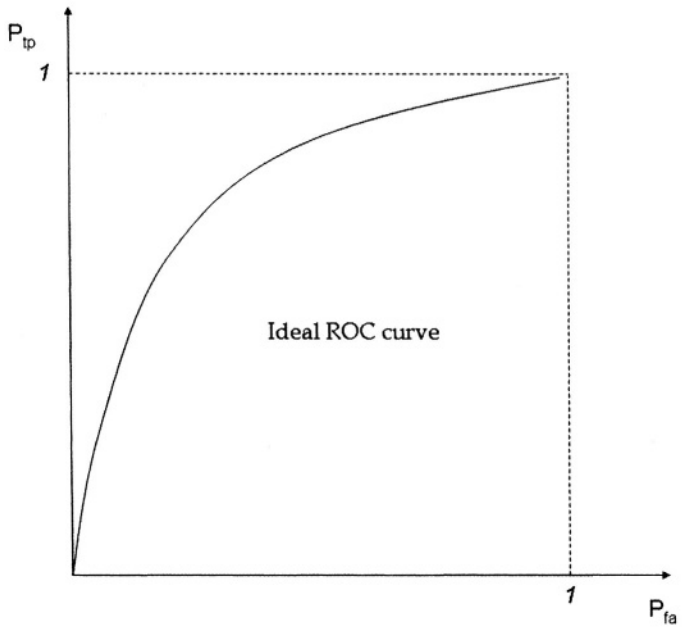


Figure 1.7. An ideal ROC curve.

In order to estimate the reliability of the presence detection process a series of experiments were run. A webcam was fixed looking down a computer keyboard, acquiring video at frame rate; a visual process was learning the model of the background and then extracting foreground pixels, used to estimate the presence. Another process, in parallel, was storing information on the keystrokes of the user. Both processes being time-stamped provided a way of comparing ground truth data - the keyboard strokes - and the estimated presence - the foreground detection. The reliability of the visual process can

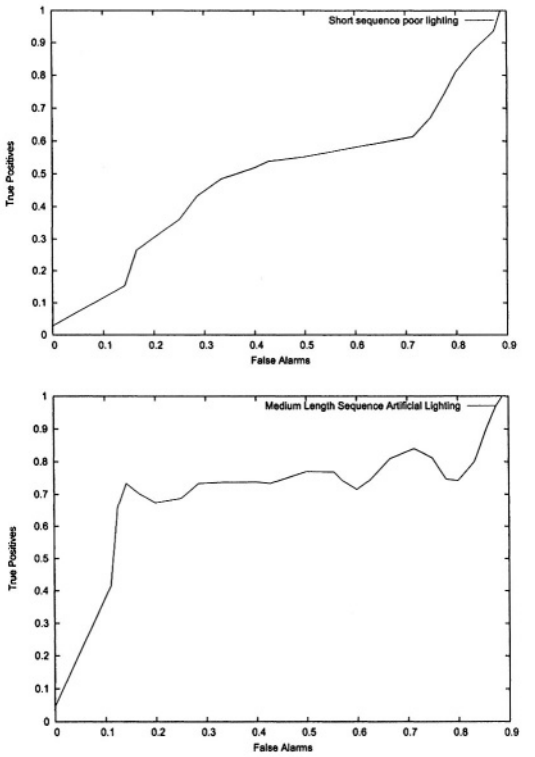


Figure 1.8. ROC curves: (top) Short experiment with poor lighting, (bottom) Medium length experiment with artificial lighting.

be demonstrated using the *receiver operating characteristic* (ROC) [14]. A ROC curve plots couples of points (P_{fa}, P_{tp}), representing respectively the probability of false alarms and the probability of true positives. ROC curves have been widely used in statistics and more recently in Computer Vision, to illustrate the robustness of Vision algorithms [10]. Figure 1.7 shows an ideal ROC curve. In essence, the probability of true positives must always be high. A pair (P_{fa}, P_{tp}) is estimated by counting, over a time window the number of *true positives* (N_{tp}), *true negatives* (N_{tn}), *false positives* (N_{fp}) and *false negatives* (N_{fn}), using the following two formulas:

$$P_{fa} = 1 - \frac{N_{tn}}{N_{tn} + N_{fp}} \qquad P_{tp} = \frac{N_{tp}}{N_{tp} + N_{fn}} \tag{1.1}$$

The two graphs of Figure 1.8 illustrate the ROC curves for two experiments. Figure 1.8(top) shows the result of a short experiment run in the evening with

poor lighting conditions, while Figure 1.8(bottom) shows the result of a medium sequence run with artificial lighting. By looking at the two graphs one can easily notice that the longer the experiment, the more robust the system becomes.

3.3 Vision in the iDorm

Computer Vision can be easily integrated in the iDorm. A cluster of cameras could be installed in the environment, monitoring the activity of the occupant. A part from making use of the described presence detection algorithm, it is possible to create a temporal model of the scene, keeping track of where the person has been and when storing spatial and temporal information as a Markov chain as shown in [17][8][7] and also described in Chapters 7 and 6 of the book.

4. Conclusions

This chapter has introduced AmI as a novel paradigm able to create new synergies between human and machine. An AmI system is meant to work transparently, to proactively aid the user. All chapters in this collection illustrate examples of techniques that could be employed in a fully integrated AmI environment. This Chapter has also introduced Computer Vision as an essential component for an AmI system. The specific technique of motion detection has been described as one of the most popular and useful in Computer Vision and, most likely, one of the most important for an AmI compliant system.

References

[1] D. A.Forsyth and J.Ponce. *Computer Vision A Modern Approach.* Prentice Hall, 2003.

[2] F. Doctor, H. Hagras, and V. Callaghan. A type-2 fuzzy embedded agent for ubiquitous computing environments. In *Proceedings of the IEEE International Conference on Fuzzy Systems,* 2004.

[3] F. Doctor, H.Hagras, and V. Callaghan. An intelligent fuzzy agent approach for realising ambient intelligence in intelligent inhabited environments. *IEEE Transactions on Systems, Man and Cybernetics, Part A,* 2005.

[4] H. Hagras, V. Callaghan, M. Colley, G. Clarke, and H. Duman. *Fusion of Soft Computing and Hard Computing for Autonomous Robotic Systems,* chapter Online Learning and Adaptation for Intelligent Embedded Agents Operating in Domestic Environments book, pages 293–323. Studies in Fuzziness and Soft Computing Series , Physica-Verlag, 2002.

[5] H. Hagras, V. Callaghan, M. Colley, G. Clarke, A. Pounds-Cornish, and H. Duman. Creating an ambient-intelligence environment using embedded agents. *IEEE Intelligent Systems,* 2004.

[6] H. Hagras, M. Colley, V. Callaghan, G. Clarke, H. Duman, and A. Holmes. A fuzzy incremental synchronous learning technique for embedded agents learning and control in intelligent inhabited environments. In *Proceedings of the IEEE International Conference on Fuzzy systems,* pages 139–145, 2002.

[7] M.Brand. An entropic estimator for structure discovery. Technical report, MERL - A Mitsubishi Electric Research Laboratory, 1998.

[8] M.Brand, N.Oliver, and A.Pentland. Coupled hidden markov models for complex action recognition. In *Proceedings of Computer Vision and Pattern Recognition conference,* pages 994–999, 1997.

[9] N.Shadbolt. Ambient intelligence. *IEEE Intelligent Systems,* pages 2–3, 2003.

[10] F. Oberti, A. Teschioni, and C. S. Regazzoni. Roc curves for performance evaluation of video sequences processing systems for surveillance applications. In *Proceedings of the International on Image Processing,* pages 949–953, 1999.

[11] P.Winston. *Artificial Intelligence.* Addison-Wesley, 1992.

[12] G. Riva, P. Loreti, M. Lunghi, F. Vatalaro, and F. Davide. *Being There: Concepts, effects and measurement of user presence in synthetic environments,* chapter Presence 2010: The Emergence of Ambient Intelligence, pages 59–82. IOS Press, 2003.

[13] R.Jain, R.Kasturi, and Brian G. Schunck. *Machine Vision.* McGraw-Hill, 1995.

[14] R.O.Duda, P.E.Hart, and D.G.Stork. *Pattern Classification.* Wiley, 2001.

[15] S.Russell and P.Norvig. *Artificial Intelligence A Modern Approach.* Prentice Hall, 2003.

[16] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. In *Proceedings of the Conference on Computer Vision and Pattern Recognition,* pages 246–252, 1999.

[17] V.Kettnaker and M.Brand. Minimum-entropy models of scene activity. In *Proceedings of Computer Vision and Pattern Recognition conference,* pages 281–286, 1999.

Chapter 2

TOWARDS AMBIENT INTELLIGENCE FOR THE DOMESTIC CARE OF THE ELDERLY

S.Bahadori,[1] A.Cesta,[2] L.Iocchi,[1] G.R.Leone,[1] D.Nardi,[1] F.Pecora,[2] R.Rasconi[2] and L.Scozzafava[1]

[1]*Dipartimento di Informatica e Sistemistica, University of Rome, La Sapienza, Italy*
{bahadori, iocchi, leone, nardi, scozzafava}@dis.uniroma1.it

[2]*Planning and Scheduling Team, Italian National Research Council, Italy*
{a.cesta, fpecora, rasconi}@istc.cnr.it

Keywords: Artificial Intelligence, Ambient Intelligence, intelligent sensors, situation monitoring, assistance, e-services.

1. Introduction

Today, modern societies are tackling an important problem, namely the progressive aging of the population. It is often necessary for elderly people to leave their homes to go live with their relatives or become guests of a health-care institution. This phenomenon has serious social and economic consequences. The long term goal of the research developed by the ROBOCARE project (see http://robocare.istc.cnr.it for details) is to contribute to raising the quality of life of elderly persons. In particular, we are pursuing the idea of developing support technology which can play a role in allowing vulnerable elderly people to lead an independent lifestyle in their own homes.

Research in Ambient Intelligence has begun to address some of the issues related to this problem. Nonetheless, most systems developed in this context are strongly sensor-centric, meaning that they are capable of recognizing simple emergency situations and firing alarms consequently. In this paper we aim at augmenting the scope of intervention of the overall system, by endowing it with more complex schemes of intervention. In particular, we are exploring the possible applications of intelligent systems in the domestic environment, studying how AI can be employed to realize cognitively-enhanced embedded

technology which is capable of *supporting* and *monitoring* the elderly person in his or her daily tasks. This implies the presence not only of "classical" sensory capabilities, rather it requires *integrating* such features with complex symbolic reasoning functionalities in a highly connected infrastructure.

In this context, our efforts are striving to integrate state-of-the-art technology in the fields of robotics, sensors, planning & scheduling and multi-agent systems in order to produce one, multi-functional entity to be deployed in a model of a domestic environment. The system is composed of a multitude of agents, each of which provides a set of *services.* The high level coordination of these agents is induced by an Active Supervision Framework (ASF), whose role is to provide effective mechanisms to invoke the agents' services. In this article, we show the two most significative components of the framework, which constitute the basic ingredients of an illustrative system, exposing the basic functionalities which are required for the entire, more complex, framework. First, we describe the *people localization and tracking agent,* a stereo camera which provides the 'still-image' and 'streaming video' services, by means of which it is capable of determining the 3D position of a person in the environment and tracking the person's position when he or she moves. Second, the *execution monitoring agent,* a CSP-based scheduling component which follows the execution of a predefined set of daily activities of the assisted elderly person, providing consistency checking services by reacting to contingencies and foreseeing inconsistencies in the schedule.

These two ingredients constitute an initial prototypical system which is currently deployed in a testbed domestic environment. The integration of these two components is achieved by means of *e-service oriented middleware.* This infrastructure allows all the agents in the system to provide their functionalities in the form of services. Under this paradigm, each agent provides a service which other agents can request, thus implementing basic mechanisms for data exchange and cooperation.

This paper is organized as follows: after describing in greater detail the desiderata for the integrated system, section 2 briefly outlines some basic implementation choices of the e-service infrastructure. Section 3 then illustrates the people localization and tracking service, detailing the basic functioning mechanisms of the stereo-camera based sensor which is currently tested in the domestic environment. Section 4 is dedicated to the description of the execution monitoring service, which is based on a CSP representation of activity work-flow. Following the description of the two basic components, section 5 shows by means of an example how the two components work together in a real-world instance, namely the ROBOCARE Domestic Environment. Finally, the last section gives some conclusions and outlines future work.

2. An Integrated Supervision System

The intelligent system developed for the ROBOCARE domestic environment is composed of a multiplicity of hardware and software agents. All agents can

be thought of as components of a single complex system, whose aim is to create an intelligent and supporting environment. In particular, the target users of this application are elderly people whose every-day independence may be improved by such a monitoring infrastructure.

Currently, the embedded technology in the ROBOCARE environment provides monitoring-specific services, such as the 'still-image' and 'people tracking' services, provided by a fixed camera, the 'find object' service provided by a mobile robot, and the 'visualize' service exported by a Personal Data Assistant. Every agent is present in the system as a set of services, which the other agents can reserve and use. In this light, it is possible to abstract away the physical components, focusing on varying levels of detail by defining 'super-agents' as a collection of services provided by various physically different agents. For instance, the 'remote-visualization' service is implemented by the combination of the camera agent and the PDA. If one of the two devices is not available (i.e. they do not make their service available), then also the super-agent cannot offer its service, and the request must be put on hold. In this schema, when there are redundant services (for instance, a 'still-image' service exported by two mobile robots), the super-agent which exports the combined service is not bound to a particular physical device, rather it is embodied by the first available robot on a preference basis.

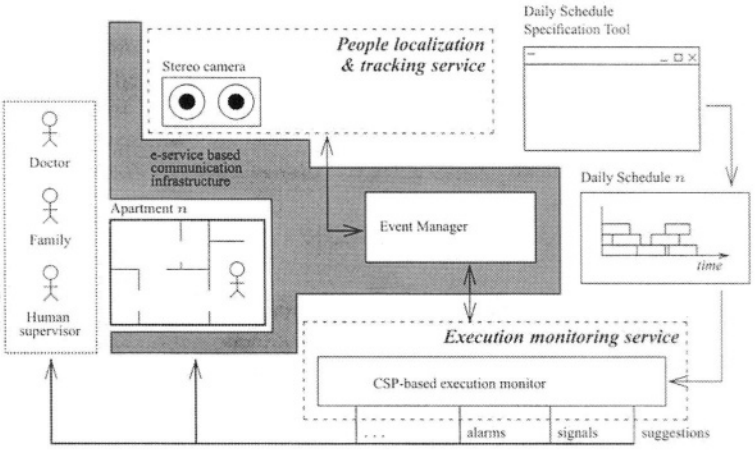


Figure 2.1. A schema of the current implementation of the *Active Supervision Framework.* By integrating the services provided by a stereo-camera and a CSP-based execution monitor, the ASF is capable of monitoring the daily activities of an assisted elderly person in his or her domestic environment.

Fig. 2.1 shows a system in which a stereo-camera is "coupled" with an execution monitoring module. Together, these two components provide basic

services for monitoring the daily schedule of an assisted elderly person in his or her apartment. On one hand, the stereo-camera observes the environment, and by means of its 3D localization capabilities, compiles information on the *current state* of the environment. This information can be processed by the CSP based execution monitoring module. By means of an internal representation of the assisted person's *nominal* schedule, this module attempts to recognize inconsistencies in the actual execution of the activities as it is performed by the assisted person. The loop is closed through the output of the execution monitoring service, which can deliver suggestions, activate alarms and other such signals. In the rest of this paper, we detail the basic functioning mechanisms of the two components.

It is important to notice at this point that one key issue in the implementation of this architecture is what we have called in fig. 2.1 the "event manager". The role of this component is to *arbitrate* the service requests of the various agents. As an arbiter, it is the responsibility of the event manager to orchestrate the flow of information to and from the services, performing load balancing tasks and maintaining global system consistency. The event manager processes all service requests, dispatching them to the appropriate agent, performing synchronization tasks and administering the use of the communication infrastructure which connects all the components of the system. In conclusion, the event manager is what allows the service-providing agents to function together in a highly cooperative fashion.

As a consequence of viewing each component of the system as an agent which provides and can request services, the communication infrastructure has been implemented following the *e-service* paradigm. We will briefly describe this infrastructure in the following subsection. We conclude this section by saying that the integrated system shown in fig. 2.1 constitutes a prototypical implementation of what we call an *Active Supervision Framework* (ASF). The ASF can be seen as a complex infrastructure which provides *monitoring capabilities* in the domestic environment of an assisted elderly person. It provides a coordination framework for the components of the system by implementing service publishing functionalities and coordinating the service-providing agents (by means of the event manager). For instance, if an assisted elderly person decides to perform an activity which contrasts with a particular medication requirement (e.g., no eating for two hours after having taken a particular medication), the system should recognize the inconsistency (by employing the services provided by the agents) and deduce a contingent plan to solve the situation (suggestions, alarms, and so on). Our effort to implement a complete ASF for a domestic environment starts here, with the integration of the two basic services we have mentioned above.

2.1 E-service Based Integration Schemata

Service oriented computing is a relatively new approach to the development of integrated systems. Service oriented architectures (SOA) have evolved over the last years to support high performance, scalability, reliability and availability. In basic terms a service is an application that can be accessed through a programmable interface. Different distributed computing protocols such as CORBA, DCOM and RMI have been adopted to access these services. Although they are very effective for building a specific application, the tight coupling used in these architectures limits the re-usability of individual components. Each of these protocols is constrained by vendor implementation, platform and development languages that severely limit interoperability. Moreover, none of these technologies operate over the Internet with the simple use of the HTTP protocol. This is a real obstacle due to security issues.

E-services represent the convergence between SOA and the web (hence they are often referred to as web-services). They are applications which offer particular functionalities in the form of services. Their functionalities are easily reusable (without knowing how the service is implemented) and they are accessed via the Internet protocol (HTTP) using universally accepted data formats (XML). These standard elements ensure easy integration of heterogeneous components. The e-service interface adds a layer of abstraction that makes the connection flexible and adaptable. Thus e-services have emerged as a powerful mechanism for integrating different systems and assets. This entails an obvious advantage for the ROBOCARE project, which is composed of many units which develop components using their own proprietary software tools. By adopting the e-service infrastructure, the developers of individual units have the only burden of complying with the e-service interface specifications.

There is also a long-term advantage that arises from the present direction of the market. The Internet is is not only a network of computers; there are many electronic devices (cell phones, web-TV, PDAs, etc.) that are capable of surfing the web. In the near future more and more domestic components (washing machines, fridges, ovens, etc.) will offer the possibility of being controlled via remote web access. Probably, they will publish their e-services to interact with other applications. This is very important in our case because these devices are part of the daily routine of a person. The possibility of having an open system which can easily integrate a new component built from a third party is a great challenge.

Although a great deal of work has been done by major IT companies as well as by the organizations concerned with standardization (W3C and OASIS), e-service technologies are still in their initial phase of development. Today we have standards for Data Formats (XML), Service Interaction (SOAP), Interface Description (WSDL) and Publishing and Retrieving (UDDI). In the context of this article it is not our goal to detail the technicalities of the implementation, rather we will describe the interface of the e-services and the dynamics of the

```
type class DeviceData
  id : integer
  type : integer
  name : string
end

type class AddressData
  IP : string
  port : integer
end

type class ImageData
  width : integer
  height : integer
  buf : array[MAXINT] of integer
end

eservice Stereo_Camera
  begin property
    identification : DeviceData
  end property
  begin message
    IN FrameRequest(info : AddressData; to : DeviceData)
    OUT SendFrame(frame : ImageData)
    IN StopRequest()
  end message
  begin state machine
    STATE_NUMBER := 3
    TRANSITION initial TO 1 : FrameRequest/SendFrame
    TRANSITION 1 TO 1 : FrameRequest/SendFrame
    TRANSITION 1 TO final : StopRequest/ϵ
  end state machine
end
```

Figure 2.2. The Stereo Camera e-service formal specification.

communication between services conceptually. In the ROBOCARE domestic environment, we use the PARIDE (Process-based frAmework for oRchestration of Dynamic E-services) framework [10], which works according to a higher meta-level with respect to the XML-based languages. The mapping on technological models will be addressed at later steps in the development process.

The aim of PARIDE is to define a conceptual model of e-services which is independent of specific technologies. It is referred to as *e-service schema.* The main feature of this conceptual model is that it allows to specify not only an e-service's static interfaces, but also its behavior and evolution over time. For this purpose a state machine representation is used. Every change in the state machine is triggered by an incoming message and optionally generates an output. Specifically, an e-service schema consists of two parts: the first describes its interfaces, and the latter describes the conversations of the e-service, that is, possible interactions the e-service can be involved in. In order to show how this works, we describe the e-service of a Stereo Camera Agent so that it exports the 'still-image' service. The complete conceptual schema of this e-service is detailed in fig. 2.2. The very first part is the definition of the needed data structures (type classes). The name of the e-service (Stereo_Camera) identifies the "type" of service, while the property field describes the characteristics which are associated to the service. Thus, if there are different agents which provide the same service, they will all have the same name and different property fields. The messages resemble the public methods of a Class in Object-Oriented Programming languages: they are the only way an e-service can interact with the world.

3. People and Robot Localization and Tracking System

The component we are describing in this section has been developed in order to provide information to the overall system regarding pose, trajectories, and detection of special events about persons and robots acting in the experimental domestic environment. We call this component People and Robot Localization and Tracking Agent (LTA). This module exports a number of services, that may also be customized according to the need of other modules. A set of services exported by the module is: "photo, whereis, whichpose, howlong_in_a_pose, howlong_here, howmany_persons, howmany_robots, robot_close_to_a_person, what_activity". This information is very useful for other agents in order to assess the situation in the environment and to take decisions about changes that are required (see for example the Execution Monitoring Agent described in section 4). Therefore, the LTA is a primary component that has the objective of recognizing interesting situations and reporting them to other decision-making agents.

The basic technique involved is stereo vision in order to detect and track persons and robots in the environment. Moreover, from the extraction of a set of features regarding tracked objects (like eccentricity, height, size, etc.) it is

possible to recognize specific situations and poses (like for example if a person lays down on the ground, or if she is sitting on a table).

The general form of this problem presents many difficulties: first of all, object tracking is difficult when many similar objects move in the same space; second, object recognition is generally difficult when the environment and the agents cannot be adequately structured; third, when moving observers are used for monitoring large environments the problem becomes harder since it is necessary to take into account also the noise introduced by the motion of the observer.

With respect to the above difficulties we present a solution that makes the following choices: i) we limit the number of moving objects in the environment to be at most two or three persons plus any number of robots, but in such a way to exclude very crowded environments; ii) persons are not marked in any way, while robots can be visually recognized by placing on top of them special markers; iii) we make use of a fixed stereo vision system that is able to monitor only a portion of the area in the domestic environment.

The goal of the present work is thus to elaborate the information computed by a stereo vision system placed in a fixed position in the environment, in order to determine the position and the trajectories of moving objects. Since the stereo camera is fixed in the environment, we exploit the knowledge of the background in order to identify other moving objects (persons and robots). Persons are recognized by a model matching between the extracted information and some predefined values (e.g. eccentricity, height, etc.), while robots are recognized through their markers.

In the literature there are several works on people localization and tracking through a vision system, aiming at realizing systems for different applications. These systems are based on the use of a single camera (see [13] for example), stereo vision [6, 1] or multiple cameras [15].

The systems based on a single camera are used for modeling and reconstructing a 2D representation of human beings (e.g. heads and hands) and mainly used for virtual reality and gesture recognition applications. In the ROBO-CARE project we are interested in determining the 3D position of a person in the environment and to track his/her trajectory while he/she moves. To this end stereo vision or multiple cameras can be effectively used in order to compute the 3D position of objects in the scene. In this way, the system is more efficient in object recognition and tracking and also provides the coordinates of such objects in the environment. For example, the stereo vision system described in [1] focuses on the development of a real-time system for person detection and tracking by using a model matching technique. Also the work in [6] uses information about stereo vision for identifying and tracking persons in an environment; person recognition is here implemented by using a set of patterns to be matched.

The prototype implementation of our system has the objective of providing for real-time processing of stereo images and for identifying people and robot

trajectories in the environment. Preliminary experiments on the system show the effectiveness of the approach, a sufficient accuracy for many tasks, and good computational performance.

3.1 System architecture and implementation

The system we are developing is composed of hardware and software components. Hardware sensor is a pair of Firewire webcams which constitute the Stereo Camera. They need accurate setting in order to work properly. The images coming from this sensor are managed by the Stereo Camera Agent. This software architecture is based on three main modules (fig. 2.3): *foreground/background segmentation,* that is able to distinguish pixels in the image that are coming from the background from the ones that represent the foreground (i.e. persons or robots moving in the environment); *Stereo computation and 3D segmentation,* that evaluates disparities only for the foreground in the two images and computes the 3D position of a set of points; these 3D points are computed by the stereo algorithm and are clustered in continuous regions; *Object identification and tracking,* that associates each 3D cluster of points to a specific robot (when its special marker is recognized) or to a generic person (if it is compatible with some predefined values). The use of a Kalman Filter makes the estimated position of objects in the space more reliable.

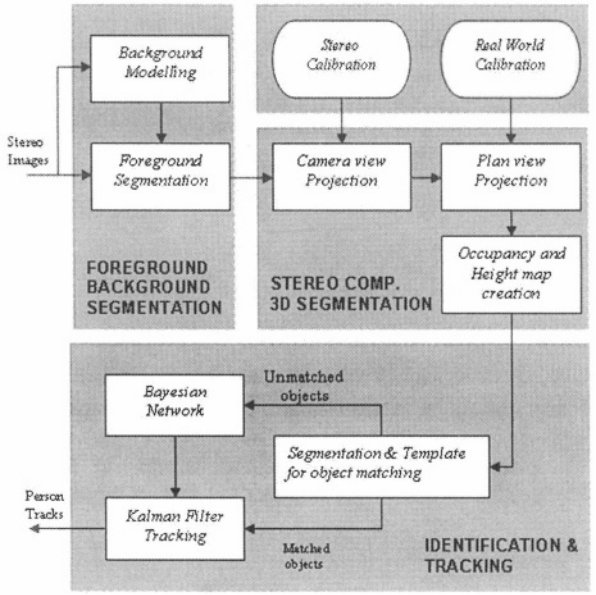


Figure 2.3. The software architecture is based on three main modules

The cameras of the stereo system are placed in a parallel with 18 cm of distance between them. They are installed at 220 cm of altitude from the ground and they point down with an angle of 20° with respect to the horizon line. With this configuration, considering the 62° field of view for each camera, the system is able to cover an area of 2.2 × 1.8 meters. The calibration of the camera is performed by fitting a model of a known *calibration object* to a number of images taken by the cameras. It can be recognized and measured by an automated calibration procedure (for more details, see the Small Vision System [9]).

Foreground/background segmentation. This task is obtained by computing the image difference between a stored image representing the background and the current image. In order to deal with the fact that the background can change over time (for example pieces of furniture or objects on a table can be moved), a special routine is used, which is able to detect these changes after processing a number of frames, updating the background model dynamically. Approaches for background updating [5, 8] usually maintain a model of the background that is updated when parameters of the actual frame are different from the model. In this way, however, persons that remain in the same position for a sufficiently long interval of time may be erroneously integrated in the background model. Since this situation is very common in a domestic environment (e.g. a person sitting at the table), we have devised a new method for background modeling and updating, that is able to distinguish animated objects from inanimated ones and to update the background models only for inanimated objects. In this way the system is both able to avoid to insert persons in the background model and to quickly integrate other objects in it. The method we present is based on the observation that animated objects (e.g. persons) change their shape even if they are still in a place. Therefore an analysis of the edges of an image is sufficient to identify the regions of the image in which there are (even very small) motions from the others. More specifically, we compute the horizontal and vertical edges of an image by means of a Sobel edge extractor operator $H_t(x, y)$ and $V_t(x, y)$ and then compute the *activity* of a pixel in the image as

$$A_t(x, y) = \beta \Delta E_t + (1 - \beta) A_{t-1}$$

where β is a factor that weights new measures with respect to previous ones, ΔE_t is the difference between the edges computed at time $t - 1$ with those computed at time t and can be computed for example as

$$\sqrt{((H_t(x, y) - H_{t-1}(x, y))^2 + (V_t(x, y) - V_{t-1}(x, y))^2)}$$

In our system, we thus update the background model only if the activity $A_t(x, y)$ is greater than a threshold. In this way, for example, if a person sits at a table and at the same time places a bottle on it, after a while the bottle will be integrated in the background (since its edges have low activity), while the person will not.

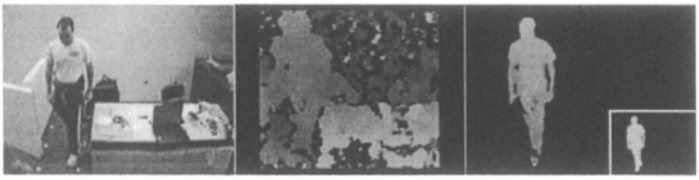


Figure 2.4. (a) original image (b) disparity (c) filtered disparity. Brighter points represent closer objects

Stereo computation and 3D segmentation. The stereo vision system we are using provides a real-time implementation of a correlation-based stereo algorithm. Disparity images represent points that are closer to the cameras with brighter pixels (see fig. 2.4b). Disparity images are computed only for foreground objects and are filtered in order to remove typical stereo noise due to false matches and little texture (fig. 2.4c) and then, through a geometric triangulation, we retrieve the 3D coordinates of all the pixels matched in the two images; this set of points is the 3D model of the scene seen through the cameras. 3D points are transformed in real world reference system through the external calibration [12] performed beforehand. The segmentation phase that follows stereo computation is in charge of clustering the set of 3D points in clusters, each representing a single object in the environment. From the set of 3D points computed, we build two maps that will be used for object identification and tracking: an *occupancy map* (fig. 2.5b) and a *height map* (fig. 2.5c). The former represents the projection of 3D points on the ground, while the latter marks every ground point in the space with the maximum height of the object occupying this point. The filtered height map is computed by applying a filter on the 3D points, which allows to discard points that do not belong to the object (fig. 2.5d).

Object identification and tracking. Each cluster of 3D points is associated to the representation of a moving object in the environment (either a person or a robot) that is tracked by the system. This step takes into account a simple model of a person that is computed from the occupancy and the height maps described above, and the recognition of robot-markers. Those clusters that do not match any of the specifications are simply discarded.

In order to track persons over time in presence of occlusions or when they exit and re-enter the scene, we associate three different templates for every identified object: T_C is a color-based template, T_H is a template based on the height map, and T_O is a template based on the occupancy map. Templates T_H and T_O are directly extracted from the respective height and occupancy maps. The template T_C is a 2-dimensional histogram that takes into account colors

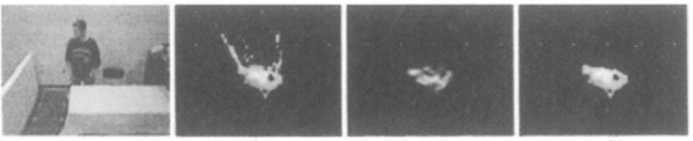


Figure 2.5. (a) original image (b) occupancy map (c) height map (d) filtered height map. Brighter points represent higher objects

associated to every object. It is represented as a $m \times m$ (we use m= 16) matrix TC which discretizes the 2-dimensional color space given by $rg = r - g$ and $by = \frac{2b-r-g}{2}$. Thus, $TC(rg, by)$ is the number of pixels in the image whose color (r,g,b) satisfy the above equations. Object recognition over time is thus performed by computing similarities of the three templates computed for every object. Moreover, in order to effectively track objects in the scene, we use a Kalman Filter on the 2D ground position of the object (i.e. on the ground projection of its center of mass) in order to filter occasional errors in stereo computation, 3D localization, and object identification. Finally, a Bayesian Network is used in order to determine situations that may arise when persons leave or enter the scene.

Although there are some situations in which the above method return erroneous results, especially when several objects are in the environment, a preliminary set of experimental results that are described in the next section shows the feasibility of the approach.

Experimental evaluation. A set of experiments to evaluate accuracy and computational efficiency were performed.

For measuring the *accuracy* of the system we used 9 markers with different distances and angles in scenario. The result after 40 experiments is shown in Table 2.1. These experiments have been carried out using the standard external calibration system.

The performance of *computational efficiency* without the grabbing using a 700 Mhz processor with the presence of one object to track is about 130 ms, by running the grabbing and filtering procedures the calculation time increase to 400ms with one object, 500 ms with two, and 650 ms for three moving objects in the scenario. The major part of the calculation time is dedicated to gaussian filtering and noise detection of the background.

4. The Plan Execution Monitoring System

So far we have shown the sensing capabilities we have deployed in the domestic environment. In particular, the stereo vision based people localization and tracking service is capable of compiling symbolic information (such as the

Marker	Distance	Angle	Err.medium	Err.Variance
M1	3.00 m	00°	7.3cm	6-8.2 cm
M2	2.70 m	00°	7.1cm	6-7.7 cm
M3	2.40 m	00°	6.8cm	5.8-6.7cm
M4	3.00 m	32°	9.8cm	7.8-10.7cm
M5	2.70 m	35°	9.6cm	7.8-9.7cm
M6	2.40 m	37°	8.8cm	7.8-9.7cm
M7	3.00 m	-32°	9.2cm	7.8-10.5cm
M8	2.70 m	-35°	9.6cm	7.8-9.7cm
M9	2.40 m	-37°	8.8cm	7.8-9.7cm

Table 2.1. Accuracy results.

presence, position and permanence of persons and/or objects) from the environment. This information defines, to a certain extent, the current state of the environment. The goal of this section is to describe an *execution monitoring system,* whose task is to follow the evolution of such states and to guarantee that these evolutions match with a set of predefined requirements. These requirements are expressed by means of a set of *activities* whose execution must generally occur within the scope of a complex set of *constraints.* These sets of activities are called *schedules.*

More specifically, a schedule consists of a certain number of activities, each of a predetermined duration and requiring some resources in order to be executed. One key point is the fact that activities can in general be temporally constrained, either individually or among one another: for instance, some operation in a schedule might be constrained not to start before, or not to finish after, a certain instant; in addition, there might be several *precedence constraints* between any two activities in the schedule: for instance, activity B might not be allowed to start before the end of activity A, and so forth.

The problem we describe in this section concerns how to manage a predefined schedule while it is being executed in a real world environment, namely the ROBOCARE Domestic Environment. We want the execution monitor to ensure schedule feasibility, taking into account the *real* status of the execution as it is reported by the people localization and tracking service.

The issue of schedule *consistency* has two aspects: on one hand, the totality of the schedule's temporal constraints, i.e. *release time* constraints, *deadline* constraints, *precedence* constraints as well as others, should be kept satisfied at all times, as they constitute an integral part of the schedule specifications. A schedule where all the temporal constraints are satisfied, is said to be *temporally consistent;* on the other hand, *resource consistency* must be constantly maintained as well, since it is obviously not possible to perform operations when the necessary resources are not available.

One of the major difficulties arising when working in real environments consists in counteracting the effects that the possible unexpected events may

have on the schedule *in a timely manner.* Indeed, when schedule adjustments must be performed, we are interested in detecting whether or not the contingency has caused an inconsistency in the daily schedule of the assisted person.

Research in scheduling and execution monitoring has produced two approaches to this problem, namely the predictive approach and the reactive approach [7]. The execution monitoring system we describe herein is based on the latter. In slightly simplified terms, we can say that according to the reactive approach, the execution monitoring system attempts to maintain consistency by manipulating the schedule every time it is deemed necessary. The current state of execution of the assisted person's tasks is provided by the environmental sensory services invoked by the execution monitor. Thus, the monitoring system follows the actual execution of activities and can properly react to the occurrence of unexpected events.

The task of the system is therefore twofold: on one hand, it is to represent the possible damages on the schedule, fire the proper repair action and continuously guarantee its executability; on the other hand, it is to ensure that all the scheduled activities are in fact positively executed, issuing warnings and/or alarms in the opposite case. To this aim, an *Execution Monitor* has been developed, which exhibits reactive behavior and conveniently re-adjusts the schedule's activities by means of the *ISES* procedure (Iterative Sampling Earliest Solutions) [3], a constraint-based method originally designed to solve the RCPSP/max problem (Resource Constrained Project Scheduling Problem with Time Windows).

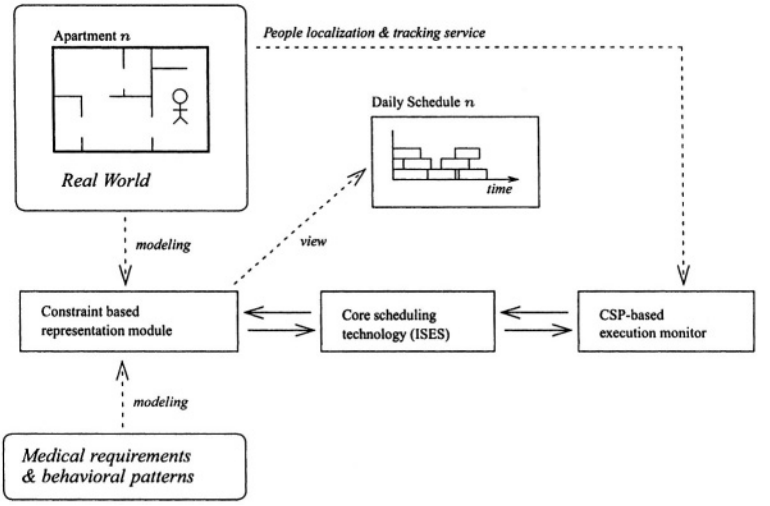


Figure 2.6. The execution monitoring system.

As shown in fig. 2.6, the execution monitor is interfaced with (a) a core scheduling system, through which it has access to the constraint representation of the schedule in its current state, and (b) with the real world, through the environmental sensory services which are responsible for signaling the exogenous events. The Schedule Execution Monitor has been developed as an integration to the *O-OSCAR* (Object-Oriented Scheduling ARchitecture) tool [2], an existing constraint-based software architecture for the solution of complex scheduling problems. The details of the core scheduling technology are outside the scope of this article. Let us now focus on how contingencies are represented and dealt with in our CSP-based execution monitoring component.

4.1 Representing Contingencies

One key feature of the Execution Monitoring System is the ability of modeling a realistic set of possible exogenous events. Information about all deviations from the nominal timing of the activities to be executed is perceived by a number of sensors distributed in the environment (such as cameras), and such information is properly processed and translated into meaningful input for the Representation Module, which constantly stores the world representation (see fig. 2.6). Typical consequences triggered by an exogenous event may reflect in the presence in the schedule of either a *temporal inconsistency* or a *resource inconsistency.* The first type of inconsistency occurs, for instance, if a delay has pushed some activities beyond some pre-defined temporal deadlines. A resource inconsistency, on the other hand, can happen if the delay has pushed an activity in a time zone where the overall resource requirement exceeds the maximum resource capacity, due to the requests of the other activities which operate in the same interval.

Three kinds of events can currently be dealt with by the execution monitor, as they represent a realistic set of incidents for the ROBOCARE scenario: *delays* of activities, variations in activity *durations* and *resource breakdowns.* Let us comment on each of these cases briefly.

As mentioned above, a sensor response may induce a sudden and unexpected time shift on one activity. This is represented by inserting a new *precedence constraint* between a particular time point (namely, the *origin* of the temporal network) and the *start time* of the delayed activity.

The next exogenous event which can be modeled in the execution monitor is a change of duration of an activity. This change is represented by substituting the activity with a new one having the same characteristics as for resource requirements, but of course different duration (which may not necessarily be *larger*).

To conclude, we model a resource breakdown by the simple insertion of a new *ghost* activity with the only aim of adding another source of contention in the schedule. The ghost activity will make use of the resources which have collapsed, *ofexactly the capacity that has to be collapsed.* In other words, the

condemned resource will be "eaten out" by the ghost activity, obtaining as an overall effect, its partial or total breakdown.

4.2 The Execution Monitor

As shown in fig. 2.6, contingencies are posted to the core scheduler, which is responsible for updating the constraint-based representation of the world, maintaining it perfectly consistent with the evolution of the real environment; the main issue is that updating the data stored in the representation module in accordance to the information gathered from the environment may introduce some inconsistency in the schedule representation. The execution monitor reacts to these inconsistencies as they are detected, namely attempting to take the schedule back to a consistent state, so as to keep it executable.

The repair action is performed by exploiting the capabilities of the ISES algorithm, which is used as a "black box"; in other words, schedule revisions are approached as *global* re-scheduling actions, without focusing on a particular area of the schedule, an approach similar to [11]. This global approach requires some preventive action to be taken before the ISES procedure is fired, in order to have the necessary control on schedule repair choices. In other words, we can guide the revision process by preventively constraining the activities, depending on the strategies we want to realize.

A number of primitives have been developed which make the dynamic insertion and deletion of temporal constraints possible. At present, these primitives handle the insertion of four types of constraints: (1) *Precedence Constraints,* which impose a temporal relationship between two activities A and B such that A *cannot start before the end of activity* B; (2) *Deadline Constraints,* which impose a time boundary on an activity's end time, i.e. the activity *cannot end after a certain deadline*; (3) *Release Time Constraints,* which impose a time boundary on an activity's start time, i.e. the activity *cannot start before a certain release time;* (4) *FixTime Constraints,* which are similar to the previous type, but impose a more rigid constraint on the activity start time, namely that the activity is *not allowed to start neither before, nor after a determined fixed instant*

By means of these four primitives, the execution monitor can effectively employ the core scheduling module to adjust the current schedule in the event that an unforseen event occurs in the context of the assisted person's nominal behavior. The pseudo-code in fig. 2.7 shows the detail of the execution algorithm, where *T* and *execSchedule* represent, respectively, time and the current nominal schedule. At predetermined intervals of time, the environment is sensed by invoking the services of the sensory subsystem, in order to detect any possible deviation between the expected and the actual situation; if unforeseen events have occurred, they are modeled in terms of the contingency representation schemata shown in Section 4.1. The next step consists in checking the *temporal consistency,* as in the schedule updating process we may have added to the

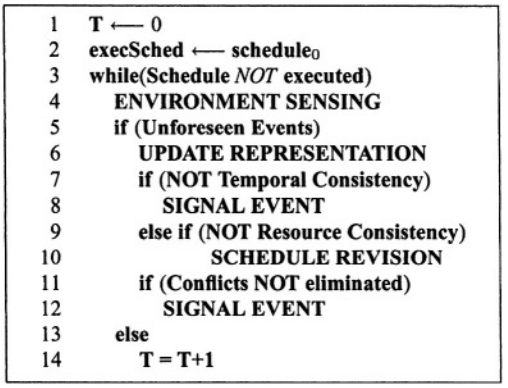


Figure 2.7. The schedule execution algorithm.

representation some temporal constraints which are in conflict with the existing ones.

If consistency is lost, the algorithm must signal the event, as the presence of an anomalous situation is detected. In this case, no repair action is possible unless some previously imposed constraints are relaxed; the signal will then be properly captured and used for instance to issue a warning or to fire an alarm. If time consistency is not spoiled, *resource consistency* must be checked as well, because the occurrence of the exogenous event may have introduced some resource conflicts in the schedule, although leaving it temporally consistent.

If no resource conflicts are present, the execution of the schedule may continue; otherwise, a *schedule revision* must be performed, in the attempt to eliminate resource contention. If the schedule revision process succeeds in eliminating the conflicts, execution may continue; otherwise the algorithm must again signal the occurrence of the anomaly for further processing.

Let us take a closer look at the way the activities in the schedule are actually manipulated during execution and repair. As previously stated, the approach used in our Execution Monitor can be considered as *global* [4, 14] in that the revision procedure accepts the schedule *as a whole,* tries to solve all the conflicts and returns the solution. In other words, the ISES procedure does not make any difference between terminated, started or yet-to-start activities, and has no concept of *time.* As a consequence, the only chance at our disposal to exert some control over the activities is to do it in a preventive way, that is, *before* the ISES procedure begins the manipulation of the schedule.

Such control is necessary for at least two reasons: (a) we want to keep the solutions *physically consistent* at all times; (b) we want to retain the possibility to satisfy a set of *preferences* given by the users.

The next paragraphs focus on some practical issues arising during schedule execution that we have to face in order to obtain meaningful results from the re-

vision process. We assume the schedule under execution with *current execution time* = t_E.

Physical Consistency. Consistency must be satisfied at all times, because the current schedule represents the nominal behavior of the assisted person. There are many ways in which physical consistency may be spoiled as a result of an inattentive action; for instance, the re-scheduling procedure may try to re-allocate some activities which have already started execution.

Clearly, this represents an inconsistent situation and must always be avoided. The problem is solved by inserting a new *FixTime constraint* for every activity whose start time $st = t_E$. By doing so, we impose a very strict temporal constraint on the activity start time: *all the solutions found by ISES which require a temporal shift of the constrained activity, will be rejected.*

As another example, the re-scheduling procedure may allocate some activities *at the left* of t_E in the temporal axis, which would be equivalent to allocating operations *in the past.* All we have to do in this case is to introduce in the schedule as many *Release Time constraints* as there are activities whose start time st is greater or equal than t_E. In other words, we constrain all the activities which have not yet started, not to begin execution before the current execution time. Again, this does not necessarily mean that these activities will be moved by the core scheduling algorithm: anyway, should they be re-allocated, they would certainly be positioned at the right of t_E.

Preferences Management. In many cases it is essential that any revised solution be as close as possible to the last consistent solution found by ISES; the closer any two solutions are, the higher their level of *continuity.*

It is in fact desirable (and plausible) that, despite the possible exogenous events that may occur during the execution of a schedule, this remains as similar as possible to the initial schedule, since it models the behavior of the assisted person. Schedule continuity can be controlled by leaving or removing the *precedence constraints* possibly imposed in the last execution of ISES. It is known that ISES resolves the conflicts by inserting a certain number of extra precedence constraints between the activities, in order to separate them in the areas of greater resource contention; these extra constraints are not part of the original problem and are only there to solve a particular resource conflict.

If ISES is run many times consecutively, it has been observed that the consecutive schedules which are obtained show higher levels of continuity if the constraints which were added at previous runs are maintained. Obviously this is due to the lower degree of freedom retained by the activities in the two cases: the more constrained the activities are, the lower the possibility that the new solution differs from the old one.

As a last observation, with a proper handling of the temporal constraints it is also possible to bias the schedule in order to satisfy user's preferences; for

instance, before schedule revision it could be possible to specify the *degree of mobility* of the activities, such as maximum delays, preferred anticipations, and so on. By exerting this kind of preventive control, it is possible to express preferences on the behavior of every individual activity before schedule revision, thus obtaining a solution which best suites the user's desires.

5. Integrating Sensing and Execution Monitoring: a Running Example

In this section we will give a description of illustrative situations in which our system monitors and controls the daily activities of an elderly person. On the basis of the following simple examples, we intend to give the readers a flavour of the potential of our tool when employed in more realistic and complex scenarios.

The agents involved in the monitoring activity are, on one hand, the people localization and tracking service described in section 3, and on the other the execution monitoring component we have just described in the previous section. The first component retains the ability to detect the presence of a person in the environment and deliver the coordinates of his or her position obtained by image analysis; on top of this basic capability, a series of *situations* can be estimated, such as when a person is sleeping, eating, and so on. On the other hand, the task of the execution monitoring agent is to guarantee the correct schedule execution ensuring its consistency at all times. Basic information on the state of the assisted elderly person is gathered by invoking the services of the localization and tracking system. This information is processed by the execution monitoring system in order to assess the actual state of the assisted person with respect to the nominal schedule.

For the sake of illustration let us suppose that the next action to be monitored is the lunch activity (fig. 2.8).

The activity to be monitored is subject to a number of previously imposed temporal constraints; such constraints are intended to model the timing of any individual operation, as well as to model the possible synchronizations among the several activities in the schedule. In this example (see fig. 2.8 (a)), the operation of eating a meal is characterized by a duration **d**, a start time **st** and an end time **et**. A proper imposition of the temporal constraints is necessary to model a situation where there is a certain amount of flexibility regarding the possible start time and end time of the activity: this flexibility is modeled through the specification of a *release time constraint* and a *deadline constraint,* defining respectively an earliest start time **est** and a latest finish time **lft** for the activity, such that $\mathbf{st} \geq \mathbf{est}$ and $\mathbf{et} \leq \mathbf{lft}$. As a consequence, the activity will retain a certain degree of mobility on the temporal axis, as a too strict timing would not be realistically acceptable. When the execution monitor requests information on the status of the assisted person, the people localization and tracking service analyzes the information it observes through the Stereo Camera. In this specific case, the basic image processing can deliver the position in

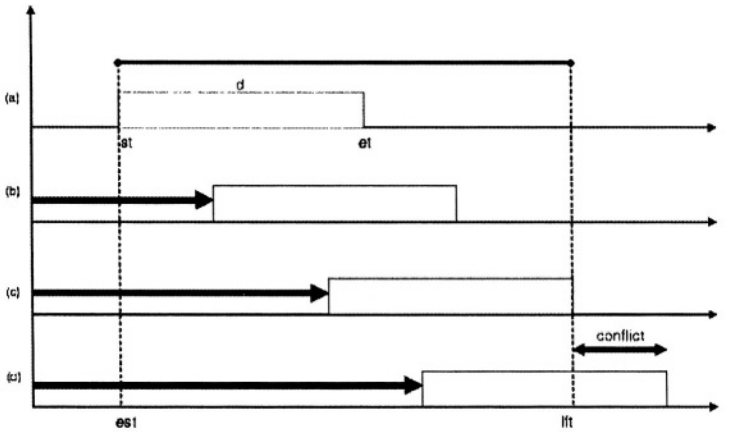


Figure 2.8. A first example.

the room of the tracked person compared with the coordinates of a series of landmarks (such as elements of furniture). Moreover, based on contextual information and landmark analysis, the LTA service can verify whether the person is indeed sitting at the table. The sensory service replies to the execution monitor's request with this information and activity monitoring proceeds under nominal conditions. In the opposite case, the execution monitor may conclude that, indeed, the person is not eating. This interaction occurs according to the e-service state machine shown in fig. 2.9. In this example, the assisted person is expected to start having lunch not before time **t = est**. This means that at **t = est** the execution monitor requests a confirmation of this from the sensory service. If this is not the case, the world representation must be modified accordingly, by inserting a *delay* on the monitored activity, that is, by imposing a new precedence constraint (see fig. 2.8 (b)). The extent of the delay can be decided depending on a variety of factors, mostly related to the frequency of the execution monitoring cycle.

Obviously, the insertion of the new precedence constraint may have direct consequences on the consistency of the schedule: fig. 2.8 (b) depicts the sit-

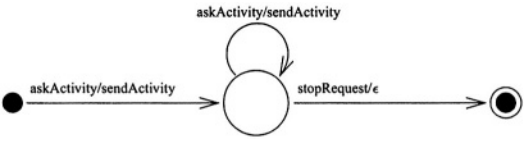


Figure 2.9. The 'what-activity' e-service state machine, whose invocation occurs through the askActivity/sendActivity messages.

uation where the temporal right shift imposed on the delayed activity is still compatible with all the previous temporal constraints: in this example, the only constraint which could be violated by the new insertion is the deadline constraint insisting on the monitored activity.

The *environment sensing/update representation* cycle continues until the imposition of a further delay on the same activity eventually generates a time conflict (fig. 2.8 (c)(d)), due to the fact that the deadline constraint is violated. The insertion of the last precedence constraint is disallowed, the system thus recognizes the inconsistency and immediately signals the event, by firing an alarm or issuing a warning.

In order to highlight the high level of expressiveness the execution monitor is able to offer, we present a slightly more complex example (see fig. 2.10). The new schedule is composed of six activities, according to the following table:

A1: breakfast **A2: lunch** **A3: dinner**
A4, A5, A6: first, second, third medical treatment

Despite the low number of activities in the schedule, the modeled situation is already non trivial: in fact, **A1**, **A2** and **A3** are characterized by their own **est** and **lft**, defining a *release time* and a *deadline* constraint for each activity. In fig. 2.10(a) these constraints are represented as bold lines starting right above the interested activity. **A1**, **A2** and **A3** are also temporally connected among one another: in fact, breakfast and lunch must be separated by a minimum interval as they cannot be too close to each other; the same thing holds between lunch and dinner. As a consequence, the breakfast activity can be delayed without affecting the timing of the lunch, as long as the temporal distance between them keeps greater than the minimum allowed; should this minimum be violated, a delay on the first activity would inevitably impact on the start time of the second. The described effect is easily obtainable by forcing of two precedence constraints, between **A1** and **A2**, and between **A2** and **A3**. In the figure, such constraints are represented as bold arrows starting right below the interested activity (the length of the arrow representing the minimum distance allowed between the activities).

The three medical treatments (activities **A4, A5** and **A6**) deserve special attention; the situation we model here, reflects the typical circumstance where taking medicines is strongly related to the timing of meals: in this specific case, we suppose that the first medicine (activity **A4**) should be taken no later than the interval ΔT_1 after the end of activity **A1**; the second medicine (activity **A5**) should be taken *immediately before* **A2**, and that the third medicine (activity **A6**) should be taken *immediately after* **A3**.

It is easy to see that the temporal relationships in this schedule involve a relatively complex interdependence among the activities, as one single delay may have several effects: fig. 2.10(b) shows how a delay on **A1** (which has

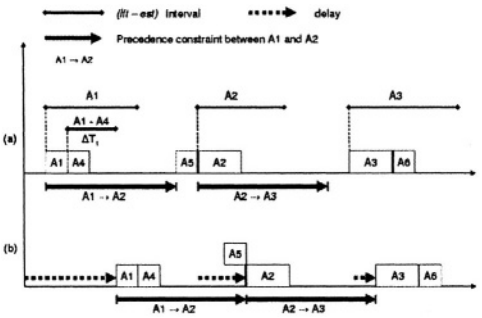


Figure 2.10. A second example.

direct repercussions on **A4**) may reflect on a delay on **A2** (and consequently on **A5**), as well as on **A3** (and consequently on **A6**). In this way, the system is able to immediately recognize the occurrence of a constraint violation even in case of significant temporal distance between the event which caused the delay and the instant of the violation. Notice that the activity which is directly involved in the delay is **A2** and not **A5**; **A5** is simply "dragged" by **A2** as a consequence of the strict temporal relationship between **A5** and **A2**.

Fig. 2.10(b) depicts a case where no constraint violation occurs; yet, in a real execution environment, many are the possibilities where the opposite may be true: the assisted person may for instance have breakfast and fail to take the first medicine in time, or may delay the lunch outside the allowed temporal window, or may forget to take the third medicine immediately after dinner. In each case, the system aims at maintaining the consistency of the flow of actions through a clever reallocation on the temporal axis.

One point which should be highlighted is that the patient's desires are taken into account by way of defining not too rigid time bounds on the schedule, as long as her or his preferences do not collide with some temporal limitations which are not relaxable as they might be imposed on the basis of essential medical prescriptions.

6. Conclusions and Future Work

In this article we have described an integration of two intelligent components, namely a Stereo Camera based environmental sensor and a CSP based execution monitoring software agent. The first component provides people localization and tracking functionalities, while the second is capable of recognizing inconsistencies in the execution of an elderly person's daily activities, dynamically reacting to exogenous events.

Since the two systems interact strongly (the execution monitoring service grounds its reactivity on the environmental observations made by the localization and tracking component), the high-level coordination mechanism has been

implemented in the form of e-services. Following the e-service philosophy, the two components are seen as agents, whose functionalities are exported as services.

These basic ingredients constitute an initial prototypical system which is currently developed for a testbed domestic environment. The ensemble of services provided by the two interacting components we have described aims at discovering the state of the assisted person and of the environment. This information is interpreted into a complex symbolic representation which can be used by the supervising entity to monitor the environment and to adapt to unexpected or unforeseeable perturbations in the expected behavior of the assisted person. Clearly, the reliability and effectiveness of the supervisory system is determined by the quality of the information returned by the sensing agents on one hand, and by the nature of the intervention the supervising entity imposes upon reacting to contingent events on the other. Currently, our system can recognize simple states based on the permanence of the assisted person in a relative position. Thanks to this symbolic information, the supervisor can monitor the correct execution of simple activities which involve these states, and signal inconsistencies which may occur as a result of delays in the sensing of an "expected" state,

In order to increase the effectiveness of the global supervisory system, our work in the immediate future will strive to i) enhance the array of situations the sensory agents can recognize (such as human posture recognition); ii) combine and integrate input from other devices such as mobile robots; iii) devise effective and pro-active contingent plans to broaden the scope of action of the reactive execution monitoring service.

Ultimately, the success of the ROBOCARE project's ambitious goals depends strongly on the integration of techniques from many disciplines.

Hopefully the proposed framework will provide novel perspectives in Ambient Intelligence, as well as a number of technically interesting solutions to component integration. We expect this work to forebode new sources for environmental information gathering, such as robotic sensors. Also, it will be interesting to integrate services offered by robotic devices with high-tech domotic components. Lastly, and certainly not least importantly, we are starting to address acceptability issues related to environmental monitoring.

Acknowledgments

This research is partially supported by MIUR (Italian Ministry of Education, University and Research) under project ROBOCARE (A Multi-Agent System with Intelligent Fixed and Mobile Robotic Components).

References

[1] D. Beymer and K. Konolige. Real-time tracking of multiple people using stereo. In *Proc. of IEEE Frame Rate Workshop,* 1999.

[2] A. Cesta, G. Cortellessa, A. Oddi, N. Policella, and A. Susi. A Constraint-Based Architecture for Flexible Support to Activity Scheduling. In *Lecture Notes in Artificial Intelligence, N.2175.* Springer, 2001.

[3] A. Cesta, A. Oddi, and S. F. Smith. A Constrained-Based Method for Project Scheduling with Time Windows. *Journal of Heuristics,* 8(1):109–135, 2002.

[4] L. K. Church and R. Uzsoy. Analysis of Periodic and Event-Driven Rescheduling Policies in Dynamic Shops. *Inter. J. Comp. Integr. Manufact.,* 5:153–163, 1991.

[5] T. Darrell, D. Demirdjian, N. Checka, and P. Felzenswalb. Plan-view trajectory estimation with dense stereo background models. In *In Proc. of the International Conference on Computer Vision,* 2001.

[6] T. Darrell, G. Gordon, M. Harville, and J. Woodfill. Integrated person tracking using stereo, color, and pattern detection. In *IEEE Conf. on Computer Vision and Pattern Recognition,* 1998.

[7] A. Davenport and J. C. Beck. A Survey of Techniques for Scheduling with Uncertainty. http://www.eil.utoronto.ca/profiles/chris/chris.papers.html.

[8] I. Haritaoglu, D. Harwood, and L. Davis. W4s: A real-time system for detection and tracking people in 2.5d. In *Proc. of ECCV98,* 1998.

[9] K. Konolige. Small vision systems: Hardware and implementation. In *Proc. of 8th International Symposium on Robotics Research,* 1997.

[10] M. Mecella and B. Pernici. Building flexible and cooperative applications based on e-services. Technical Report 21-2002, DIS - Università di Roma La Sapienza, 2002.

[11] S. F. Smith. OPIS: A Methodology and Architecture for Reactive Scheduling. In M. Zweben and S. M. Fox, editors, *Intelligent Scheduling.* Morgan Kaufmann, 1994.

[12] R. Y. Tsai. An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition,* 364–374, 1986.

[13] C. Wren, A, Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. In *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 1997.

[14] S. D. Wu, H. Storer,and P. C. Chang. One-machine rescheduling heuristics with efficiency and stability as criteria. *Comput. Oper. Res.,* 20:1–14, 1993.

[15] D. Yang, H. Gonzalez-Banos, and L. Guibas. Counting people in crowds with a real-time network of image sensors. In *Proc. of ICCV,* 2003.

Chapter 3

SCALING AMBIENT INTELLIGENCE

Compositional Devices

P.Marti[1], H.H.Lund[2]

[1] *Communication Science Department, University of Siena, Italy*
marti@unisi.it

[2] *The Maersk Mc-Kinney Moller Institute for Production Technology, University of Southern Denmark, Denmark*
hhl@mip.sdu.dk

Keywords: Ambient intelligence, interaction design, robotics, active tools, configurable environments.

1. Ambient Intelligence: the contribution of different disciplines

Ambient Intelligence represents a vision of the future where people are surrounded by electronic artifacts and environments, sensitive and responsive. Ambient intelligence technologies are expected to combine concepts of ubiquitous computing and intelligent systems putting humans in the centre of technological developments. This represents a long-term objective for European research bringing together researchers across multiple disciplines like computer science, electronics and mechanical engineering, design, architecture, social sciences, software engineering. Key concepts of ambient intelligence are:

- Ubiquitous Computing: that is wired, wireless and ad-hoc networking that exploit highly portable or else numerous, very-low-cost computing devices [17]; discovery mechanisms, software architectures, system integration and prototyping, portable devices;
- Context Awareness: sensors, tracking and positioning, smart devices, wearable, models of context of use, software architectures for multi platform interfaces;
- Intelligence: learning algorithms, user profiling, personalisation and adaptivity, autonomous intelligence, agent based user interfaces; and

- Natural user-system interaction: ambient interfaces, multimodal interaction, innovative interaction styles and concepts.
- Appreciation of the social interactions of objects in environments, and the cultural values the new environments contribute to determine.

At a general level, the interest for Ambient Systems is driven by both technology, design and user orientation. Such systems should allow to achieve interoperability of devices and integration of new services through available tools; and to orchestrate devices and entities to support and enhance human activities. To achieve this, systems have to be so designed as to support a range of human activities, and be intimately inter-twined with physical settings, consisting of spaces, places and everyday objects and materials.

These physical settings and devices share some important characteristics: ubiquity, transparency, intelligence, furthermore they can be personalized, they are adaptive and anticipatory. Ubiquity refers to a situation in which we are surrounded by a multitude of interconnected embedded systems. Transparency indicates that the surrounding systems are invisible and moved into the background of our surroundings. Intelligence refers to the fact that the digital surroundings exhibit some form of adaptation, e.g. by recognizing the people that live/interact in these surroundings in order to adapt themselves to them, learn from their behavior, and possibly show emotion.

Furthermore, as said above devices for ambient intelligence should be personalisable, adaptive and anticipatory, characteristics that may receive a fundamental contribution from modern artificial intelligence.

Adaptation can be defined simply as the act of changing to fit different conditions. The investigation and understanding of adaptation does not only lead to the possibility of creating adaptive devices, but may also be the corner-stone for creating devices that can be personalised. If devices are to be personalised, they should be able to change to the individual users need, i.e. they should adapt since they should change to fit different conditions (to fit different users, in this case). Also the anticipatory characteristic of ambient intelligence devices may demand adaptation, since anticipation demands the ability of a system to internally extrapolate future interactions with the surrounding environment, and hence adapt in an internal system. The extrapolation can be viewed as an adaptation process (a change in the internal system to fit the future condition). In this light, we view modern artificial intelligence as a most valuable field for supporting ambient intelligence.

But modern artificial intelligence, or more precisely embodied artificial intelligence, may also play an even more crucial role in the creation of ambient intelligence. Again, looking at the characteristics of ambient intelligence, we find that devices should be built into our natural surroundings and we should be able to interact with these devices in a natural way. This suggests that devices must have natural qualities like physicality and sensibility. Embodied artificial intelligence puts emphasis on the physical reality in the creation of intelligence, and suggests that intelligence cannot be abstracted away from the physical body and interaction with the physical, surrounding environment. Embodied artificial intelligence research, e.g. [20] [7], tells us that when creating

physical entities with adaptive characteristics it is often advantageous to try to find the right balance between control, hardware, material, and energy use, and that this balance may be found through a bottom-up approach. The bottom-up approach is characterised by initially creating a minimal system with a basic behaviour, and step-by-step add components (in terms of electronic hardware, control, material, energy consumption) only as they become necessary for new, more advanced behaviours built on top of each other. We believe that embodied artificial intelligence research is one of the disciplines that may provide some answers on how to create ambient intelligence in different domains. In the development of ambient intelligence, we may also find inspiration from the development and research in tangible interfaces, since also within this field, research is focused on the development of seamless interaction with physical information processing systems. Tangible interfaces [11] may represent a valid opportunity for the development of novel interactive technologies that can overcome the limitation of the current computer-based technologies constrained to screen, mouse and keyboard interaction. Several research groups developed haptic and tangible interfaces, and today, the many approaches to tangible interfaces differ in implementation and focus, while, at the same time, sharing certain main characteristics. Using various technical means, physical objects are coupled with digital representations. Any change in the physical arrangement is recognized and interpreted as a controlling action for the digital counterpart. This is in particular the feature that characterizes most of the existing prototypes of tangible interfaces. These physical objects are mainly interfaces either to a digital counterpart (e.g. navigational systems) or to a physical target system (e.g. use of tangible devices as a control systems of other physical or digital worlds). An example of the first category are the navigational blocks [2] that provide visitors of a virtual museum with a direct manipulation experience using physical blocks to navigate a data space. This interface uses the visitor's actions in the physical world to control a computational environment, in this case a database of historical information. An example of the second category is the Tangible Computation Bricks [22], physical building blocks augmented with embedded micro-processors that implement a programming language for scientific exploration. In McNerney's implementation, building blocks can only stack in one dimension, and they allow only for sequential control. Whereas this may be suitable in some cases, in other cases a more general approach allowing two or three dimensional construction and parallel control may be suitable (see I-BLOCKS control (arithmetic, behaviours, neural networks) description and the emotional construction scenario below).

These two examples, as most of the currently available prototypes of tangible interfaces, share the characteristic to specify a computation that is performed by a target system. In a sense the tangible interface is still separated by any produced output either in the physical or in the virtual world.

In this chapter, we describe a specific kind of new devices for ambient intelligence. We call these devices I-BLOCKS (Intelligent Blocks), means for exploring the design of a concept of flexible, adaptive and physical components to build intelligent artifacts and environments. With the I-BLOCKS technology, we try to take a step ahead with respect to other existing implementations

of building blocks, developing building devices able to simultaneously perform computations and to act as output devices of the intended functionality. They are not control systems but both input and output devices that are constructed by the users. Therefore our objective is to develop a concept of seamless interface to manipulate physical objects (the building blocks and the constructions obtained assembling them), to build conceptual structures (the meaning associated to each block, e.g. a math block, word block), and to compose actions (combination of output blocks like motors, LEDs, loudspeakers). Manipulating I-BLOCKS do not only mean constructing physical or conceptual structures but composing actions for building complex behaviours.

2. I-BLOCKS technology

I-BLOCKS technology consists of 'intelligent' building blocks that can be manipulated to create both physical functional and conceptual structures [5], [6]. From a technological point of view, the I-BLOCKS support our more philosophical claim that both body (physical structure) and brain (functional structure) play a crucial role in intelligence. The body and the brain of natural existences are co-evolved to fit to each other and the surrounding environment. Similarly, the 'body' and 'brain' of artificial, physical entities should be co-evolved to support embodied intelligence. The focus on building both physical and functional structures with the I-BLOCKS also lead to the possibility of investigating the concept of 'programming by building' [5], in which programming of a specific behaviour simply consists of building physical structures known to express that specific behaviour.

To allow everyday users to develop functionality of artefacts, it is important to make the process of functionality creation accessible. We believe that for everyday users it is difficult and unintuitive to split the process of artifact development into two processes of physical creation (e.g. physical construction of a robot or a mobile phone interface) and functional creation (e.g. programming of the robot or the mobile phone). Especially, we aim to avoid that everyday users are required to be able to program the pre-built physical structures in a traditional programming language. The programming in a traditional programming language demands the everyday user to learn both syntax and semantics, an approach that would exclude most people from using our technology. If we want to achieve ambient intelligence with integration of technology into our environment, so that people can freely and interactively utilize it, we need to find another approach. Therefore, we suggest moving away from programming the artefacts with traditional programming languages, and instead provide methods that allow people to 'program by building' without the need for any a priori knowledge about programming languages. Indeed, we even suggest to completely removing the traditional host computer (e.g. a PC) from the creative process.

The I-BLOCKS tool that supports investigation of this innovative way of manipulating conceptual structures consists of a number of 'intelligent' building blocks (I-BLOCKS) that each contain processing and communication capabilities. Each I-BLOCK has a physical expression (e.g. a cube or a sphere). When

attaching more I-BLOCKS together, a user may create a physical structure of I-BLOCKS that process and communicate with each other, depending on how the I-BLOCKS are physically connected to each other. Interaction with the surrounding environment happens through I-BLOCKS that obtain sensory input or produces actuation output. So the overall behaviour of an 'intelligent artefact' created by the user with the I-BLOCKS depends on the physical shape of the creation, the processing in the I-BLOCKS, and the interaction between the creation and the surrounding environment (e.g. the user themselves).

Our first implementation of I-BLOCKS uses an electronic circuit containing a PIC16F876 40-pin 8 bit CMOS Flash microcontroller for processing, and provides four 2-ways serial connections in each I-BLOCK for communication (see Figure 3.1). In order to better visualise the concept, we have chosen to make the housing out of rectangular LEGO DUPLO bricks [1]. So in this case, each building block contains the four serial two-way connections as two connections on the top and two connections on the bottom of each brick. In other implementations, there may be more or less connections, for instance there may be six connections (one on each side) in a cubic building block, or another number in a spherical building block.

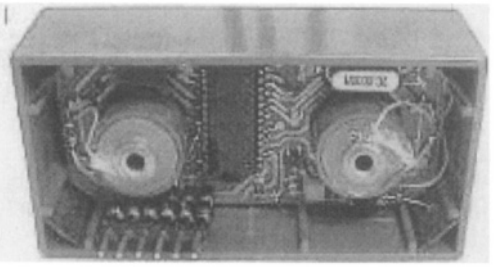


Figure 3.1. A building block with microprocessor and communication channels. ©H.H.Lund, 2002.

Energy power from a battery building block is transported through the construction of I-BLOCKS via connectors in the corners on the bottom on each block and connectors in the studs on top of each block.

There exist different types of I-BLOCKS that all share the same standard technology of providing processing and communication capabilities. We term these standard building blocks. In a number of cases, the standard building blocks are extended with the addition of sensors in order to become input building blocks, and in a number of cases extended with the addition of actuation in order to become output building blocks. Sensor building blocks include building blocks with LDR sensors, IR sensors, microphones, switches, potentiometer,

[1] LEGO and LEGO DUPLO are trademarks of LEGO System A/S.

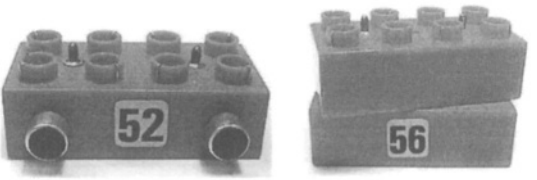


Figure 3.2. Examples of intelligent building blocks implemented in LEGO DUPLO. In this implementation, there are two connectors on the top and two on the bottom of each neural building block. On each stud, there is connection for power transfer. Left: example of sensor building block that contains two microphones. Right: example of motor building block that contains a servo motor that allows the top element to turn. ©H. H. Lund, 2002.

flex sensors, accelerometer, and output building blocks include building blocks with servo motor, DC motor, IR emitter, LEDs, sound generator, etc. (see examples in Figure 3.2). Some blocks may also combine input and output such as I-BLOCKS with ultrasound emitter and receiver, and I-BLOCKS with radio emitter and receiver.

In order to verify the technological possibilities of the I-BLOCKS, e.g. versatility of control methods, we implemented different kinds of processing in the I-BLOCKS, making the I-BLOCKS becoming arithmetic blocks [5], behaviour blocks [5], neural blocks [6], spiking neural blocks [13], language blocks [8], and emotional blocks [19]. In most of these cases, the specific control implementation was driven by a wish to investigate the I-BLOCKS in a pre-defined activity for the users, e.g. as defined by teachers, therapists, engineers, etc. However, it seems apparent that one of the strengths in the I-BLOCKS system lies in the possibility for the user to be creative and expressive, so a second strand of control implementations has focused on developing I-BLOCKS control that supports free activities that are not pre-defined by the designer. Both kinds of implementations have their own pros and cons, and can be utilised in different kinds of scenarios, as will be described below.

3. Design process

In general, we believe that effective ambient intelligent solutions should offer the opportunity to the users of meaningful, engaging and intellectually interesting activities. They should definitely address specific objectives that can support users to perform potentially new activities in an effective and stimulating way. To achieve such objective, we articulated the design process on four components: User Research, Concept Development, Content Development, Enabling Technology.

User Research In envisaging scenarios of use of our I-BLOCKS, we first of all investigated how these tools mediate construction of meaning and organization of knowledge. The description of how these artefacts may interact with other physical and cultural artefacts already present in the environment

is seen as equally important. This phase of work included rich ethnographic observation of users at work with the I-BLOCKS, e.g. children in learning and play contexts, and participatory design sessions with stakeholders to define requirements, discuss potentialities on the technology and evaluate solutions.

Concept Development In order to create "an experience" it is necessary to develop a system that supports the creation and management of behaviours, meaning and senses. Our I-BLOCKS are designed to support the manipulation not only of physical and functional components but also conceptual and sensorial contents. Furthermore, in order to create "one collective experience" we studied elements of cooperation, rules of composition, language, memory, interpretation and representation. Parts of these elements are still open points and research topics but a challenging aspect of studying the manipulation of I-BLOCKS is the definition of a new concept of manipulation of actions allowed by the I-BLOCKS rather than simply manipulation of objects and use of their functionality.

Content Development We developed different kits of I-BLOCKS (see the scenarios described in the following later) representing the physical and conceptual structures that the users can manipulate in their activity. The arithmetic I-BLOCKS described below are examples of contents that can be implemented in the I-BLOCKS, other examples are linguistic components like elements of a sentence to compose. An open research question is at which level the I-BLOCKS should be pre-programmed and at which granularity and how much space is left for children to create their own contents.

Enabling Technology The I-BLOCK technology is continuously adapted and evolved according to the requirements of the specific applications. As a general aim, it supports the manipulation of physical, function and conceptual structures.

4. Scaling Ambient Intelligence at level of compositional devices: predefined activities

So far, we applied this design process to the development of different scenarios. In most cases, the activity would become pre-defined for the end-user, e.g. working with specific arithmetic problems, language grammar problems, etc. Though the problem domains are pre-defined in these cases, and therefore we term these pre-defined activities, it should be noted that the construction within this problem domain is free for the exploration by the end-user. But in all cases, there will be an output from the I-BLOCKS construction that provides a feedback to the end-user on the suitability of the construction within the particular, pre-defined activity domain. Hence, in the active construction process, the end-user is provided with feedback from the I-BLOCKS construction that may provide guidance within the particular problem domain.

4.1 Arithmetic training

A simple example of such a pre-defined problem domain activity is the arithmetic training with arithmetic building blocks. The arithmetic building blocks are quite illustrative, since they clearly show the role of the morphology of the construction in defining the functionality. The arithmetic blocks include blocks for addition, subtraction, multiplication and division, sensor input blocks for setting input values, and output blocks to present output values. As an example let us look at the task of calculation of the results of arithmetic expressions, and here specifically the task to present a result for the following expression:

$$(x + y) * z \tag{3.1}$$

where x, y and z are either standard sensory inputs or user set inputs (the user can set input values with sensor building blocks by pressing a switch or turning a potentiometer). The task has been solved correctly for a solution that presents the right result with regards to every possible input. This system with arithmetic blocks solves this task quite easily as shown in Figure 3.3, because this system is built for arithmetic operation. The correct result will only be presented on the display (the block on the lower left corner) for as long as the sub-results and the final result do not exceed the a value of 255, which is the maximum value possible for the value data type, and for communication in general.

The built structure can be seen in Figure 3.3, where the binary display block on the lower left corner shows the result of adding two user set values (on the top) and then multiplying it with the user set value (on the right side on top of the multiplication block).

This example is illustrative for the role of morphology in determining the functionality of the creation. If the morphology is changed, then the functionality (output) is changed accordingly. Imagine exchanging the two standard building blocks that perform the two arithmetic operations of multiplication and addition. By doing so, the functionality would no longer be $(x + y) * z$, but rather $(x * y) + z$. Or if the substructure of two blocks performing $*z$ is moved one level upwards in the structure, then the resulting overall structure will have the functionality of $x + (y * z)$. So it is evident that even small changes of the morphology result in changes in the overall functionality of the artefact. Imagine that the user has set the input values to $x = 3$, $y = 7$, and $z = 15$, then the result (output) of the original creation would be (3 + 7) * 15 = 150, but (3 * 7) + 15 = 36 with the second morphology, and 3 + (7 * 15) = 108 with the third morphology.

In the example, the user is 'programming by building', i.e. the user is constructing different arithmetic expressions by building different physical structures and the structures each produce an output according to the particular physical shape. There is no traditional programming going on, but the user is simply manipulating with the physical construction in order to create an appropriate functionality. In this way, we obtain a much more natural user - system interaction than in the traditional case of first constructing a hardware artefact and then programming the artefact in a programming language such as assembler, Fortran, C, C++, Java or similar. Obtaining such a more natural interaction

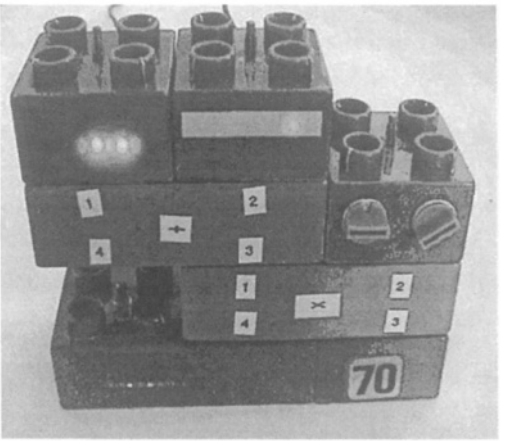


Figure 3.3. AA construction with arithmetic I-BLOCKS that performs an arithmetic calculation: $(x + y) * z$. The user can set the value of the small blocks on the top (x and y) and on the right side (z). The result is displayed as binary number with LEDs on the lower left block. ©H. H. Lund, 2002.

is crucial for ambient intelligence, since ambient intelligence demands not only an intuitive use, but also the availability to all everyday users.

4.2 Storytelling Play Scenario

We applied the design process, described in section 3, to the development of a storytelling play scenario, based on work performed in collaboration with Scuola elementare Tozzi (Siena, Italy), Scuola elementare Marliana (Pistoia, Italy), Scuola elementare Traversagna (Pistoia, Italy). The objectives of this scenario, called also the "living tree scenarios", concern the potentiality of I-BLOCKS to support creative processes in developing narratives and in externalising emotions associated to "living entities" developed by children. The living tree is an installation placed in a park that can be configured by the users. Trees can grow up and change according to weather conditions and seasons; they can simulate the movements of sunflowers, following the sun and to be reactive to temperature and wind. The user can personalize the tree through the trees memory: the possibility to put sounds or words into the blocks. The main idea becomes to have an installation, in a public open space, where children can improve their creativity making/constructing their own "toys", using the I-BLOCKS. We want to create a sort of collaborative space where children can negotiate their abilities of constructing stories around the *living tree* attributing emotional states to the structure. The emotional state of the single tree is being represented by an expression/behaviour of the construction (e.g. flashing lights for meaning happiness).

In order to investigate this possibility, we made a first mock-up using the LEGO DUPLO implementation of I-BLOCKS. The tree is able to react to the environment, in which it is situated, and the behaviour of the tree is decided by wind, temperature and light conditions. The specific behaviour of each parameter is:

- Wind: the tree moves randomly according to wind.
- Temperature: the tree changes colour according to temperature.
- Light: the tree moves its light sensors towards the light source.

So the living tree mock-up was implemented using push-button brick (for modelling wind pressure), turn-button brick (for modelling temperature changes), LED-brick (for modelling changing of colour), LDR-brick (for allowing movement towards light), top-turned motor brick (for movement), battery brick (for power supply). The user can build different constructions (trees), and the behaviour of the tree will depend on the physical construction and the interaction with the tree. A representative example is shown on Figure 3.4. The scenarios

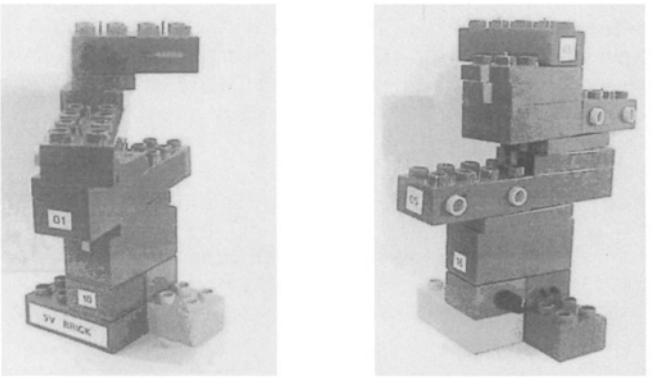


Figure 3.4. A mock-up example of the living tree made from the small I-BLOCKS, front view and back view. The construction will turn towards light, change LED activation, and make random moves, depending on the interaction with the surrounding environment.

shows a number of different qualities of I-BLOCKS, in particular their ability to sustain meaning construction, development of emotional knowledge and the opportunity to define an alternative semiotic system for expressing emotions.

Meaning construction In the vast majority of current software and educational games for young children plotlines and characters are already provided and there may be little, if any, scope for them to use their imagination for example, to decide who the characters should be or what they should look like or do. This suggests, therefore, that there is a big difference between how children

play in their everyday life where they like to build stories and characters [3], and what they are able to do with the kinds of technological playing currently supported by computer-based technology.

Our aim with the use of I-BLOCKS is to bring children into a creative process both at level of the construction of physical characters of the story and at the level of the behaviour and emotional attitude these characters can exhibit in the story. Given the above, a number of research challenges arise:

- How can we harness the power of technology to sustain an imaginative, improvisational process of meaning construction?
- What kinds of interactivities and external representations to implement in the I-BLOCKS to support the child in constructing their plays? For example, how much of the behaviour, emotions and other attributes of constructed characters should the user decide and how much should be pre-determined through the implementation of basic behaviour of the I-BLOCKS?

Emotional knowledge Bringing the I-BLOCKS constructions to life is not just a challenge from the point of view of technology. Since we are envisioning the possibility to build highly interactive and engaging characters, these have to be realized as individual personalities with their own desires, emotions and behaviours. This implies for children the development of cognitive abilities to make a plan to build and represent personality and emotions through the combination of I-BLOCKS. To study this, we conducted a series of pilot experiments where children were asked to draw and to build characters (living trees) using different materials (cardboard, plasticine, coloured papers etc) with emotional states (like a happy tree for example, see Figure 3.5). At the end of the task the

Figure 3.5. Pilot test where Italian children are drawing and building characters.

children were asked to describe what they realised and why they believed the character had a particular emotional state. These preliminary experiments had the objective to collect information about the different emotional states the children were able to recognise and the elements they judged *important* to represent an emotional state. These elements were later used to implement functionality of single I-BLOCKS.

Furthermore, to study how children can represent personality and emotions we reviewed the scientific literature on emotions in the childhood, in particular the 'Five Factor Model of Personality' [21] and 'The Cognitive Structure of Emotions' Model [1]. Based on evidence suggesting that some emotions are universal [18], we focused our analysis on anger, fear, happiness and sadness. These emotions can be interpreted easily by children of age 4-8 [12]. However, in our implementation these emotions do not directly map single I-BLOCKS but can be "constructed" combing I-BLOCKS behaviours obtaining changes in movement and lights that children use to mean emotional states (fear, happiness, etc). The life-likeness of a character is increased by the I-BLOCK technology that allows the construction to react to external events (e.g. temperature, touch, sound). Therefore, emotions are not pre-computed but directly manipulated by children through the physical construction of I-BLOCKS.

Semiotic system A further challenging topic of our research is the definition of a high-level language or semiotic system for defining the characters' behaviour and personality traits through the combination of I-BLOCKS. Our aim is to build up a basic kit of I-BLOCKS with a repertoire of behaviours that the children can manipulate to enable the definition of different behavioural styles and personalities.

The use of I-BLOCKS kit contributes to experiment on the above mentioned issues as described in [9].

4.3 Linguistic scenario

The linguistic scenario was defined in collaboration with the Cognitive Rehabilitation Centre, Ospedale Le Scotte (Siena, Italy). In the linguistic scenario we transfer to the I-BLOCKS a well-known task used by speech therapists in the rehabilitation of children with linguistic problems,, in order to give more feedback and more sensorial information to child. Our hypothesis is to test whether external representations, in the form of dynamic I-BLOCK constructions, would assist children in learning linguistic structures in a more effective way than with the combination of static iconic pictures like the ones currently used by speech therapists. The speech therapist indeed tries to teach to children with language problem the right structure of a sentence. During ethnographic observation sessions and interviews with therapists, we realised that the manipulation of objects is a very important feature in order to reach language skills. Every task has the form of a game, in which the speech therapist helps the child, giving scaffolding to the task.

Children with dyslexia, i.e. a difficulty in the scholastic learning, or with a SLI, i.e. Specific Language Impairment, can have problems to understand the structure of a sentence, and the speech therapist tries to help using lots of instruments. One of a task that the speech therapist purveys is to construct a sentence with special cards. These cards have different shapes: (1) Small and tall green rectangle for article, (2) Rectangle with an icon on for noun, (3) Red arrow for verb, and (4) Small square for preposition.

At the beginning all the sentences have the structure: Subject + Verb + Object, then it can be possible to add prepositions, adjectives, adverbs, and so on. In this task child manipulates directly the cards, and every card represents a well specified part of the phrase. The feedback, obviously, comes from the speech therapist. However, with the I-BLOCKS, we developed a system where the child is manipulating the structure of sentences when manipulating with the physical structure of I-BLOCKS, and at the same time receive feedback from the I-BLOCKS construction. In the first mock-up implementation, we have decided to preserve the characteristic of article (the small dimension), and to give different colours for different roles in the sentence: (1) Red small brick for article, (2) Green brick for noun, (3) Yellow display brick for verb.

So a sentence construction could take the form shown in Figure 3.6. The I-BLOCKS can now give feedback and sensorial information to the child based upon the physical structure that the child has created.

Following the Distributed Cognition approach [4], our hypothesis in designing I-BLOCKS for the linguistic scenario was that it may be possible to enhance cognition by mapping problem elements (components of a sentence) to an external, manipulative, physical and reacting construction in such a way that solutions become immediately evident and the children can receive feedback on their action of combining I-BLOCKS. [15] propose a theoretical framework in which internal representations and external representations form a "distributed representational space" that represents the abstract structures and properties of the task in "abstract task space" (p. 90). They developed this framework to support rigorous and formal analysis of distributed cognitive tasks and to assist their investigations of "representational effects [in which] different isomorphic representations of a common formal structure can cause dramatically different cognitive behaviours" (p. 88). "External representation are defined as the knowledge of the structure in the environment, as physical symbols, objects, or dimensions (e.g., written symbols, beads of abacuses, dimensions of a graph, etc.), and as external rules, constraints, or relations embedded in physical configurations (e.g., spatial relations of written digits, visual and spatial layout of diagrams, physical constraints in abacuses, etc.)" [14] p. 180).

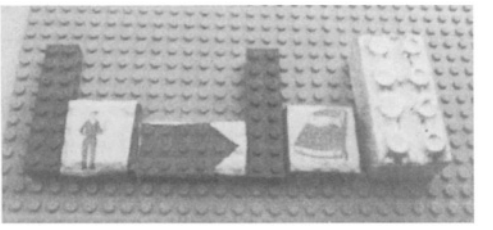


Figure 3.6. Possible construction with the I-BLOCKS mock-up for the linguistic task.

Experiments with children with dyslexia have been performed to study if the "representational determinism" of I-BLOCKS can effectively sustain lin-

guistic learning. Preliminary results performed at the Ospedale Le Scotte were encouraging and revealed a number of interesting properties of the bricks [10]:

- The interactive bricks sustained trial-and-error activity. The children were stimulated to seek different configurations of bricks and rapidly check the results.
- Children were in control of the experimental setting. They could check themselves the results of the activity without the support of the therapist.
- Children were much more involved in the activity. The same task performed with cards resulted boring and in some cases frustrating.
- Children were encouraged to reflect on the structure of the sentence in case of error.

5. Scaling Ambient Intelligence at level of compositional devices: free activities

As said before, a second strand of scenarios was focused on implementing I-BLOCKS tools that provide possibilities for the end-users to define their own problem domain, rather than to work with a specific, pre-defined problem domain as were the case with I-BLOCKS for arithmetic training, cognitive rehabilitation, and storytelling activities.

The rationale for developing this kind of I-BLOCKS is partly triggered by the investigation of I-BLOCKS use amongst users from different cultures (e.g. Denmark, Italy, Finland, Tanzania), where we find that under some circumstances the activities will, quite naturally, have to be individualised to the particular culture. For instance, it seemed appropriate during our explorative use of I-BLOCKS in Africa (see e.g. [8]) to allow the users to develop artefacts with a purpose closer to the everyday observations found in the daily life in their own culture than simply imposing building artefacts known from a Western culture. Similarly, differences in activities are quite obvious between children and adult, between school and hospital, etc.

The investigation of I-BLOCKS for free activities is targeted towards the development of a set of building blocks that can function as a set of atoms, which can be combined together in a large number of combinations to provide the opportunity to create the largest variety of applications. Hence, for this purpose we view the individual I-BLOCKS as atoms, and they are each developed to provide opportunities in expanding the number of possible end applications in the combination with other I-BLOCKS. In order to make the initial set of such I-BLOCKS for free activities, the authors together with J. Nielsen, L. Giusti, and K. Lundberg used the observations from the experiments with I-BLOCKS for pre-defined activities described above to develop a number of example uses that may hold a variety of application possibilities and demand a large degree of versatility from the I-BLOCKS implementation. For instance, these included a kid's room, in which we could imagine a variety of I-BLOCKS artefacts such as an alarm clock, a light controller, an intruder alert system, a thermostat, and play tools such as a versatile music instrument or a compass that could be used

both inside and outside the room. The kids' room scenario was used to guide the development of the first I-BLOCKS implementation for free activities, since it provided a scenario that demanded the development of I-BLOCKS, which can be combined in a large variety of ways to construct the many different artefacts needed in the kid's room. Later, the suitability and versatility of these I-BLOCKS for free activities were verified with users in Tanzania, who used these I-BLOCKS to construct artefacts of their own invention (e.g. in their scenario of a car alarm with radio communication to a mobile phone).

The I-BLOCKS that we developed included sensor I-BLOCKS such as tilt sensor (with accelerometer), microphones, flex sensor, and actuator I-BLOCKS such as LEDs, sound generator, motor (for vibration) I-BLOCKS. Also, standard I-BLOCKS were implemented to become:

- **Inverter block**: Inverts its input, so that its output will be negative, when the input is positive and vice versa.

- **Threshold block**: Sums its inputs and compares it with its user-determined threshold. The physical block has a push button to set the threshold level, and 8 LEDs to show the threshold level.

- **Timer block**: The timer block is implemented in a LCD display block, and sends out values between 127 and 255. The timer starts at 127 and counts up to 255 one-by-one when the start button is pressed. The timer does either count up every second or minute, dependent on what the user has set it to.

- **Memory block**: Records its input for a maximum 10secs, when the record button is pressed and stores it in the block's EEPROM afterwards. The data can then be played back.

- **Warm/cold block**: small blocks that are red or blue, and which sends a high or low value on the output channels.

Also, together with T. D. Ngo, we developed radio communication I-BLOCKS that can be used together with this set of I-BLOCKS for the free activities. Each radio communication I-BLOCK contains both radio emitter and receiver. With this set of I-BLOCKS, we found in explorative studies with university students in Siena (Italy) and Iringa (Tanzania) that they were able to develop a large variety of simple artefacts such as alarms dependent on user set thresholds and light input, alarms dependent on a timer, musical instruments, etc. Also, students were able to develop more complex groups of interacting artefacts, such as an alarm dependent on proximity and touch, which with the radio communication would send alarms to another artefact that would vibrate or emit different sounds - in a scenario modelling a car alarm, which sends its alarm to a mobile phone - see Figure 3.7.

Figure 3.7. The development of two communicating artefacts with the I-BLOCKS for free activities. When the user interacts with the I-BLOCKS artefact on the top through approaching it or touching it, the artefact sends a message through radio communication to the I-BLOCKS artefact on the bottom that will respond with different motor actions or sound patterns.

6. Scaling Ambient Intelligence at the level of configurable environments: future scenarios

The use of intelligent building blocks as ambient intelligence systems is not restricted to the small scale as exemplified above with the I-BLOCKS. Indeed, the modular aspect of building blocks may also provide interesting ambient intelligence possibilities in other scenarios, e.g. on a larger scale or for self-reconfiguration. We are currently making the first investigations of such expansions of the concept to different scales and uses in an academic-industrial research project on augmenting playgrounds with intelligent processing and responses to user interaction, and in an EU FET project on the development of self-reconfigurable and self-repairing robots. We briefly describe these possible future expansions in relation to the intelligent building blocks concept.

6.1 The Augmented Playground

We investigate the utilisation of building blocks on a large scale with an implementation of *tangible tiles* developed for a playground scenario with the aim to increase physical activity amongst the youth. The Western countries experience increasing problems related to obesity, and often the response from the society becomes new politics related to diet information campaigns or health care. However, a complementary route may be to increase the possibilities for physical activity - for the youth this should not only be limited to sports but to a general attractiveness to further physical activity. Unfortunately, during the

latest decades, the spaces for physical play have been limited, most notably in the city space. Most city spaces may be defined as hostile rather than friendly to children's physical play.

We believe that a variety of our ambient intelligence approach presented above may provide new opportunities to increase the possibilities for physical play in the modern city space. In order to investigate this, we work together with the companies Kompan (playground producer) and Danfoss Universe (fun and science park), and the academic partners from Mads Clausen Institute, and the Danish University of Education on the development of tangible tiles as the first technological implementation of a long design plan towards small, distributed units with processing and communication capabilities that can be utilised anywhere in the modern city space for enhancing play opportunities.

The first implementation consists of tangible tiles to be configured on the ground *(2D),* whereas further development is being done in going towards other building block implementations for $2^{1/2}D$ and $3D$ (e.g. smaller objects with radio communication). The tangible tiles adhere to our definition of building blocks in having a physical expression, and being able to process and communicate (through communication, sensing and actuation). For the first implementation of tangible tiles, we moulded a 40cm*40cm rubber pad that children can jump on, see Figure 3.8 and Figure 3.9. Each tile contains an ATMEL AVR 8 micro processor and a communication board. Output is provided with 9 points each with two ultra bright LEDs, a red and a blue LED. Plexiglas rods are moulded into the surface in a 3 by 3 grid. Each rod can be enlightened by blue or red light from the LEDs at the bottom of the tile. A piezoelectric sensor between two aluminium plates is used to sense the children stepping on the title. The signal from the sensor is amplified with a fixed amplifier before being fed to the micro controller. A regulator allows the tile to run on 9V, for instance so that a standard 9V battery can be used in each tile. An additional processor on a serial line is used to control the communication on four lines (and could be expanded to wireless communication in future). Other prototype tiles with additional vibration output were also tested.

The tiles can be put together in different configurations, e.g. $4 * 4$ tiles in a square or $1 * 16$ tiles in a line.

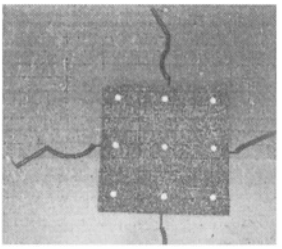


Figure 3.8. A tangible tile as a rubber pad with processing, input (pressure), output (light) and communication on the four communication lines.

With the tangible tiles it becomes easy to implement plays such as Red vs. Blue where the colour of a tile will turn from one colour to the other when jumped upon. Here the overall behaviour of the system will depend on the interaction from the users, who by the interactions with the system decides the overall light patterns. Another simple implementation may be the creation of lines and patterns - each jump on the tile creates a new line using some of the 9 LEDs in the individual tile or on neighbouring tiles by utilising the communication (e.g. a light snake chasing a child or a light tail behind the child).

Figure 3.9. The tangible tiles put in different configurations to create different plays indoor and outdoor.

The basic implementation described above is currently being adapted to sustain different kinds of play with purposes that are not restricted to the stimulation of sensory-motor coordination like in traditional playgrounds. For example, tests are currently underway to study a variety of complementary objectives like the stimulation of senses to successfully accomplish a game (e.g. blindfold children who have to communicate tactile sensations to discover, in a cooperative way, couples of titles with similar properties); spatial orientation, dialogical coordination; production of cognitive strategies to win the game; cooperation through the sharing of space, attention, discovery of rules.

Based on these tests at local schools of children playing with the tangible tiles, it became clear that there may be five areas which contain central aspects in children's approach to play tools, namely (1) performance, (2) engagement; (3) initialisation of play, (4) inspiration for new plays, and (5) play with game play. Interestingly, the possibility to physically reconfigure the tiles provided numerous new play opportunities for the children in allowing them to reconfigure the physical arrangement and thereby the overall behaviour of the system. In a similar way to the manipulation on small scale with the I-BLOCKS, on the larger scale with the tangible tiles the children can construct different functionalities of the system by the physical arrangement of the building blocks (i.e. the tangible tiles). Indeed, by changing the physical arrangement of tiles, the children may, for instance, develop new performances or change level of difficulty in plays with game play. The latter can be illustrated by our implementation of a Pong game. In this game, the light patterns move in a random pattern from one side of tiles to another side, and a child should jump on the tile where the light appear on the end row. If the children place the tiles in four rows, then the light pattern has to traverse four tiles before a child has to jump on a tile, whereas if

the children make a physical configuration with only three rows, then the light pattern has to traverse only three tiles in between jumps on the tiles, making the game faster and more difficult. Hence, the physical arrangement of the tiles may define the level of difficulty of the game. Figure 3.10 shows two examples of the Pong game with $2 * 5$ tiles and $2 * 3$ tiles.

In general, as with the I-BLOCKS, the overall behaviour of an 'intelligent artefact' created by the users with the tangible tiles depends on the physical shape of the creation, the processing in the tangible tiles, and the interaction between the creation and the surrounding environment (e.g. the users themselves).

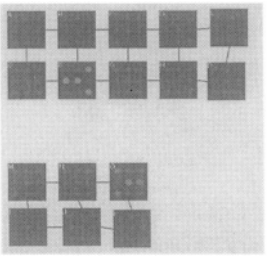


Figure 3.10. Two different physical configurations with the tangible tiles. On the top, the light pattern has to traverse 5 rows of tiles, whereas on the bottom it has to traverse only 3 tiles, making the Pong game more difficult.

6.2 Self-reconfigurable Robots

In the case of I-BLOCKS and Tangible Tiles, the modularity is used to allow end-users (e.g. children) to configure building blocks to construct the overall behaviour of the artefact. However, it is also possible to utilise this concept to create self-assembling robotic artefacts, where it is not the user who reconfigures the system into the appropriate configuration, but it is the system itself that does so. Again, in order to utilise this concept, it is necessary to have building blocks with a physical expression, processing and communication (through neighbourhood communication, sensing, and/or actuation). Notably, for achieving self-reconfiguration, the building blocks need to provide physical displacement in some way.

We are developing building blocks for self-assembly called the ATRON modules (see Figure 3.11). The basic idea behind the ATRON modules is to make have two half cells joint together by a rotation mechanism. On each half cell, there are two female and two actuated male connectors, by which a module can connect to the neighbouring modules. A module may communicate with neighbouring modules through IR communication. The modules can be placed in a surface-centred cubic lattice structure (corresponding to the titanium atoms in the CuTi3 crystal lattice), and move in this structure to self-reconfigure into

different overall arrangements or movements. (Further details regarding the ATRON hardware module can be found in [16])

If a first ATRON module is attached to a second neighbouring module and detached on other connection points, the second neighbouring module may move the first ATRON module by turning around equator with the rotation mechanism. Hence, the first ATRON module may be moved to another position in the lattice structure where it may attach itself to another module in the structure and, for instance, detach itself from the second ATRON module that transported it to the new position.

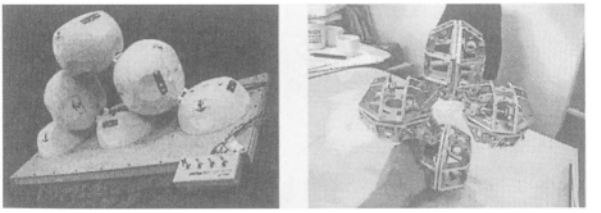


Figure 3.11. The first and the final hardware prototype of the ATRON modules.

As illustrated above, the reconfiguration of the overall system becomes a process of transitions in the lattice structure. Simulations show that, if we can perform the individual transitions in our hardware implementation in a reliable manner, numerous distributed control possibilities exists and will lead to self-reconfigurable and mobile systems. So the ATRON modules aim at providing self-assembly, where the I-BLOCKS and the Tangible Tiles relies on the user to perform the assembly of the building blocks. However, the technological approach is quite similar in that each ATRON module is a building block with physical properties, processing and communication capabilities. So the overall characteristic of the system of ATRON modules is - as with the other building block systems - decided by their physical arrangement, their processing, and their interaction with the environment.

7. Discussion and conclusions

When providing ambient intelligence solutions we believe it to be important to utilize the new ambient intelligence possibilities to provide further creativity possibilities than those that are available with traditional edutainment tools. Ambient intelligence allow people to freely and interactively utilize the new technology that is integrated into our surrounding environment in numerous, small devices. Our goal is to enhance people's creativity by introducing the new kind of I-BLOCKS devices that supports this ambient intelligence vision. Hence, the free and interactive utilization of the new technology should not only allow people to become users, but should allow people to become creators of the new solutions.

Therefore, in this chapter, we have described a new tool and some scenarios in which it can be used in a creative manner by everyday users. The possible scenarios span different scales. We emphasise the development for both predefined activities, which are suitable for meeting some specific education and rehabilitation objectives, and for free activities, which are targeted towards allowing users to define both the scenarios and the creative constructions for these scenarios.

Intentionally, we developed the I-BLOCKS to become small devices with only limited processing within each I-BLOCK, so that the overall behaviour of a construction depends on a distributed processing among the collection of I-BLOCKS in the construction. We are essentially trying to define the atoms or cells or building blocks that allow the best affordances and furthest possibilities for engaging activities. We believe that a centralised approach of providing most processing within one centralised unit (as is the traditional approach in most technological entertainment systems) will limit the creative possibilities, and therefore we are trying to distribute processing to all I-BLOCKS in order to provide an open system, where the user can create new behaviours by assembling many I-BLOCKS into a physical structure.

I-BLOCKS allow rich sensorial interaction where reality can be explored, decomposed, built and analysed. What a child builds can be combined with the products of other children in a continuous negotiation process where the evolution of transformation of the construction can be used as a way to understand the other point's of view.

Although the first implementation of I-BLOCKS provided interesting insights for designing ambient intelligence solutions, yet many improvements are still possible as was evident from the testing currently underway. For instance, our examples of future scenarios show that it may be possible to use a similar concept for work a larger scale for enhancing physical activity, and also to develop self-assembling robots. Likewise, important issues remain that to be addressed. Most of them are at the interaction level, that is at the level for example, where you need to refine the way in which intentions are mapped into action, where actions take on shapes that evoke their meaning and functionality, where the I-BLOCKS react promptly and consistently to the action and intentions, and where the modality and form of the feedback is meaningful for the user.

Acknowledgments

Part of the implementation work was performed by J. Nielsen, S. Jensen, M. Pedersen, K. Lundberg, T. D. Ngo, T. Klitbo, C. Balslev. L. Giusti, V. Palma, A. Rullo, I. Bartolucci, M. Vesisenaho, and E. Suttinen collaborated to the user research and testing with children. They all provided valuable discussions and contributed to the definition of the project vision. The work is partly sponsored by the Danish National Research Council project 'Intelligent Artefacts'. The future scenarios are collaboration work in the IT-Korridor project Body Games with Kompan, Danfoss Universe, Danish University of Education, and Mads

Clausen Institute, and in the EU FET project HYDRA with University of Zurich, University of Edinburgh, and LEGO.

References

[1] A.Ortony, G.L.Clore, and A.Collins. *The Cognitive Structure of Emotions.* Cambridge University Press, 1998.

[2] Ken Camarata, Ellen Yi-Luen Do, Brian R. Johnson, and Mark D. Gross. Navigational blocks: navigating information space with tangible media. In *Proceedings of the 7th international conference on Intelligent user interfaces,* pages 31–38. ACM Press, 2002.

[3] C.Fusai, B.Saudelli, P.Marti, F.Decortis, and A.Rizzo. Media composition and narrative performance at school. *Journal of Computer Assisted Learning,* 19:177–185, 2003.

[4] E.L.Hutchins. *Cognition in the wild.* MIT Press, 1995.

[5] H.H.Lund. Intelligent artefacts. In *Proceedings of 8th International Symposium on Artificial Life and Robotics,* pages I11–I14, 2003.

[6] H.H.Lund. Neural building blocks. In *Proceedings of 1st International IEEE EMB Conference on Neural engineering,* pages 446–449, 2003.

[7] H.H.Lund, L.Pagliarini, L.Paramonov, and M.W.Jørgensen. Embodied ai in humanoids. In *Proceedings of 8th International Symposium on Artificial Life and Robotics,* pages 369–372, 2003.

[8] H.H.Lund and M.Vesisenaho. I-blocks in an african context. In *Proceedings of 9th International Symposium on Artificial Life and Robotics,* pages 17–I12, 2004.

[9] H.H.Lund and P.Marti. Physical and conceptual constructions in advanced learning environments. *Interaction Studies,* 5(2):269–299, 2004.

[10] H.H.Lund, P.Marti, and V.Palma. Educational robotics: Manipulative technologies for cognitive rehabilitation. In *Proceedings of 9th International Symposium on Artificial Life and Robotics,* pages I1–I6, 2004.

[11] H.Ishii and B.Ullmer. Tangible bits: towards seamless interfaces between people, bits and atoms. In *Proceedings of Computer and Human Interaction,* pages 234–241, 1997.

[12] I.Reichenbach and J.Masters. Children's use of expressive and contextual cues in judgements of emotions. *Child Development,* 54:102–141, 1983.

[13] J.Nielsen and H.H.Lund. Spiking neural building block robot with hebbian learning. In *Proceedings of Intelligent Robots and Systems,* pages 1363–1369,2003.

[14] J.Zhang. The nature of external representations in problem solving. *Cognitive Science,* 21 (2): 179–217, 1997.

[15] J.Zhang and D.A.Norman. Representations in distributed cognitive tasks. *Cognitive Science,* 18:87–122, 1994.

[16] M.W.Jørgensen, E.H.Ostergaard, and H.H.Lund. Modular atron: Modules for a self-reconrigurable robot. In *Proceedings ofInternational Conference on Intelligent Robots and Systems,* page to appear, 2004.

[17] N.Shadbolt. Ambient intelligence. *IEEE Intelligent Systems,* pages 2–3, 2003.

[18] P.Ekman. *Basic Emotions,* chapter An Argument for Basic Emotions, pages 169–200. Lawrence Erlbaum, 1992.

[19] P.Marti and H.H.Lund. Emotional constructions in ambient intelligence environments. *Intelligenza Artificiale,* 1(1):22–27, 2004.

[20] R.Pfeifer and C.Scheier. *Understanding Intelligence.* MIT Press, 1999.

[21] R.R.McCrae and O.P.John. An introduction to the five-factor model and its applications. *Journal of Personality,* 60:175–215, 1992.

[22] T.McNerney. *Tangible Programming Bricks: An Approach to Making Programming Accessible to Everyone.* Master's Thesis, MIT Media Lab, Cambridge, MA, 2000.

Chapter 4

VIDEO AND RADIO ATTRIBUTES EXTRACTION FOR HETEROGENEOUS LOCATION ESTIMATION

A Context-based Ambient Intelligence Architecture

S.Piva, R.Singh, M.Gandetto and C.S.Regazzoni
Department of Biophysical and Electronical Engineering, University of Genova, Italy
piva@dibe.unige.it

Keywords: AmI architecture, context awareness, multimodal Location feature extraction, radio-based location, video-based location

1. Introduction

Thanks to advances in a range of heterogeneous fields, from video analysis and understanding to software agents, from multi-sensor context assessment to pervasive communications, research on Ambient Intelligence systems is making significant progress towards the implementation of Smart Spaces (SS) where users are provided with an ensemble of ubiquitous virtual services [1][2][3][4]. ISTAG (Information Society Technology Advisory Group) gives a more formal definition of Ambient Intelligence (AmI) that points out how it should provide technologies to surround users with intelligent sensors and interfaces and to support human interactions [5]. While the specific services being provided can vary significantly according to the application domain, a common requirement for many AmI systems is that of supporting their users with guidance on available services, safety warnings and navigation aids. Providing effective assistance to users complex scenarios requires giving to the system a certain degree of awareness of the user's general preferences, current activities and real time condition [2]. In a word the system needs knowledge about the context in which user and system itself are acting. Context perception and understanding involves all the issues related to intelligent sensors and the management of heterogeneous information sources. As a human being perceives through his senses a huge amount of information and exploits them to decide for his actions, an artificial intelligent system based on contextual sensing must be provided with the appropriate sensors set and with the not easily achievable ability to extract interesting data at a higher abstraction level [6]. The central concept of context awareness

represents the possibility for the system of biasing itself and its reactions to the environment [7][8]. A key role among many useful features is represented by the detection of the user position inside the monitored environment because many applications are location dependent. The successful location of objects of potential interest and their identification are still open issues in the state of the art; anyhow different positioning approaches are available based on: Video [9] and Radio Signals [10]. The wide diffusion of radio devices on the market has given the opportunity to improve the functionality of SSs as most of the moving devices are equipped with such systems. The identification of users is not a problem for Radio-Based Systems (RBS) while is a challenge for Video-Based ones (VBS). The easier implementation of location tasks for RBSs has the price of less accurate localization data measurements. A tradeoff solution is then a system able to integrate heterogeneous sensors to perform localization exploiting the features of these two systems. Pursuing this aim, an architecture exploiting CCD-Video Cameras and 802.11 Wireless LAN positioning methodologies is presented; in particular the paper specifically addresses one of the first step towards fusion that is, according to a popular Data Fusion model [10], Data Alignment. It is easy to understand that the development of such a system is challenged by a large number of open problems, not last the design of an architecture assembling the fundamental functions of a Smart Space: Perception, Analysis, Decision, and Action (including Communication with the user). The aim is then the one to define a general structure able to take into account the main issues related to the management of the many different heterogeneous technologies characterizing an Ambient Intelligence environment. First of all in this work we propose a logical architecture we think has the generality and the completeness to allow the analysis of many of the possible problems raised by a Smart Space. An interesting inspiration point to deal with this kind of issue is to study the principles at the bases of the human brain working. A good inspiration can be found in [11] where a model for self consciousness is reported and motivated on neuro-physiological bases. This chapter is organized as follows: after this introduction, section 2 reports the most known ambient intelligence system implementations in the state of the art; section 3 defines the main issues and functionalities characterizing an ambient intelligence system; the proposed architecture and its inspiring model is described in section 4 while some concepts about the central topic of context awareness can be found in section 5. Some results for the location sensing task are depicted in section 6 and finally in 7 we draw some conclusions about the work.

2. Related work

Ambient intelligence is at the moment an open research field, still not bounded for what concerns topics of interest and related issues. In addition, systems which would be addressed as commercially appealing by the market are still developed at prototypal level. Technology is not enough mature to consider Ambient Intelligence a reality for users or a commercial product: many researches often deal with a single aspect (or a few) of a whole complex system. Although this, a certain number of integrated projects are in progress to show

the potentiality of this technology in actual test beds. Among the others, it is mandatory to remember the first Ambient Intelligence system presented by Trivedi, Huang e Mikic. It uses several cameras and microphones to acquire information on the surrounding environment [6]. According to concepts developed of the Oxygen project at M.I.T. (Massachusetts Institute of Technology), in next future computational power will be freely available everywhere and it will not be necessary to carry computers or personal communication terminals to access it [12]. More recently at the Artificial Intelligence Lab of the M.I.T. an Intelligent Room has been designed and developed. Research has been focused on natural and reliable vocal interfaces, context-specific user-transparent system reactions, dynamic resource allocation and natural cooperation among different contiguous environments [13][14]. On the private companies side, a strong interest in this research field can be found in the Dutch Philips where one of the most important examples of this kind is represented by the Phenom project [15]. Phenom is a home-oriented long-term research project, which aims at creating an environment that shows a context aware behaviour. At the Georgia Institute of Technology a long term research group is developing a project called The Aware Home Research Initiative (AHRI). It is an interdisciplinary research endeavour aimed at addressing the fundamental technical, design, and social challenges presented by the creation of context aware home environment [16][17]. Because of the topical argument, several works are still ongoing, such as a project funded by the Dutch Ministry of Economic Affairs: Ambience. The project involves academic partners as well as research divisions of private companies. The goal of the Ambience project is to jointly create networked context aware environments [18]. Another important work aiming at defining and realizing a complete and complex test bed is represented by VICOM (Virtual Immersive COMmunication) [19]. This Italian three-year project involves many academic and private partners and is developing strategies for context data extraction and management as well as multiple sensor systems and ad-hoc networking communication technologies [20]. To expand the vision about the related ongoing researches on specific topics the reader can also refer to [21][22][23] and for the central issue explored in this paper, context extraction, good sources can be found in the works by Neerincx and Streefkerk [24], Caarls and Persa for a multisensor fusion-based data extraction [25]. An example of association of mobility and context awareness can be found in [26]. Many publications on these topics are results of the work of Prof. James Crowley of the Institute National Polytechnique de Grenoble who directs the Prima project [27] about perception and integration for smart spaces [28][29]. Another specific reference for context based system design is finally represented by [30].

3. Main tasks of Ambient Intelligence systems

In Figure 4.1 the idea of the intelligent environment surrounding the user is depicted. In the user-centered approach the system is designed to be the less intrusive possible: the idea is the one to allow the use and the enjoyment of additional functionalities and facilities without the need of any training to learn

the way to interact with new technologies. The design has to start from the user needs and preferences and to give the smart space the ability to automatically adapt to them. The functionalities cooperating to this aim are something that tries to resemble an intelligent behavior in which a sensing task has the duty to collect data about the environment, data and information that becoming the input of the analysis task are processed to extract useful and concise higher level metadata the decision logics employs to choose the best action to undertake relatively to its metrics. Then the chosen action is realized by the action/communication oriented smart space function in order to apply an influence on the user itself. Consequences of these actions are again captured by the sensing tasks to close the loop and to judge the result of the processed in optimizing the interaction between the system and the actors interacting with its domain. In this work we go deeper inside the sensing task to explore a method to associate different techniques in obtaining contextual data, in particular for what concerns the position of the interacting objects.

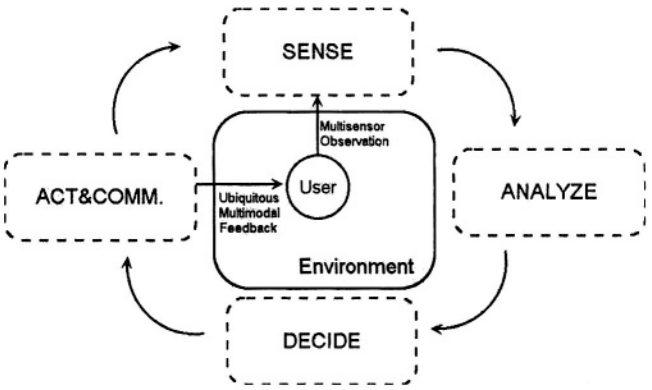


Figure 4.1. User centered ambient intelligence closed loop

4. Architecture design

4.1 Inspiration

Dealing with the complex issues related to Ambient Intelligence systems rises the need for a coherent and complete to allow a proper control on designing actions. In order to validate the design of a general structure to manage the main issues related to a Smart Space, our approach is the one to take inspiration from a theoretical human conscience model. The desired inspiration can be found in the work of Antonio Damasio [11] where a model for brain and conscious reasoning is described and motivated on neuro-physiological bases. This approach results particularly useful in the design of a complex distributed systems characterized by high-level analysis and interaction capabilities. Systems of this kind needs

to be aware of the behavior of all objects acting in their scope of sensing as well as of its own components' status and reactions. In this sense, awareness of the context, in which the system acts, is not enough; the system has also in some way to be conscious of its own internal state. In this section we thus introduce in details the cited model in order to use it as the root for our architecture design. In the description of Damasio, the acquisition and the behavior of what we call self consciousness comes from the interaction of two components: the organism and the objects and in terms of the relationships they hold in the course of their natural interactions. For our purposes the organism in question is the AmI system whereas the object is any entity that gets to be known by the system; the relationships between organism and object are the contents of the knowledge we call consciousness. Seen in this perspective, consciousness consists of constructing knowledge about two facts:

- the organism is involved in relating to some object;
- perceived objects cause changes in the organism [11].

While the human perceiving system changes dutifully at the mercy of the objects it interacts with, a number of brain regions whose job is to regulate the life process does not change at all in terms of the kind of object they represent. The degree of change occurring in the object –the body– is quite small. This is because only a narrow range of body states is compatible with life and the organism is genetically designed to maintain that narrow range and equipped to seek it. The body's internal state must be relatively stable by comparison to the environment surrounding it. So the deep roots for the 'self' are to be found in the ensemble of brain devices that continuously and non-consciously maintain the body state within the narrow range and relative stability required for survival. Following the Damasio model, the state of activity within the ensemble of such devices is defined Proto-Self, whereas the "non-conscious" part of self is represented by the Core self and Autobiographical Self. The following are the definitions of the three components concurring to form the human consciousness:

The Proto-Self is an interconnected and temporarily coherent collection of neural patterns which represent the state of the organism, moment by moment, at multiple levels of the brain. We are not conscious of the proto-self.

The Autobiographical Self is a conscious part of self, based on autobiographical memory which is constituted by implicit memories of multiple instances of individual experience of the past and of the anticipated future. This memory grows continuously with life experience but can be partly remodeled to reflect new experiences.

The Core Self: is generated for any object that provokes the core-consciousness mechanism that is to say the continuous environment awareness due to the analysis of stimulating objects. Because of the permanent availability of provoking objects, it is continuously generated and thus appears continuous in time. The mechanism of Core Self requires the presence of Proto-Self. Core Self can be triggered by any object. The mechanism of production of Core Self undergoes minimal across a lifetime. We are conscious of the Core Self.

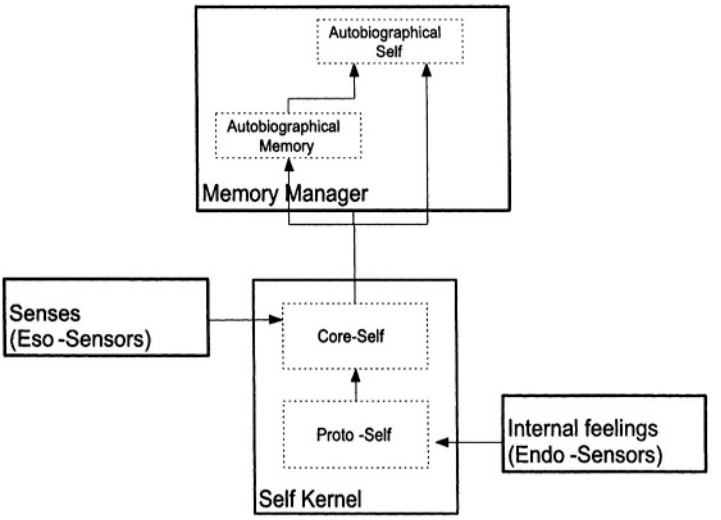


Figure 4.2. Damasio's Human Conscience Model. The described components are depicted in the diagram along with their connections. The Proto-Self and Core-Self are linked to senses and to internal organs by means of nerves connections and electric messages.

In Figure 4.2 Damasio's model is depicted, as it can be seen Proto Self constitutes the basis for the Core Self whereas the Autobiographical Self has got inferences coming from the Core Self and the Autobiographical Memory. This means the Autobiographical Self, that can be seen as the short term memory, is continuously composed by recalled elements coming from the long term memory (Autobiographical Memory).

4.2 Mapping the Model into an AmI Architecture

As previously stated our aim is to try and exploit the abstract concepts proposed by Damasio and translate them in the architecture of an artificial system able to be aware of the context it is working in. Proto-Self functionality can be realized through a system's internal state context analysis: the non-conscious human monitoring on the inner parts of the organism can be translated into the use of internal sensors. In both cases the aim to control critical variables and to analyze the relationship between environmental (external) events and internal state conditions. Core-Self is instead represented in its artificial intelligence counterpart, as the physical context representation, namely the collection and association of the continuous observation data coming from the senses (for the human brain) or the external sensors (for what concerns a context awareness based system). Autobiographical self, based on the use of autobiographical memory represents the conscious part of self. The two concepts are strictly

connected and can be seen as the short-term memory and the long-term memory respectively in the model of the human brain. They both constitute the idea of growing experience that can be implemented with associative techniques working on a knowledge base.

4.3 Artificial Sensing

Sensors own a key role in the definition and in the input of both Proto-self and Core-self: in an intelligent system they represent the bridge between the 'brain', namely the intelligent core of the organism, and the world, being it internal or external reality. Sensors (or receptors) are used by the system to keep contact to the interesting data and variables of the working environment (External World) as well as of its own internal state (Internal World). Considering this kind of distinction we can classify these devices or software agents into two groups that do not differ for technological aspects but for the aim of their observation, that is to say their observed domain. This means we can distinguish between Endo-receptors and Eso-receptors and associate them respectively to Proto-self and Core-Self (Figure 4.2). In particular, as suggested by their name, Endo-receptors are devoted to the observation of the internal state of the system: that is to say devices fit for analyzing internal components or variables proper of devices making part of the whole organism (system). Instances of sensors belonging to this class are: computational units (i.e.: Desktop PCs present in the environment available for users), devices' status sensors, thermal sensors, safety-oriented sensors (smoke, gas, fire, water infiltration, etc.), lighting sensors and so on. Endo-receptors are what is needed to realize the concept of the Proto-Self, as described in the introductory part. With Eso-receptors we refer to all the devices used by the structure to keep track of the events occurring in the observed domain and to collect data about the target of the analysis, being humans or other external interacting objects. Eso-receptors are the counterpart of the human senses, in this category fall sensors such as video sensors (working in visible or infra-red wave length fields), radio sensors (i.e. WLAN, Aps, BlueTooth, Global Positioning Systems (GPS)), standard or directional microphones, weight sensors, fingerprints readers, electro-magnetic waves emission scanners, photoelectric cells, etc. The focus of this work on external sensing tasks will be described in next sections.

4.4 Proposed structure

The complete structure of an AmI system here proposed (Figure 4.3) is defined in terms of interconnected logical modules. Each module implements a particular functionality such as sensing (i.e.: Endo/Eso receptors), analyzing (i.e.: Context Manager and Memory Manager), deciding (i.e.: Decision Manager) and acting (i.e.: Actuators and Multimode Communicator). In this sense, the system steps through the aforementioned functionalities with inferences on an Internal World (i.e.: the "artificial organism", the system itself) and on an External World (i.e.: the "objects", users). The latter environment represents those elements not under the direct control of the system. This means the system

is aware of the objects interacting in its domain, but it cannot apply a direct inference on them. For example, in the realized test-bed, we apply the interaction towards the external world (the users) by adapting message communication.

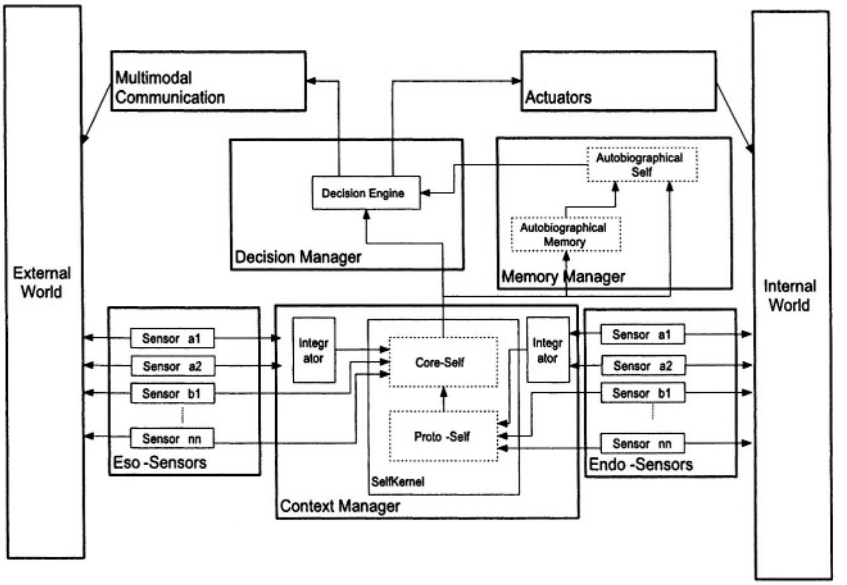


Figure 4.3. The proposed AmI logical architecture. Internal and External World represent the system itself and the objects and users interacting with the Smart Space, respectively. System adaptivity is explicitly expressed in the Decision Manager and in the Multimodal Communication modules

The only bridge keeping a connection between the intelligence of the system and the external world is expressed, as previously stated, by the pro-active functionalities it is provided with. Referring to the model, this is what represents the source of the continuous stimuli received by the Core-self. On the other hand, the Internal World is every thing the system can directly control: physical devices, internal software parameters but also physical components the system can apply its inference on. As it can be seen in Figure 4.3 the core part of the architecture that will be further described in next paragraph, takes origin from the Self Kernel sub module introduced in the description of Damasio's model. In particular Proto-self and Core-self are encapsulated in a Context Manager (CM) devoted to the analysis of heterogeneous data and to the generation of contextual information. The ensemble of internal and external state information is encoded by the Context Manager a hierarchical representation, where a super-state (e.g. ARRIVAL, meaning a new moving object entering the scene) gives the highest-level summarization of the context, while lower-level attributes

Table 4.1. Hierarchical representation of the context

Level		Description
1	Event level	super-states
2	Behaviour level	trajectories of users
3	Object level	identity of users
4	Feature level	number and position of users, system components status
5	Signal level	raw, non processed data

describe the position and behavior of individual people, objects and resources [31]. The reference hierarchy is summarized in Table 4.1.

In such a structure, lower levels contribute to the estimation of the higher ones, a super state (an event) can be evaluated by considering objects and related features(i.e., the object's position). Event and Behaviour levels define the context in which objects and users act, whereas the remaining levels contribute to the definition of more specific and personalized description for objects. Autobiographical Self with Autobiographical Memory, respectively embody system's long and short-term memory; they constitute the Memory Manager module (MM) devoted to the storage under the form of symbolic metadata of events produced by Context Manager. In addition it forms a Knowledge–Base for higher level modules. In an artificial system this concepts represents a static memory in which all the information needed to take decisions are stored and a buffer keeping trace of the instantaneous variables used to manage data processing. The Decision Manager depicted in the diagram uses the context awareness represented by the Self Kernel output to make the system reacting and have its influence on the AmI domain, as either internal or external world. To provide its results, the decision logic also relies on the direct observations of the receptors and on the past data provided by the Memory Manager. The decision can affect the Internal and External Worlds in different ways:

- Internal world: directly through some physical or software Actuator Modules (i.e.: Actuators in the diagram) such as mechanical tools which close or open windows and doors, as well as thermostatic devices affecting the rooms' temperature or in general controlling and modifying parts of the system etc.
- External world: through Multimodal Communication channels: the decision concerns the choice of the proper information to be delivered and the choice of the best communication channel to address the user.

In the scope of this work we further describe in the following sections the issues related to sensing tasks, particularly dealing with video and radio-based location sensors.

5. Context aware systems

The ability of an Ambient Intelligence system to adapt its behavior to the peculiar features of the user and of the environment, as previously introduced, is related to sensing and to context information extraction. But what do we

mean with context? To give a formal definition, we can say contextual information can be defined as an ordered multilevel set of declarative information concerning events occurring both within the sensing domain of a Smart Space (SS) and within the communication and actions domains of the SS itself. Relations among events are also included (explicitly or implicitly) within contextual information [32]. An event can be defined as the occurrence of some fact that can be perceived by or be communicated to the SS; an event is characterized by attributes that basically answer questions about where (position) and when (time) the event occurred. Other attributes involve what (core) consists the event of, who (identity) is involved in the event, and possibly why (reason) the event occurred [33]. Events can be used to represent any information that can characterize the situation of an interacting user as well as of a Smart Space component, i.e. an entity. An entity can be a person, a place, or an object that is relevant to the interaction between a user and an application. The user and the SS parts themselves are entities. The multilevel nature of contextual information is related to the possibility of detecting and representing events at multiple abstraction levels. Context-awareness refers to the property of a SS to internally represent in terms of events the state of its users, of their surroundings and of SS parts, as well as to be provided with rules that make it possible to adapt its behaviour accordingly. Therefore, in a context aware SS, contextual information can be either used by itself, i.e. it can have its own value for the interacting user, or it can be used to select which services can be provided to the user and which interaction modalities are more suited to let him/her access such services. In brief, an estimate of the user context can be used to optimize the adaptation and personalization process in such a way to maximize the service value. Pascoe introduced a set of four context-aware capabilities that applications can support [34]:

- Contextual sensing: a system detects the context and simply presents it to the user, augmenting the user's sensory system.
- Contextual adaptation: a system uses the context to adapt its behaviour instead of providing a uniform interface in all situations.
- Contextual resource discovery: a system can locate and use resources which share part or all of its context.
- Contextual augmentation: a system augments the environment with additional information, associating digital data with the current context.

This distinction results particularly useful in the design of a complex distributed systems characterized by a high-level analysis and interaction capabilities. In the example we present we particularly take into account capability of Context Adaptation with the consideration that many user oriented application are dependent upon his location inside the monitored environment.

5.1 Location feature

As previously stated in this section we go inside the sensing functionalities of a Smart Space and choose to deal with the issues related to context aware

information of the active objects, in particular its positioning, into the system domain. So which are those context features, that give the maximum information related to the object, should be used in the Ambient Intelligence system? What are those context aware information that are peculiar to the object, how it can benefit the AmI functional capability and which are those inferences that can be drawn based on selected context information, are the general questions that a system designer has to come across while designing the efficient AmI system. While looking for these answers, it is but obvious to look forward towards location as a context feature because firstly it gives the particular information about the object's position, Secondly, almost all the location sensing devices momentarily identify the object, thirdly, based on the location information other inferences such as object's intention, its mood, orientation, general behavior, track, identity and many important contextual data can be extracted. The location is defined as the spatial position of the object of interest in the known environment. The spatial position is given with respect to some particular reference position. This reference position is function of those sensor devices which momentarily identifies and extract the location information. Location data has been a very important information in navigation, military purposes, exploring earth, finding oneself in an unknown environment, in crime detection and prevention, controlling mentally challenge people's activity, civil applications, controlling the activity in the museum and nation's important security places etc.. Due to the need of the automatic services and heavy user's demand for location based services, and the presence of multi- wireless devices in the market, the research to implement and provide easy location services is continuously going on. The common reason of such interest is that automatically extracted knowledge of position in space and time of objects in an environment is the basis of multiple possible forms of intelligent cooperation with the object themselves, varying from pure perception, to active interaction, to personalized remote or short range communications. Having realised that knowledge of location is an important context aware information in the AmI system, we are trying to explore the possibility of combing and using multi-location sensors existing in the same environment. In order to validate the possibility of fusing heterogeneous sensors, we have incorporated two sensors namely, Video and Radio based location system. The idea is to extract the video and radio attributes and combine them efficiently to obtain as much inference as possible. As a step towards this, in this chapter, the attributes extraction methodologies, i.e. position, for the above two mentioned sensors are explored and described below. The effort is in the direction of proving the fact that the above two sensors can be combined and aligned to obtain the context aware information such as identity and accurate position.

5.2 The formalism

The formalism here-in-after used to describe the logical functional overview of the proposed systems assumes that a set of heterogeneous sensors $S = \bar{S}^c : c = 1, ...N_s$ is divided in, N_s, different classes

$\bar{S}^c = S_i^c : i = 1, ...N_{\bar{S}^c}$ where $N_{\bar{S}^c}$ is equal to the number of sensors in class (c^{th}) . Each sensor is directly connected to a dedicated Computational Units (i.e.: CU) belonging to the set $U = u_l : l = 1, ..., N_\mu$ with N_μ equal to the total number of corresponding sensors. Each CU acquires data providing Object Reports (OR) $\bar{r}_{i,m}^c(k)$ for each object m found at time k from i^{th} sensors in c^{th} class. OR is represented as a multidimensional vector composed by different features related to the detected object:

$$\bar{r}_{i,m}^c(k) = [\bar{f}_1^i(k), \ldots, \bar{f}_{N_r}^i(k)] \tag{4.1}$$

with N_r the total number of features $\{hatf$ in the report. For each detected physical object tracks are instantiated and updated:

$$T_m(k) = \hat{r}_m(K-k) : k = 0, ..., K \tag{4.2}$$

where K is the current time, m = detected object. Tracks are sequences of estimated reports $\hat{r}_m(i)$ derived from integration of heterogeneous ORs:

$$\hat{r}_{i,m}(k) = [\hat{f}_1^i(k), \ldots, \hat{f}_{N_r}^i(k)] \tag{4.3}$$

In the presented case, only two classes of Sensors have been included in the

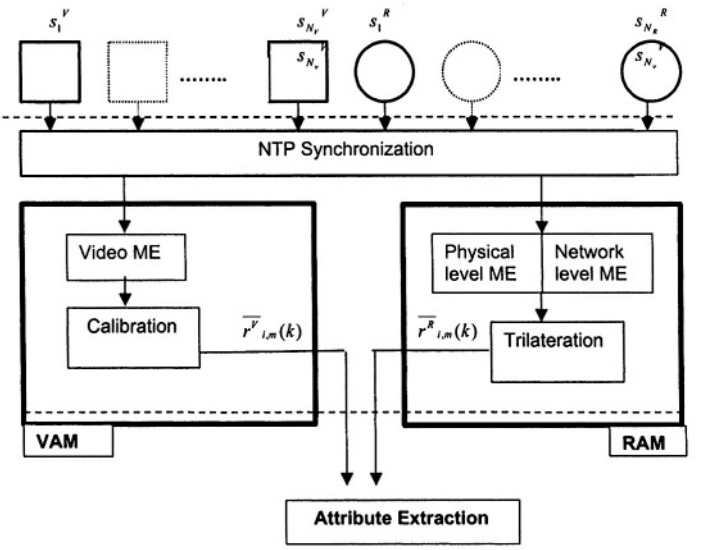


Figure 4.4. Functional architecture of the extraction strategy

architecture: Static CCD Video Cameras and 802.11 WLAN Base Stations (i.e.: classes c = *R, V).* In particular, a Video Analisys Module (i.e.: VAM) takes care of extracting metadata from video sources whereas the Radio Analisys Module (i.e.: RAM) does the same with Base Stations. Output Object Reports are respectively addressed as Video Object Reports (VOR) and Radio Object Reports (ROR)$\bar{r}_{i,m}^{c=V,R}(k)$ shown in Figure 4.4.

5.3 Alignment and Extraction of Video and Radio Object Reports

Introduction. As it can be seen from Figure 4.4, dedicated sub-modules (i.e.: Video ME, Physical Level ME and Network Level ME) are specifically devoted to the extraction of metadata that is coded under the form of an Object Report:

$$\bar{r}_{i,m}^{c}(k) = [\bar{p}^{i}(k), \bar{id}^{i}(k), \bar{c}^{i}(k)] \tag{4.4}$$

where $\bar{p}^{i}(k), \bar{id}^{i}(k), \bar{c}^{i}(k)$, respectively indicates position, id and class (e.g.: pedestrian, vehicle, others) of detected objects moving in the monitored environment.

Video object extraction. Video Objects Reports $\bar{r}_{i,m}^{V}(k)$ are evaluated by Video Metadata Extractors (i.e.: VME) at each timestamp . VME takes as input raw video frames from synchronized grabbers; typically, chain of logical tasks can be assembled in order to process Video data [9], the first step is however a Dynamic Change Detection (see Figure 4.2 right) performing the difference between the current image and a reference one (i.e. background). Each moving area (called Blob) detected in the scene is bounded by a rectangle to which a numerical label is assigned (Figure 4.2 left). Thanks to the detection of temporal correspondences among bounding boxes, a graph-based temporal representation of the dynamics of the image primitives can be built. The core part of such systems is however represented by tracker algorithm, that outputs to the Calibration submodule OR ($\bar{r}_{i,m}^{V}(k)$) with features:

$$\begin{aligned} \bar{f}_1^i(k) &= \bar{p}_1^i(k) = [x_I, y_I] \\ \bar{f}_2^i(k) &= \bar{id}_1^i(k) = [id] \\ \bar{f}_3^i(k) &= \bar{c}_1^i(k) = [c] \end{aligned} \tag{4.5}$$

where (x_I, y_I) are the coordinates (in pixels) of the center of mass in the Image Plane for the m-th object (i.e.: blob) at time k detected by i^{th} sensor whereas the scalars id and c respectively indicates the tracked id (progressive integer number, e.g.: 1,2..) and class of the object (integer number, e.g.: 1=human, 2=vehicle, 3=others).

Video Object Spatial Alignment. Spatial alignment for Video ORs is achieved through Camera Calibration. Camera calibration [40][41] is the pro-

cess by which optical and geometric features of Video Cameras can be determined (Figure 4.5 shows detection of features). Generally, these features are addressed as intrinsic and extrinsic parameters and they allow estimation of a correspondence between coordinates in the Image Plane (x_I, y_I) and in the $3D$ Real World Space (x_W, y_W, z_W). After the 3-D conversion the last step is represented by the projection on $2D$ Map Plane (x_M, y_M). Various methods

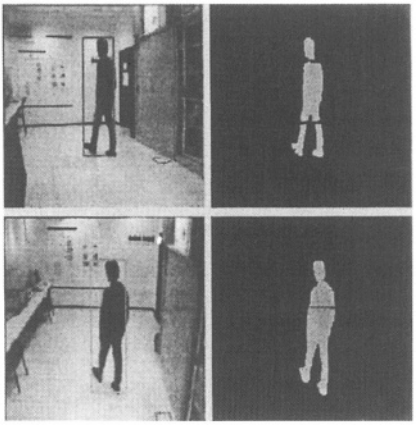


Figure 4.5. Right: Binary Change Detection images showing moving pixels in the current scene. Left: Two views for the available Video Cameras, moving object is highlighted by Bounding Boxes.

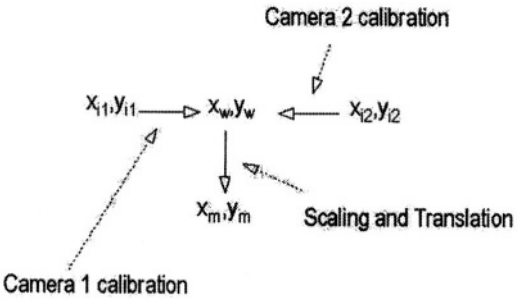


Figure 4.6. Joint calibration methodology.

have been proposed to perform calibration: some uses non-linear optimisation techniques [42], others systems of linear equations. Camera calibration we use

is based on classic Tsai method [41]. In the presented system, all video sensors have been calibrated with a common calibration strategy. In Figure 4.6, the chosen approach is outlined. First of all cameras are calibrated on reference images in a unique map then a common reference point has to be found in order to make the system able to switch between the different reference systems. Origin of World Space (x_W^0, y_W^0, z_W^0) represents a good choice because it is common to the all the cameras. The alignment algorithm can be decomposed in the following steps:

1 Image Coordinates (x_I, y_I) are converted in World Coordinates (x_W, y_W, z_W) through calibration.

2 (x_W, y_W, z_W) are converted in Map coordinated (x_M, y_M) with translation and scaling transformations.

3 OR is rewritten as :

$$\bar{r}_{i,m}^V(k) = [\bar{p}^i(k), \bar{id}^i(k), \bar{c}^i(k)] = [x_M, y_M, id, c] \tag{4.6}$$

It is important to note that, in this case,id and c features which should be stationary (i.e.: identity and class of an object do not change over time) values are subject to variations over time due to induced errors in the Change Detection and Tracking steps.

Radio object extraction. The Radio Objects can be defined as those objects which possess electronic wireless communication facilities (e.g.: Bluetooth or WLAN cards). The wireless communication network as a part of the video surveillance system shares the information related to the objects. The Radio Objects Reports (i.e. ROR)$\bar{r}_{i,m}^R(k)$ are evaluated by receiving the signals sent by objects device to three base stations (BS). BS as a part of the wireless system are able to communicate with the radio object via receivers and they recognize RORs via their network ID. The system is based on a path-loss model of the signal power transmitted from/to APs and receivers. The observed power is converted into distances using path-loss equations, eq.8 [42][45].

$$S = S_o - 10\,\alpha\, log\frac{d}{d_o} \tag{4.7}$$

Where S is the received power in dB, p_o is the received power at a reference distance (d=1 meter), d is the distance between transmitter and receiver and a is the path-loss exponent. Unfortunately, due to the presence of multi-path fading and noise interference in the environment, the received power is not only dependent on the path loss. Therefore, Equation 4.7 can be represented as Equation 4.8 where the observed power $S + N$, due to fading and path-loss is

$$(S + N) = S_o - 10\,\alpha\, log\frac{d}{d_o} + X_\sigma = S + X_\sigma \tag{4.8}$$

Where random variable X_σ represents the medium-scale fading in the channel and is typically reported to be Gaussian random variable with zero-mean (in dB) and variance σ^2, also represented as $N(0, \sigma)$ [45][48]. The Probability Density Function (p.d.f.) of the received power in eq. (9) is $N(\bar{S}, \sigma)$ with $\bar{S}$ mean and Standard Deviation, σ.

Having estimated *(S + N)* and X_σ, it is possible to compute the distance between transmitter and receiver using Equation 4.8. The distances obtained by the transceiver are tri-laterated [36] to estimate the position (X_M, Y_M) in the common Map Plane and to fill an OR following the policy applied to the video case:

$$\bar{r}^R_{i,m}(k) = [\bar{p}^i(k), \bar{id}^i(k), \bar{c}^i(k)] = [x_M, y_M, id, c] \tag{4.9}$$

In this case the unstable feature is expected to be position whereas identity and class are constant over time.

Radio Object Reports Spatial Alignment. Figure 4.7 shows the logical functioning architecture of the WLAN network for estimation of the position $\bar{p}^i(k)$ using received signal strength (RSS) features. The $RSS_{1,2,3}$ are the received signal strength of the signal at the object device end. The Path-Loss Equation 4.8 uses this information to evaluate the distance d_{123} that is considered to be the distance between the BS and the object. Given the problem of presence of multi-path fading and noise in received signal, and their negative effect on position accuracy, it is desired to enhance the accuracy in two steps:

- At signal level using Pre-Post Cursor Multi-path Mitigator [37].
- At feature level using Feature Function (FF) which is created in the offline phase of the spatial alignment [46].

Spatial alignment and projection in the common 2-D map is performed in two phases:

1 Offline phase

2 Online phase

The offline phase consists of signal strength data collection at several predefined positions in the test site. Based on the RSSobserved (i.e.: collected power strength) at known positions and RSStheoratical,(i.e. the theoretically computed power strength for known distances) the RSSmultipath+noise can be computed as the difference between the above two signal values. The ratio, δ, in Equation 4.10, is obtained for each known distance between transmitter and set of different position. The polynomial fit function, which is FF, is computed based on the collected signal measurements, where:

$$\delta = (RSS_{multipath+noise}/RSS_{observed}) \tag{4.10}$$

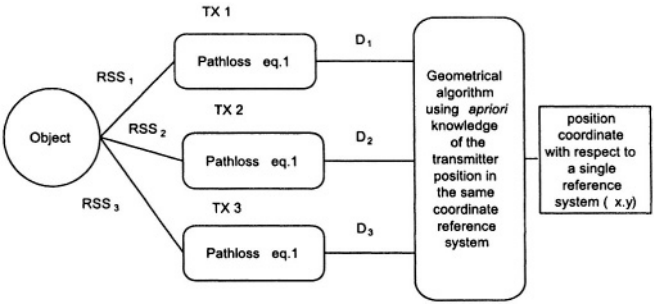


Figure 4.7. Logical functioning architecture of the WLAN Positioning network using RSS features.

The function FF is utilized in an online phase. During online phase, user's signal are collected by each transmitter and 2D position is computed as explained in last section. The details about position method using wlan 802.11b system can be found in [46]. In Figure 4.8, the view of the experimental site is shown. The BS are localized within the map and identified by dots. Three circles are centred in transmitters with radius equal to estimated transmitter distance (i.e.: d_{123}). The overlapping region of the three circles represent the most probable region where the target has to be located.

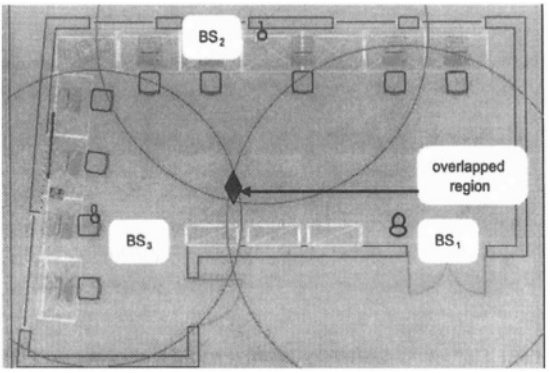


Figure 4.8. View of the experimental site with Base Stations and overlapping region.

6. Results

6.1 The environment

The environment in which the tests took place can be divided in two topologies:

- Experimental test bed from which video object reports have been extracted;
- Simulated test bed from which radio object reports have been extracted.

Once the two results set have been gathered, the heterogeneous location information are sent to a processing unit to start the alignment process.

6.2 Results for video object extraction

The following mock-up architecture has been set-up in the laboratory of Biophysical and Electronic Engineering Department at University of Genoa: two CCD-Video Cameras with 352x288 pixels resolution, 10 fps frame rate and partially overlapped fields of view are used in a $9x7$ meters room. The two cameras are connected to a processing unit which performs the video object extraction. In Figure 4.5 the output screenshots of the Video Metadata Extractor during the tests are reported for the two fields of view. In Figure 4.9 it is possible to see the video object position on the room map. Tracks come from the VME after calibration. As it can be seen two different tracks are available: the first one comes from camera one and the other from the second video sensor.

6.3 Results for radio object extraction

In this part of the result section, the simulation environment to examine the proposed radio object extraction system is described. The system has been developed and simulated in Mathworks Matlab 6.0 environment. The transmitter used in the simulator is using the CCK Coding and DQPSK modulation for the transmitted signal at 11Mbps and it satisfies all the requirements specified in [47]. The multipath fading present in the channel for indoor Line of Sight (LOS) is Rice distributed with AWGN and delay spread of 100 ns [48]. The effect of Doppler spread has been neglected as the terminal is considered to be almost static in indoor ambience. The room used for Video Experimental Trial is simulated. The parameter used in path-loss are n= 1,6-1,8 P0= -20 dBm ± 1 dB. The three transmitters' position are known on the map as shown in Figure 4.5. The signals from three transmitters are generated for a given positions, multipath fading level and noise effect, that is to say a random attenuation with fixed delays, is added to it. At the terminal end, the proposed method at signal and feature level is applied on the radio signals. The results in terms of relative positioning error is obtained as shown in Figure 4.10, moreover as in Figure 4.9, in Figure 4.12 the tracks of user position are plotted for the four different methodologies.

Figure 4.10 shows the error obtained in five simulated positions. The positions are estimated in four cases: without any mitigator (track called crude in

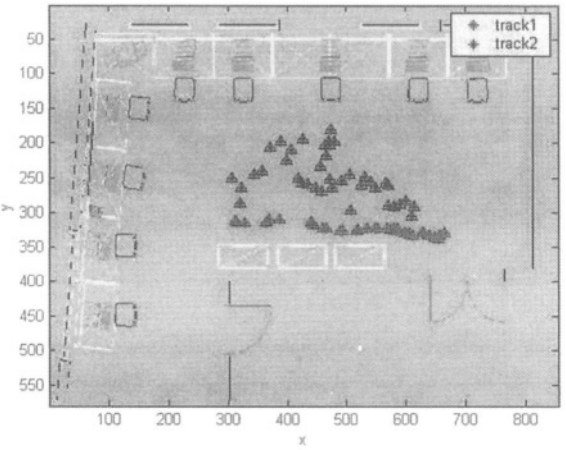


Figure 4.9. 2-D map of the site in which tests took place. Two tracks are aligned and projected for Video Object Reports.

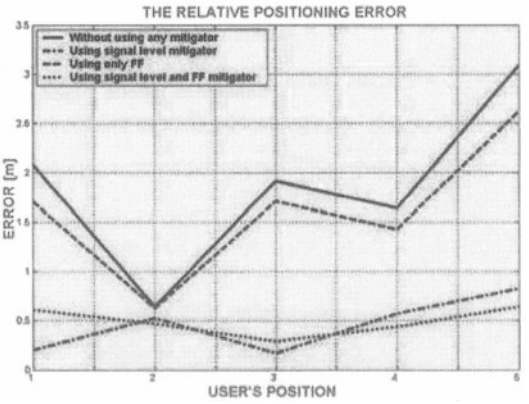


Figure 4.10. The simulation results after utilizing both signal level and FF.

the figure), using feature level mitigator (FF) (called features), using pre-post cursor signal level mitigator (called pre-post) and using both mitigator together (called feature+prepost). As it can be seen in Figure 4.10 that system without using any mitigation is affected by a big positioning error, whereas pre-post

cursor mitigation reduces this error to 33%. The FF algorithm is prominently improving the accuracy by 75%. The best positioning accuracy recovered is, however, with the use of both level of mitigator indicated by a 79% result.

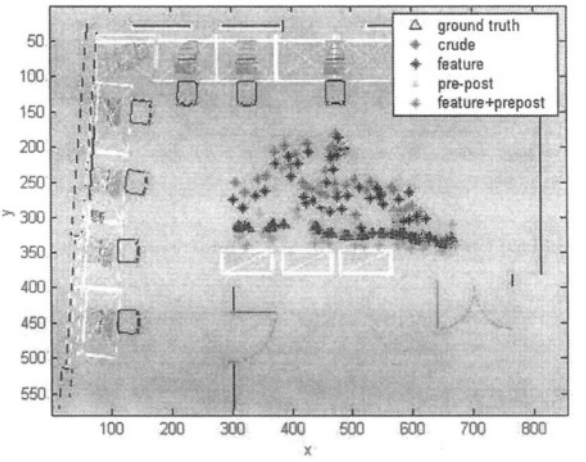


Figure 4.11. The track for Radio Object Reports

6.4 Alignment results

Presented qualitative results are a preliminary attempt to show how Radio and Video data can be aligned in a common ground plane in order to be fused. In addition, error in localization, especially in Radio sensor (in case of both mitigator) and Video sensor is encouraging and it will allow association between the heterogeneous tracks. In Figure 4.12 the results of two heterogeneous sensors are plotted together.

7. Conclusions

Ambient Environment provides the intelligent environment around the object. The efficiency of the AmI is highly dependent upon the context aware information. The context aware information is provided by the multiple sensors co-existing in the environment. The functionality of the AmI is structured into four main task. First, sensing task has the duty to collect data about the environment, data and information that becoming the input of the analysis task which are processed to extract useful and concise higher level metadata, the decision logics employs to choose the best action to undertake relatively to its metrics. Then the chosen action is realized by the action/communication

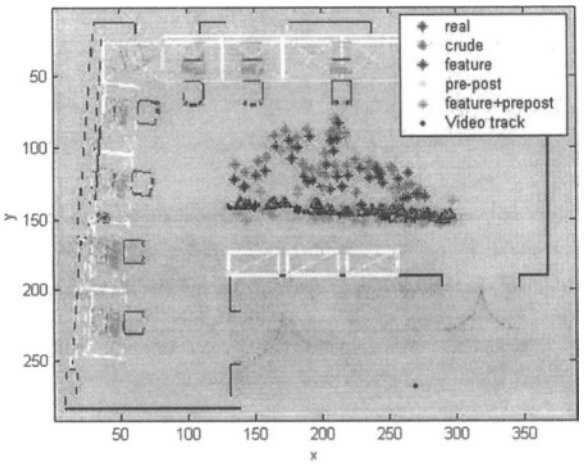


Figure 4.12. 2-D map of the site in which tests took place.

oriented smart space function in order to apply an influence on the user itself. As the context aware information is extracted through sensing task, the context aware system has been described deeply. The location information as a very important context aware information has been explained and described with special attention on the fusion aspect between multi-sensor. The multi-sensors in particular has been considered to be video and radio sensors. The attributes extraction in case of both the sensor has been described and explained in detail. Then based on the extracted feature along with the position information, task of alignment is performed. The results are explained that shows that based on the extracted location attributes it is positioning to do the association and estimation of the precise positioning. The effort is mainly on the shown proof that it is possible to align multi-positioning sensors by extracting the track obtained by each sensors. As a step towards their fusion, this provides the sufficient ground to efficiently fuse the multi-location sensors.

8. Acknowledgments

This work was performed under co-financing of the MIUR within the project FIRB-VICOM and the project Per2.

References

[1] D. Garlan, D. Siewiorek, A. Smailagic, and P. Steenkiste, "Project Aura: Toward Distraction-Free Pervasive Computing", IEEE Pervasive Computing,

April-June 2002.

[2] L. Marchesotti, C. Bonamico, C. Regazzoni, and F. Lavagetto, "Video Processing and Understanding Tools for Augmented Multisensor Perception and Mobile User Interaction in Smart Spaces", International Journal of Image and Graphics, 2004 (to appear).

[3] A. Nijholt, "Disappearing Computers, Social Actors and Embodied Agents", Proceedings of 2003 International Conference on CYBERWORLDS, Singapore, pp. 128-134, 2003.

[4] F. Vatalaro, "Dalle Telecomunicazioni alla Telepresenza Immersiva", Notiziario Tecnico Telecom Italia, anno 11, n°3, Dicembre 2002.

[5] ISTAG, "Scenarios for Ambient Intelligence in 2010". http://www.cordis.lu/istag.htm

[6] M. Trivedi, K. Huang and I. Mikic, "Intelligent Environments and Active Camera Networks", IEEE Transactions on Systems Man and Cybernetics, October 2000.

[7] VICOM project, "Virtual Immersive COMmunications", 2002-2005. http://www.vicom-project.it

[8] L. Marchesotti, L. Marcenaro and C. Regazzoni, "Heterogeneous Data Collection and Representation within a Distributed Smart Space Architecture", Proceedings of ACIVS 2002 (Advanced Concepts for Intelligent Vision Systems), Ghent, Belgium, 2002.

[9] L. Marcenaro, F. Oberti, G.L. Foresti and C.S. Regazzoni, "Distributed architectures and logical task decomposition in multimedia surveillance systems", Proceedings of the IEEE, Vol.89, N.10, Oct.. 2001, pp. 1355 -1367.

[10] E. Waltz and J. Llinas, "Multisensor data fusion" , ISBN 0-89006-277-3, 1990 Artech House, Inc.

[11] A. R. Damasio, "The Feeling of What Happens-Body, Emotion and the Making of Consciousness", Harvest Books, September 2000.

[12] http://oxygen.lcs.mit.edu/

[13] Brooks, R. A. with contributions from M. Coen, D. Dang, J. DeBonet, J. Kramer, T. Lozano-Perez, J. Mellor, P. Pook, C. Stauffer, L. Stein, M. Torrance and M. Wessler, "The Intelligent Room Project", Proceedings of the Second International Cognitive Technology Conference (CT'97), Aizu, Japan, August 1997.

[14] Lorigo, L.M., R.A. Brooks and W.E.L. Grimson, "Visually-Guided Obstacle Avoidance in Unstructured Environments", Proceedings of IROS '97, Grenoble, France, September 1997, pp. 373-379.

[15] http://www.research.philips.com/InformationCenter /Global/FArticleSummary.asp?lNodeId=712/

[16] http://www.cc.gatech.edu/fce/ahri/

[17] Kidd, et al. The Aware Home: A Living Laboratory for Ubiquitous Computing Research, Proceedings of the Second International Workshop on Cooperative Buildings, Position paper, October 1999.

[18] http://carol.wins.uva.nl/porta/ambience/index_en.html/

[19] http://vicom-project.it

[20] F. Vatalaro Dalle Telecomunicazioni alla Telepresenza Immersiva, Notiziario Tecnico Telecom Italia, anno 11, n°3, Dicembre 2002.

[21] http://www.ubiq.com/hypertext/weiser/UbiHome.html

[22] A. Pentland. Looking at people: Sensing for Ubiquitous and Wearable Computing, IEEE Trans. On PAMI, Vol. 22, n. 1, pp. 107-119, January 2000.

[23] R. Collins, A. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, and O. Hasegawa, A System for Video Surveillance and Monitoring, tech. report CMU-RI-TR-00-12, Robotics Institute, Carnegie Mellon University, May, 2000.

[24] M.A. Neerincx and J.W. Streefkerk, Interacting in Desktop and Mobile Context: Emotion, Trust and Task Performance, European Symposium on Ambient Intelligence, Eindhoven, Netherlands, November 2003.

[25] Jurjen Caarls, Pieter Jonker, Stelian Persa, Sensor Fusion for Augmented Reality, European Symposium on Ambient Intelligence, Eindhoven, Netherlands, November 2003.

[26] Heikki Ailisto, Ville Haataja, Vesa Kyllönen and Mikko Lindholm, Wearable Context Aware Terminal for Maintenance Personnel, European Symposium on Ambient Intelligence, Eindhoven, Netherlands, November 2003.

[27] http://www-prima.inrialpes.fr/Prima/

[28] J. L. Crowley, J. Coutaz, G. Rey and P. Reignier, "Perceptual Components for Context Aware Computing", UBICOMP 2002, International Conference on Ubiquitous Computing, Goteborg, Sweden, September 2002.

[29] J. L. Crowley, "Context Aware Observation of Human Activities", IEEE International Conference on Multimedia and Expo, ICME-2002, Lausanne, Aug 2002.

[30] Bauer, T. and Leake D., WordSieve: A Method for Real-Time Context Extraction. Modeling and Using Context: Proceedings of the Third International and Interdisciplinary Conference, Context 2001, Springer Verlag, 2001, pp. 30-44.

[31] L. Marchesotti, S. Piva, C.S. Regazzoni, "Structured Context Analysis Techniques in a Biologically Inspired Ambient Intelligence System", Proceedings of the IEEE on System Man and Cybernetics, Special Issue on Ambient Intelligence, 2004 (to appear).

[32] J. L. Crowley, "Context Aware Observation of Human Activities", Proceedings of the IEEE International Conference on Multimedia and Expo, ICME-2002, Lausanne, Aug 2002.

[33] L. Marchesotti, G. Scotti and C. Regazzoni "Issues in Multi-camera Dynamic Metadata Information Extraction and interpretation for Ambient Intelligence", Nato Asi 2003, NAREK center of Yerevan University, Tsakhkadzor, Armenia 18-29 August 2003

[34] J. Pascoe, "Adding Generic Contextual Capabilities to Wearable Computers", in Proceedings of 2nd International Symposium on Wearable Computers, October 1998, pp. 92-99.

[35] E. Waltz and J. Llinas, "Multisensor data fusion" , ISBN 0-89006-277-3, 1990 Artech House, Inc.

[36] Jeffrey Hightower and Gaetano Borriello, "Location Systems for Ubiquitous Computing," Computer, vol. 34, no. 8, pp. 57-66, IEEE Computer Society Press, Aug. 2001.

[37] K. Pahlavan and P. Krishnamurthy, Principles of Wireless Networks: A Unified Approach, Prentice Hall PTR, 2002.

[38] P Shan E. J. King "Cancel Multipath Interference In Spread Spectrum Communications" Wireless System Design, March 2001 Pg 49-52.

[39] Siddhartha Saha, Kamalika Chaudhuri, Dheeraj Sanghi, Pravin Bhagwat, "Location Determination of a Mobile Device Using IEEE 802.11b Access Point Signals," IEEE Wireless Communications and Networking Conference (WCNC) 2003 New Orleans, Louisiana, March 16-20, 2003.

[40] Tsai, Roger Y. (1986) "An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision," Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Miami Beach, FL, 1986, pp. 364-374.

[41] Tsai, Roger Y. (1987) "A Versatile CameraCalibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses," IEEE Journal of Robotics and Automation, Vol. RA-3, No. 4, August 1987, pp. 323-344.

[42] J.Renno, J. Orwell, G.A. Jones." Towards Plug-and-Play Visual Surveillance: Learning Tracking Models". ICIP02 (III: 453-456) September 2002 Rochester New York.

[43] L. Marcenaro, F. Oberti, G.L. Foresti and C.S. Regazzoni, "Distributed architectures and logical task decomposition in multimedia surveillance systems", Proceedings of the IEEE, Vol.89, N.10, Oct.. 2001, pp. 1355 -1367.

[44] Luca Marchesotti, Giuliano Scotti and Carlo Regazzoni, "Issues in multicamera dynamic metadata information extraction and interpretation for ambient intelligence", in press, Yerevan Armenia, NATO ASI 2003.

[45] B. Sklar, "Rayleigh Fading Channels in Mobile Digital Communication Systems Part1: Characterization," IEEE Communication Magazine, July 1997, page 90-100.

[46] R. Singh, M. Gandetto, M. Guainazzo, C.S. Regazzoni, "A Novel positioning system for static location estimation employing WLAN in indoor environment," accepted in the 15th IEEE Symposium on Personal, Indoor and Mobile Radio Communications PIMRC, IEEE, April. 2004.

[47] LAN MAN Standards Committee of the IEEE Computer Society: P802.11b, Supplement to IEEE Std 802.11-Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specification-Higher Speed

Physical Layer (PHY) Extension in the 2.4GHz band, IEEE Standard Department, 1999.

[48] G. L. Stüber, "Principle of Mobile Communication", Norwell, MA: Kluwer, 1996.

Chapter 5

DISTRIBUTED ACTIVE MULTICAMERA NETWORKS

A.Senior, A.Hampapur, L.Brown, Y-L.Tian, C-F. Shu and S. Pankanti
{aws,arunh,lisabr,yltlan,sharat,cfshu}@us.ibm.com
IBM, USA
Keywords: Video surveillance, tracking, detection, database, multiscale, multicamera.

1. Introduction

Ambient intelligence involves putting information processing devices into the world and delivering intelligently processed information to distributed information consumers —with the ultimate purpose of providing utility to human users. This process will see more and more devices embedded into everyday objects and into the environment (e.g. cars, buildings and street furniture). The aim of ambient intelligence is not to make processing power ambient, for there are economies of scale in putting as much processing as possible into dedicated farms, but to make the sensors and actuators on these devices ambient and to deliver the power of information processing ubiquitously. Efficiencies of communication necessitate the processing power and thus the intelligence itself being local to the sensors and actuators and thus equally ambient.

2. Sensing modalities

As myriads of small, specialized computing devices are deployed in the world, we see them becoming markedly different from the traditional computing paradigm of a processor box with invariable user interface components of a visual display screen, mouse and keyboard. While many specialist sensors will be developed — to detect biological agents, cosmic rays and the like, or perhaps to assist environmental regeneration by controlled release of nutrients or organisms- it is likely that a huge proportion of the devices that will be deployed will be for the purposes of interacting with humans. These sensors will be a mix of active —detecting communication directed at them by "users"— and passive — sensing activity that is taking place within range.

Vision, we argue, is a very important sensing modality for both kinds of sensors. Vision provides a rich source of information about the world- and each sensor is able to receive data from a relatively long range. Vision can tell

us about the shape of the world (from stereo and from a wide range of other cues), and is sufficient for identifying people (particularly face recognition, but also gait, ear shape and lip motion [2]). Vision can also capture a wide range of human communication (writing, facial expression, gesture, and body language, even lip-reading [9]). In interacting with people, vision is particularly important because it is the predominant modality through which most humans acquire information, and it is useful and in some cases necessary for machines to "see the world as we see it".

Sound is an important complementary sensing modality that is very rich source of information when sensing human activity, because of the importance of speech for human communication, and because of its omnidirectionality. As microphone arrays, beamforming and source separation techniques improve, sound signal aquisition is improving in range and quality.

An expanding host of other sensing modalities is available, sensing motion, position, orientation, electric, magnetic and gravitational fields, atmospheric chemicals, and biological agents.

The acquisition of this data provides a challenging fusion problem in creating, and visualizing a rich, multi-modal model of the world. However the complementary problem of filtering this torrent data down to extract the interesting facts that can be usefully acted upon or are worth storing, then delivering the information to mobile users, provides a greater, Ambient Intelligence, challenge

3. Vision for Ambient Intelligence

Because of the importance of vision, as a rich source of passively-available, informative data about the world, as discussed above, we concentrate on vision as a modality for sensing. This modality is particularly useful as a method of building up an understanding of the world for an ambient intelligent system, though it is naturally most powerful when combined with many other sensing modalities. The use of vision is also driven by the growing demand for, and feasibility of, practical systems for visual understanding of the world, particularly in the domain of visual surveillance.

Visual surveillance is an interesting domain inherently suited to ambient intelligence. Conventional surveillance systems have networks of cameras (now beginning to number in the thousands at large installations such as airports) distributed over a wide area. The video from these cameras has traditionally been centralized in a single control room where the video is recorded and observed by a small number of security guards. It is well known that security guards quickly lose alertness when facing banks of monitors displaying empty scenes. The potential for a relentless watcher of every single channel of video with perfect recall has driven the development of intelligent systems tracking objects in surveillance video. Increasingly there is also a demand to acquire such data from mobile platforms (say police cars) and deliver the intelligence so-gathered to a range of heterogenous, dispersed and often mobile devices. The vast quantity of data and the expense of cabling already encourages the distribution of the intelligence, putting the processing near the cameras, and

only broadcasting the relatively low-bandwidth information of interest to a central repository.

Having started to process surveillance video with computers, a whole range of further possibilies soon emerges. Several cameras in a given installation will be able to view the same area or at least the events in the view of one camera will correlate with the events seen by another — at the very least in the form of the same people or vehicles being seen more than once. It is clear that a richer understanding of the goings-on in a surveillance site can be achieved by integrating the data from multiple cameras. With multiple cameras comes a better, three-dimensional understanding of the world. People and vehicles can be tracked not just for short, unrelated time spans, but continuously over the entire time they are in the field of view of the cameras, and activity models can also predict their behavior when out of sight.

Such multi-camera systems increase the need for ambient intelligence as local processing systems need to communicate between themselves, sharing data from their different viewpoints, and further refining their representations before communicating to consumers of their information.

In the remainder of this chapter, we describe a number of the technical challenges involved in building up this rich world view through networks of cameras, as implemented and projected in the IBM Smart Surveillance System. In Section 4, we discuss the architectures for communication, followed by single camera object detection and tracking algorithms in Section 5.2. Section 6 describes object normalization for view-invariant reasoning about objects, and Section 7 describes multi-camera strategies for tracking. Section 8 describes work on active camera systems to acquire high-resolution data of objects tracked in static cameras. Finally Section 9 describes the delivery, storage and searching of the tracking data for secure, privacy-protecting, distributed access.

4. Architecture

The IBM Smart Surveillance system is a complete architecture for multi-camera, distributed surveillance video processing, comprising front-end image processing and camera control, local processing to integrate track information from nearby cameras, and a back-end database infrastructure that stores and redistributes the data. Client applications and browse stations access the processed data by issuing queries to the database system.

Figure 5.1 shows an overview of the complete system.

Real-time video processing in the form of tracking and object detection (Section 5) is carried out close to the camera, on conventional computers or perhaps by embedded processors or specialised boards. Encoding (and encryption) of video is also carried out on, or as close as possible to, the camera, minimizing bandwidth and wiring requirements. Optionally a privacy camera [10] may be used that preprocesses the video and transmits only privacy-protected video.

Integration operations that exploit local knowledge are also carried out close to the cameras. Such operations include track correspondence between multiple cameras, whether overlapping or close, and active camera control that requires a reliable fast feedback loop. Such systems are described in Sections 7 and 8.

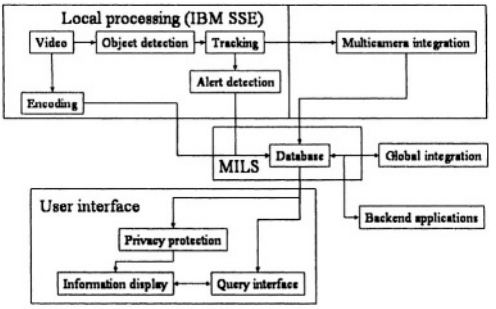


Figure 5.1. Architecture of the People Vision system, showing data flows between components.

Data from all of these processes are communicated to a global data repository in the form of a conventional, DB2 database for concise numerical and string data and IBM ContentManager for rich media objects, in particular the compressed video streams. This back-end system may be centralized or distributed. These commercial data management products handle all the complex data management tasks (such as back-up, security, expiration, distribution, scalability and fast indexing) that are not specific to video surveillance data. Digital data distribution throughout the system can be encrypted, with video using conventional (e.g. MPEG4) video compression and a flexible XML schema for track information and meta-data.

Client applications that may be registered autonomous programs (e.g. elevator controllers, fire alarms etc.) or user-controlled browse stations access the database data through conventional means such as SQL queries. Queries can be formed on database-stored metrics such as colour, size, shape, movement, time, object class. Back-end data monitors observe the data in the database and can be used to add further inferred data, such as associating the recurrence of vehicles or people at different locations or times.

5. Tracking and object detection

The PeopleVision *Smart Surveillance Engine* (SSE) is an automated video surveillance system, constructed around algorithms for object detection and tracking. Using a modular architecture, we have experimented with a number of designs for each component. Object detection and tracking are described in more detail elsewhere [4] but we here give an overview of the principles of operation.

5.1 Object detection

The core method for object detection is background subtraction. This compares an automatically acquired reference image with incoming frames of video. Differences between the two images are areas of change in the image which are considered "objects". A sequence of more sophisticated processing attempts

to refine the distinction between true objects and other differences in the images, caused by camera-shake, trees blowing in the wind, lighting changes and shadows.

We have experimented with a variety of object detection algorithms and have two principal approaches: adaptive multi-Gaussian and salient motion. The adaptive multi-Gaussian approach models each pixel as a mixture of Gaussians in color space, following [11], with refinements that examine texture as a way of distinguishing shadows (that change lightness without changing the texture) from objects (that change texture as well as the color of a pixel). Additional mechanisms are employed for actively "healing" stationary objects into the background. A preprocessor allows additional normalization mechanisms, for instance detecting and correcting for camera vibration, camera automatic gain controls and automatic white balance, as well as pixel noise correction (to remove camera noise and compression artifacts).

An alternative method of object detection employs salient motion detection to distinguish moving objects from motion in the background. Traditional background subtraction fails when there is motion in the "background" region of the image, for instance if there are trees blowing in the wind, flowing water or waves. The multi-Gaussian approach handles some of these situations, but the salient motion approach detects objects moving in front of more severe distracting motion by detecting consistency in the motion over a number of frames. Optic flow is carried out over the whole image, and motion vectors over successive frames are chained together. Regions of consistent motion over time are detected as moving objects.

5.2 Tracking

Tracking can be seen as a problem of assigning consistent identities to visible objects. Over time we obtain a number of observations of objects (detections by the background subtraction algorithm) but need to label these so that all observations of a given object are given the same label. When one object passes in front of another, partial or total occlusion takes place, with background subtraction detecting a single moving region. By occlusion handling, we hope to be able to segment this region, labelling each part appropriately, and correctly labelling the detected objects when they separate. In more complex scenes, occlusions between many objects must be dealt with.

When objects are widely separated a simple bounding box tracker is sufficient to associate a track identity with each foreground region. Bounding box tracking works by measuring the distance between each foreground region in the current frame and each object that was tracked in the previous frame, a match being declared if the object overlaps the region or lies very close.

If the foreground regions and tracks form a one-to-one mapping, then the tracking is complete and tracks are extended to include the regions in the new frame using this association. If a foreground region is not matched by any track, then a new track is created, and if a track matches no foreground region, it continues at a constant velocity, but is considered to have left the scene if it fails to match any region for a few frames.

Occasionally, a single track will be associated with two regions. For a few frames this is assumed to be a failure of background subtraction and both regions are associated with the track, but if there are consistently two or more foreground regions, then the track is split into two, to model such cases as when a group of people separate, a person leaves a vehicle, or an object is deposited by a person.

5.3 Appearance models

More complex interactions where more than one track is associated to one or more foreground regions are handled by a mechanism that uses an appearance model of each tracked object.

An appearance model consists of an image of the object — a two dimensional array of colour values with a mask indicating which pixels belong to the object. An appearance model is initialized by copying the foreground pixels of a new track. The appearance model can be correlated with detected foreground regions to track the motion of the centroid of an object being tracked by bounding box tracking. At each frame the appearance is updated by copying the current foreground pixels. During an occlusion, the foreground models of all

Figure 5.2. Appearance models from a PETS 2001 [6] video sequence, showing the appearance of model pixels, as one model recedes (left) and another approaches (right). Pixels not in the model appear black.

the tracks in the occlusion are used to explain the pixels labelled as foreground by the background subtraction mechanism. We assume a depth ordering among the tracks and try to fit the models front-to-back, building up evidence in an explanation map. The position of each object is predicted with a velocity motion model, then the front-most is localized through correlation. Pixels that fall within the foreground mask of the object are entered into the explanation map as potentially being explained by the track. Subsequent objects are correlated with only those pixels in the foreground region which have no explanation so far, and are entered into the explanation map in their turn.

The explanation map is now used to update the appearance models of objects associated with each of the existing tracks. The depth ordering is recalculated by examining those pixels where two objects overlap. Models which account for these disputed pixels better are considered to lie in front of models which match the colour of the foreground less well. The initial depth ordering at the start of an occlusion is considered to be arbitrary since such occlusions generally occlude only a small fraction of the objects. Each model is only updated in those pixels where the model was the front-most object. Regions of foreground pixels that are not explained by existing tracks are candidates for new tracks.

5.4 Track data

With this inductive procedure, track records are created for each object during the period it is visible. Track data is trickled to the database for live visualization and real-time search. A series of post-processing operations can also be carried out to filter out some false alarms and other tracking errors.

In addition to object location and appearance, we also use a classification system to decide the type of object (e.g. person or vehicle) which is also stored in the database. Because of the variability in appearance of objects, classification relies on normalization as described in the next section.

6. Normalization

Normalization of image data is an important process in order to infer physical properties of objects in the scene measured in invariant units, such as meters or miles per hour. Even if only relative quantities are required, physical properties of objects, such as their height or size, should be invariant to their location in the image. Measurements from image data must take into account the perspective distortion due to the projection of the world onto the image plane and other distortions such as lens distortion. In particular, for typical surveillance video with a far field view (i.e., the camera has its viewing direction nearly parallel to the ground plane), the farther aways an object lies, the smaller its projected image size will be. On the other hand, for an overhead camera looking down at a scene, a person standing more directly underneath the camera will appear foreshortened (Figure 6).

Investigators in digital video surveillance have recently begun to address this issue. Traditionally this has been done by semi-automatic calibration (relying on an expert) or rich geometric primitives in the image (such as parallel or orthogonal lines in the image). But realistic digital surveillance, which can be generally deployed, needs an automated solution.

Lv *et al.* [8] were the first to pioneer an effort to perform self-calibration of a camera from the tracking data obtained of a walking human. This method computes a 7 parameter transformation from 3D points in the world to 2D points in the image. The parameters are the focal length, the principal point (u_0, v_0), three rotation parameters, and the height of the camera from the ground plane. These are computed by finding three orthogonal vanishing points of the imaging system using every pair of observations of a human's projected height and location. With sufficiently high quality data, this method can be used to perform a full intrinsic and extrinsic calibration but in practice is somewhat unstable with realistic tracking data.

More recently, Bose and Grimson [3] proposed a method to perform ground plane rectification based on tracked objects moving at a constant velocity. The rectification can be either affine (making parallel world lines appear parallel in rectified image) or metric (making angles in the world plane equal to angles in the rectified image). This method assumes the ground is planar and it is possible to acquire tracks of objects moving at a constant velocity. In practice, these assumptions cannot always be satisfied. The ground is often not planar and it

is difficult observe tracks of objects moving at a constant velocity, in particular for views of pedestrians only or complex intersections/roadways.

Stauffer *et al.* [12] present a method in which projected properties P_j of a particular track j, are modeled by a simple planar system such that the value of the property varies linearly with the distance from the horizon line:

$$P_j(t) = s_j(ax_j(t) + by_j(t) + c), \tag{5.1}$$

where t represents an instance in time or the frame of the measurement. For each track j, an individual scale factor parameter s_j and three global parameters of the planar model (a, b, c) are found as the best fit to the observations (x_j, y_j) for all j. This method is applied to all tracks regardless of the object type (vehicle, pedestrian, animal etc.) The limitation of this approach is that object properties such as height and width depend heavily on the viewpoint direction, particularly for vehicles whose length and width vary greatly. Although in theory, the change in the projected property should vary nearly linearly with distance, this also assumes a planar ground surface, no occlusion, and only perspective distortion.

We propose a method which does not rely on a planar ground surface, is not limited to certain camera viewpoint directions (far field), is not linear/planar, nor does it require objects moving at a constant velocity. Our system relies either on pedestrian data obtained from our classifier or input into the system. In the former case, the classifier is run over a few days, to obtain several sequences in which pedestrians traverse the space. The classifier determines if the track is a person, a vehicle or a group of people. In each case, a confidence measure is assigned to the classification result. Over several days, sequences classified as humans, whose confidence measures are relatively high are selected as input data to the normalization system. This typically finds sequences of pedestrian data without shadows, from decent imaging conditions (no precipitation or wind) and simple pedestrian shape and motions (not carrying objects, wearing hats, holding umbrellas, or performing odd behaviors.)

For each frame j, in the sequence, the position (x_j, y_j) of the foot of the pedestrian (based on the location of the bottom of the major axis of an ellipse which is fit to the data), the length, H, and orientation, θ, of the major axis are used. Normalization is performed by a least squares fitting of a second order polynomial to this data. For each property $p \in (H, \theta)$, we minimize the sum of squares:

$$\min_{a_1,\ldots,a_6} \sum_j [p_j - p(x_j, y_j, a_1, \ldots, a_6)]^2 \tag{5.2}$$

where $a_1, \ldots, a_6$ are the coefficients of the polynomial. For each position in the image, we can predict the height and orientation of the projected image of a person (Figure 6).

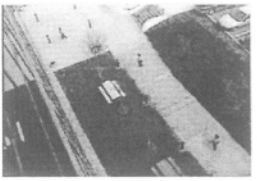
From this information, we can also normalize any of a range of metrics used by the surveillance system. Normalized metrics include area, length, major/minor axis length, major axis angle, and velocity magnitude. For subsystems which rely on frame to frame alignment of the detected object, such

as appearance-based tracking or recurrent motion estimation for the classifier, normalized metrics alleviate the need to scale to an initial segmentation and to estimate a re-scaling on a frame-to-frame basis. It is also now possible to distinguish if a projected view is getting larger for other reasons, such as change in 3D position.

Many subsystems of an automated digital surveillance system can benefit from this normalization. The frame-to-frame tracker can predict frame-to-frame size and orientation changes for appearance-based alignment, or even ignore objects which are smaller for their location than objects of interests for that location should be.

An object classifier can better distinguish physical objects based on properties which are invariant to image location such height, width or speed. For a multi-camera system with either overlapping or non-overlapping views, normalization information can be used to improve matching of object tracks. Lastly, forensic retrieval systems can now search based on absolute sizes and ignore the complex variations due to perspective distortion and the range of viewpoints across different cameras.

Size (pixels)	Orientation (degrees)
11	85
18	80
21	95



Size (pixels)	Orientation (degrees)
24	65
20	81
18	90

Figure 5.3. The top picture shows a typical surveillance far field view of a scene. A person appears smaller the farther away they are. The table to the right shows the change in size and orientation of a human at each of the three locations. The bottom picture is an overhead camera looking down at a scene. A person appears smallest when they are more directly underneath the camera. The table at right shows the change in size and orientation of the corresponding person in the scene. These size/orientation values are predicted for the given position based on prior data and can be used to normalize the live data at each position and across camera views.

7. Multi-camera coordination

Ambient intelligence becomes particularly valuable when a system includes multiple cameras. Integrating information between image processing systems that interpret the video locally requires local communication between intelligent

devices. The representational power of such systems increases far beyond the abilities of a system that replicates independent video interpretation systems.

We can distinguish three kinds of intercamera relations according to the proximity of the cameras fields of view:

- overlapping,
- close,
- distant.

Any or all of these relations may be found in a multicamera system. When cameras overlap, tracking systems can synchronously combine information to disambiguate tracking and derive richer models of objects seen from multiple viewpoints. As described in the next section, resolutions of the two cameras might be radically different, enabling significantly enhanced representations of tracked objects. Overlapping cameras also allow unambiguous continuous tracking of objects over extended regions far beyond the field of view of a single camera. [1]. Overlapping fields of view can be explicitly coded into a system with calibration, or can be learnt by watching the behaviour of tracks over some extended training period, and detecting correlations [12].

When two cameras have non-overlapping but close fields of view, similar continuous tracking can also be carried out, but this requires a degree of inference and a concomitant uncertainty. Training allows a system to learn the interconnection between cameras fields of view [7, 5] — learning when and where objects leaving one camera's field-of-view are likely to be detected in another's.

So far, in the peoplevision system we have only examined the case of cameras with overlapping fields of view. The cameras are calibrated with respect to one another using a homography (a linear transform from image coordinates in one view to image coordinates in another view, that applies to points lying on the ground plane). Objects are tracked using our conventional 2D tracking algorithms in each view, but using the homography, we can know when and where tracks in one view should be visible in another's. Matching tracks are then associated and given a common label, allowing continuous tracking over extended regions.

8. Multi-scale image acquisition

In many camera installations, resolution limits the capabilities of the system. Currently deployed surveillance system almost exclusively use analog video, and are often monochrome and stored on poor-quality, time-multiplexed analog video tapes. The blurred grey images of surveillance footage are familiar from television, seemingly never clear enough to identify a criminal. While quality is improving, and digital technologies promise higher quality and higher resolution, resolution will be limited for years to come. Supposing that a face recognition system requires 100 pixels across a face to perform recognition — to identify any face in a 100m wide space would require a 40 gigapixel static

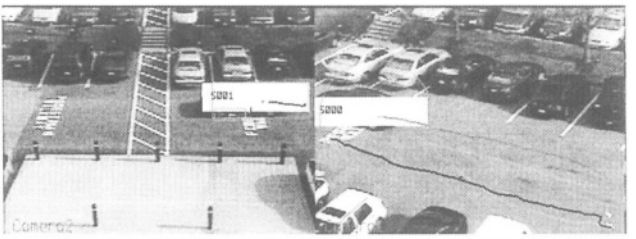


Figure 5.4. Overlapping views from two cameras. When the object in camera 2 (right) enters the field of view of camera 1 (left), the two tracks are registered and tagged as relating to the same object.

camera- far beyond current projections and data-handling capabilities. In practice, surveillance systems requiring high resolution images use the foveation principle seen in human vision — of directing a high-resolution sensor to areas of particular interest. Security guards steer pan-tilt-zoom (PTZ) cameras with a joystick, or use preset zoom positions to quickly examine points of regular interest.

As with all video surveillance though, the guards are fallible. They must be alert to detect an incident in the first place and then require great skill to track when a person must be tracked continuously across multiple cameras. (For instance, a continuous video record may be required to obtain a conviction for shoplifting.) There is the further problem that it becomes impossible to track more than one target at once. Faced with these problems, we have developed several multiscale video acquisition systems that use automatically-controlled PTZ cameras to acquire close-up images of tracked objects.

8.1 Active Head Tracking and Face Cataloging

One such multiscale acquisition system is the Active Head Tracker. This uses three or more calibrated cameras to acquire high resolution images of heads and faces. The system is conceived of as an image acquisition or preprocessing system for a number of human understanding systems, from face recognition to focus of attention determination and audio-visual speech recognition.

The system, shown in Figure 5.5 is based upon independent two-dimensional tracking in each of two static cameras. Typically the cameras are wide angle to achieve joint coverage of a wide operational area. Applying our tracking algorithms gives us the position of each independently moving object in each image. 3-dimensional calibration of the cameras allows triangulation of the two sets of object tracks to obtain both a correspondence between the objects in the two sets, and a 3D position for each object. To triangulate on a distinctive position, we triangulate the 2D centroid of the head in each view, which approximates to the projection of a 3D central point. The head is found by an algorithm that analyses the contour of the segmented object looking for an object part whose position and shape is consistent with being a head.

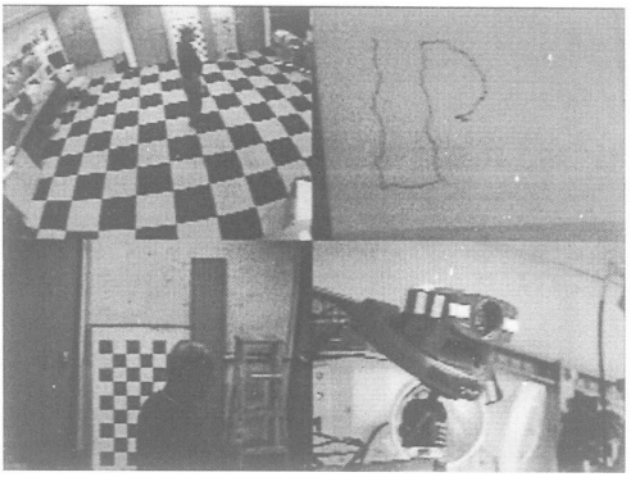


Figure 5.5. The active head tracker. Top left: a person seen by one of the fixed, wide-angle cameras. Bottom left: a close-up view captured by the PTZ camera (seen bottom right). Top right: the track of the person's head projected into the horizontal plane.

The 3D wide-baseline stereo triangulation gives a 3D position for absolute spatial indexing of object tracks and the disambiguation of occlusions that might present difficulties to a single camera, or even narrow-baseline stereo system. It also provides a target point for the active image acquisition system. A calibrated Pan-Tilt-Zoom camera is directed towards the target point and zoomed appropriately to capture the whole head area, taking into account both positioning errors from triangulation and the object speed—a wider zoom being necessary to ensure that faster objects are kept in the frame. The active head tracker provides a significant magnification compared to the static cameras.

In a refinement of this open-loop active head tracking system, a face detection system is applied to the PTZ camera output. Should the imaged head be facing the camera, a face is detected and the camera is servoed in (with a negative feedback control loop now making the system independent of any calibration, segmentation and triangulation errors) until the face fills the video image. A still or video clip is recorded and stored in a database, for human or machine face recognition.

A further extension to the system is the development of sophisticated camera scheduling policies that control the assignment of multiple cameras among the multiple people being tracked by the system. The choice of camera policy is application-dependent. The addition of a multi-camera head pose estimation system [13] that operates at head resolutions as low as 8×8 pixels and thus can be applied in the wide-angle views, allow us to determine, in an absolute, world coordinate system the head orientation of the subjects allows the system to direct at each subject the camera most likely to see the subject's face.

8.2 Uncalibrated, multiscale data acquisition

An alternative path that we have followed for the acquisition of multiscale data is to use a single, uncalibrated camera to trigger foveation by one or more active cameras. In this system, a single fixed camera observes an overview, and an operator selects in the image a number of regions of interest for which high-resolution images are desired (Figure 8.2). For each of these regions, a separate PTZ camera is steered to zoom in on the area of interest, and the steering parameters are recorded and associated with the region of interest. For each available PTZ camera, a separate set of regions and zoom coordinates can be chosen.

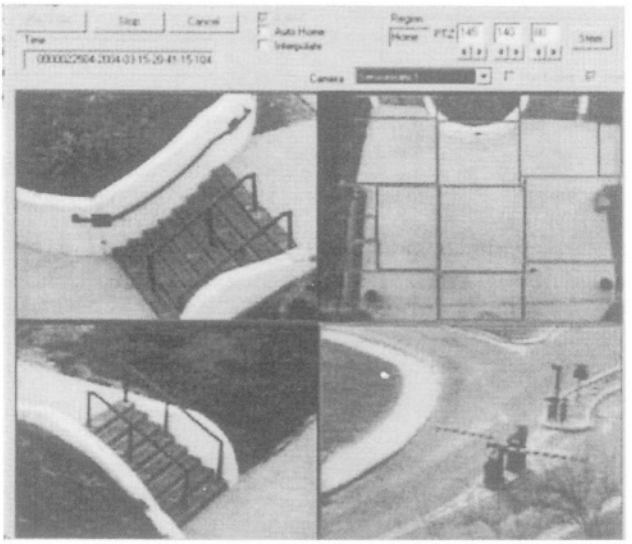


Figure 5.6. Multiscale image acquisition. The top right pane shows the static, "master", camera's view point. The other panes show the views of the steerable "slave" cameras. Boxes in the master's view show regions of interest for which steering parameters of the other cameras have been set up.

After this simple training operation, the system runs autonomously, tracking targets in the 2D view of the static camera (as in Section 5 and steering the corresponding PTZ camera to the associated PTZ coordinates when any object enters one of the regions of interest. Again camera assignment becomes an important, but application dependent aspect of the problem. A typical assignment policy might follow these rules:

- For a single object steer all cameras at the target all the time.
- Assign one camera to each of multiple objects according to which has the best view (for instance "best" might be interpreted as the camera with target area whose centre the object is closest to, or the camera which can

be steered to point at the target in the shortest time). Assign additional cameras evenly to the targets of which they have the best views.

- After following an object for a given period of time, permute the camera assignments so that alternate views of a target are acquired (e.g. left and right sides).

- Steer all available cameras to certain designated targets, e.g. shoplifters.

Once a camera is in place looking at the intended target zone, captured images are taken immediately and periodically, and are sent with tracking information to the central database. The track browsing and query program allows the user to select a track and see all the zoomed-in images associated with that track.

8.3 Extensions

A refinement of the system allows it to operate with a single camera acting as both the master (fixed) and the slave (PTZ) camera. As above, the object detection and tracking are run on the video from the camera with a wide-angle view. Regions of interest are drawn in this field of view and associated with close-up PTZ parameters for the same camera examining the area of interest. When a tracked object enters a region of interest, tracking is suspended, the camera steers to the close-up viewpoint, acquires images for some fixed period (or until some condition is met, such as there being no motion in the close-up view), and then zooms out and recommences tracking. In this way, we have designed a system that autonomously captures licence plate images ifrom every vehicle driving up to an entry barrier, while maintaining an overview of a wide area when no vehicle is at the barrier.

More complex scenarios can also be handled with the system, for instance having multiple master cameras that can each be slaved to acquire close-ups in each other's fields of view when there is no activity in their own.

Extensions to the system are being developed, including continuous control of the PTZ camera based on tracking in the fixed camera; and tracking within the moving PTZ camera view.

9. Indexing Surveillance Data

The vision and active camera control technologies described above process torrents of video data to extract streams of useful information. The information can be used to trigger real time alerts of events requiring human attention, but also provides a rich data stream for delivery to other devices, and for storage and post-hoc searching.

In our Middleware for Large Scale Surveillance (MILS) architecture, individual smart surveillance engines, or groups of collaborating smart surveillance engines that share information about common tracks, communicate tracking information to a database that may be centrally controlled or decentralized. In our implementation SSEs trickle live track data in XML data chunks to a DB2 database. The database aggregates track information and summarizes that in-

formation in track summaries that allow rapid searches over large quantities of surveillance data — from long periods of time and multiple cameras.

Searches can be on any of the data stored in the database, from track motion (position and speed at any time, or aggregated motion), to model appearance (size, colour, type) to aggregate queries that describe multiple attributes for a single track or combination queries involving multiple tracks. Larger scale queries can be statistical in nature or involve aggregation of data over long periods of time. Some examples of the queries that could be answered by such a database query engine are as follows:

- Show all blue cars travelling north-to-south this morning.
- Show the fastest vehicle track in a given time period.
- Show who abandoned this luggage and where they are now.
- Show the average speeed of people at a location.
- Show all tracks that came close to a particular person (handoff detection).

9.1 Visualization

Since the database contains the background appearance, and the appearance and motion of all objects, a highly compressed summary video can be rendered from the database contents, allowing rapid visualization of all incidents meeting search criteria, even over low-bandwidth networks. In parallel to tracking and database storage, a typical system will also encode and index a high fidelity digital video record that can be played back to visualize query results. We have created browsing applications that can deliver the summary information (e.g. to a hand held device) or the full video in response queries or browsing.

10. Privacy

Since video surveillance is a powerful tool with considerable privacy implications, we have also been investigating ways to protect privacy in a video surveillance system. The techniques we have developed centre on the idea of re-rendering video information according to the object oriented representation extracted by our video understanding system. Our ideas on video privacy are more fully explained in a separate paper [10].

The tracking, detection and classification of objects results in a separation of the video into independent streams for the background and each tracked object. Given this information, we can rerender the video manipulating each of these streams independently, for instance replacing each object by a solid rectangle that conveys the location, size and motion of an object without carrying any information about appearance and thus race, age, gender etc. Such rerendering can be tuned to the application in question and governed by access control lists and privacy policies that, for instance, allow security guards to override the obscuration, and permit law-enforcement officers access to raw, unchanged data.

Such a capability has been added into our MILS infrastructure, with tracked regions being blurred out in video playback for users with restricted privileges.

11. Conclusions

In this chapter we have presented the IBM Smart Surveillance System that is a distributed system for the understanding of visual input from a network of cameras. The system is an exploration of a particular kind of ambient intelligent system with distributed sensors, local processing and delivery of resulting data to mobile users over wireless networks. The system extracts rich useful information that can drive real-time alarms or be searched after-the-fact as an index to stored video.

References

[1] J. Black and T. Ellis. Multi camera image tracking. In *International Workshop on Performance Evaluation of Tracking and Surveillance,* 2001.

[2] R.M. Bolle, J.H. Connell, S. Pankanti, N.K. Ratha, and A.W. Senior. *Guide to Biometrics: Selection and Use.* Springer-Verlag, New York, 2003.

[3] B. Bose and E. Grimson. Ground plane rectification by tracking moving objects. In *Joint IEEE Int'l Workshop on VS-PETS,* pages 9–16, October 2003.

[4] J. Connell, A.W. Senior, A. Hampapur, Y.-L. Tian, L. Brown, and S. Pankanti. Detection and tracking in the ibm peoplevision system. In *IEEE International Conference and Multimedia Expo,* 2004.

[5] T.J. Ellis, D. Makris, and J.K. Black. Learning a multi-camera topology. In J. Ferryman, editor, *PETS/Visual Surveillance,* pages 165–171. IEEE, October 2003.

[6] A. Senior *et al.* Appearance models for occlusion handling. In *International Workshop on Performance Evaluation of Tracking and Surveillance,* 2001.

[7] S. Khan and M. Shah. Consistent labeling of tracked objects in multiple cameras with overlapping fields of view. *IEEE Trans. Pattern Analysis and Machine Intelligence,* 25(10), October 2003.

[8] F. Lv, T. Zhao, and R. Nevatia. Self-calibration of a camera from video of a walking human. In *Proc. International Conference on Pattern Recognition,* pages 562–7, August 2002.

[9] G. Potamianos, C. Neti, G. Gravier, A. Garg, and A.W. Senior. Recent advances in the automatic recognition of audiovisual speech. *Proceedings of the IEEE,* 2003.

[10] A.W. Senior, S. Pankanti, A. Hampapur, L. Brown, Y.-L. Tian, and A. Ekin. Blinkering surveillance: Enabling video privacy through computer vision. *IEEE Security and Privacy,* 2004.

[11] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *Proc. IEEE Computer Society Conference on*

Computer Vision and Pattern Recognition, Fort Collins, CO, June 23-25, pages 246–252, 1999.

[12] C. Stauffer, K. Tieu, and L. Lee. Robust automated planar normalization of tracking data. In *Joint IEEE Int'l Workshop on VS-PETS,* October 2003.

[13] Y.-L. Tian, L. Brown, J. Connell, S. Pankanti, A. Hampapur, A. Senior, and R. Bolle. Absolute head pose estimation from overhead wide-angle cameras. In *Intl. Workshop on Analysis ad Modelling of Face and Gesture,* October 2003.

Chapter 6

A DISTRIBUTED MULTIPLE CAMERA SURVEILLANCE SYSTEM

T.Ellis, J.Black, M.Xu and D.Makris
Digital Imaging Research Centre (DIRC), Kingston University, UK
{t.ellis,j.black,m.xu,d.makris}@kingston.ac.uk

1. Introduction

An important capability of an ambient intelligent environment is the capacity to detect, locate and identify objects of interest. In many cases interesting can move, and in order to provide meaningful interaction, capturing and tracking the motion creates a perceptively-enabled interface, capable of understanding and reacting to a wide range of actions and activities. CCTV systems fulfill an increasingly important role in the modern world, providing live video access to remote environments. Whilst the role of CCTV has been primarily focused on rather specific surveillance and monitoring tasks (i.e. security and traffic monitoring), the potential uses cover a much wider range.

The proliferation of video security surveillance systems over the past 5-10 years, in both public and commercial environments, is extensively used to remotely monitor activity in sensitive locations and publicly accessible spaces. In town and city centres, surveillance has been acknowledged to result in significant reductions in crime. However, in order to provide comprehensive and large area coverage of anything but the simplest environments, a large number of cameras must be employed.

In complex and cluttered environments with even moderate numbers of moving objects (e.g. 10-20) the problem of tracking individual objects is significantly complicated by occlusions in the scene, where an object may be partially occluded or totally disappear from camera view for both short or extended periods of time. Static occlusion results from objects moving behind (with respect to the camera) fixed elements in the scene (e.g. walls, bushes), whilst dynamic occlusion occurs as a result of moving objects in the scene occluding each other, where targets may merge or separate (e.g. a group of people walking together).

Information can be combined from multiple viewpoints to improve reliability, particularly taking advantage of the additional information where it minimises occlusion within the field-of-view (FOV). We treat the non-visible regions between camera views as simply another type of occlusion, and employ

spatio-temporal reasoning to match targets moving between cameras that are spatially adjacent. The "boundaries" of the system represent locations from which previously unseen targets can enter the network.

To aid robust tracking across the camera network requires the system to maintain a record of each target entering the system and throughout its duration. When a target disappears from any camera FOV, motion prediction, colour identification, and learnt route patterns are used to re-establish tracking when the target reappears. Each target is maintained as a persistent object in the active database and spatial and temporal reasoning are used to detect these activities and ensure that entries are not retained for indefinite periods.

This chapter describes a multi-camera surveillance network that can detect and track objects (principally pedestrians and vehicles) moving through an outdoor environment. The remainder of this chapter is divided into four sections. The first describes the architecture of our multi-camera surveillance system. The second considers the image analysis methods for detecting and tracking objects within a single camera view. The next section deals with the integration of information from multiple cameras. The final section describes the structure of the database.

2. System architecture

The multi view tracking framework of the surveillance system has been implemented using the architecture shown in Fig. 6.1. The system comprises a set of intelligent camera units (ICU) that detect and track moving objects in 2D image coordinates. It is assumed that the viewpoint of each ICU is fixed and has been calibrated using a set of known 3D landmark points in order to establish a common world coordinate system across the camera network. Each ICU communicates with a central multi view tracking server (MVT), which integrates the information to generate global 3D track data. Individual objects and associated tracking details are stored in a central database. The surveillance database also enables offline learning and subsequent data analysis (see Chapter 7). In addition, given the query and retrieval properties of the surveillance database it is possible to generate pseudo synthetic video sequences that can be used for performance evaluation of object tracking algorithms. The surveillance system employs a centralised control strategy as shown in Fig. 6.1. The multi view tracking server creates separate receiver threads (RT) to process the data transmitted by each intelligent camera unit (ICU) connected to the surveillance network. Each ICU transmits tracking data to each RT in the form of symbolic packets through TCP/IP sockets. The database fulfills a dual role of storing data (e.g. raw track data, ground-plane trajectories, compressed video) and serving online system parameters (e.g. camera calibration data, network ID's) to each of the ICU's, supporting dynamic update of parameters during operation.

3. Motion detection and single-view tracking

The first step in the motion detection process identifies potential regions (cues) associated with object motion by using frame differencing, subtracting

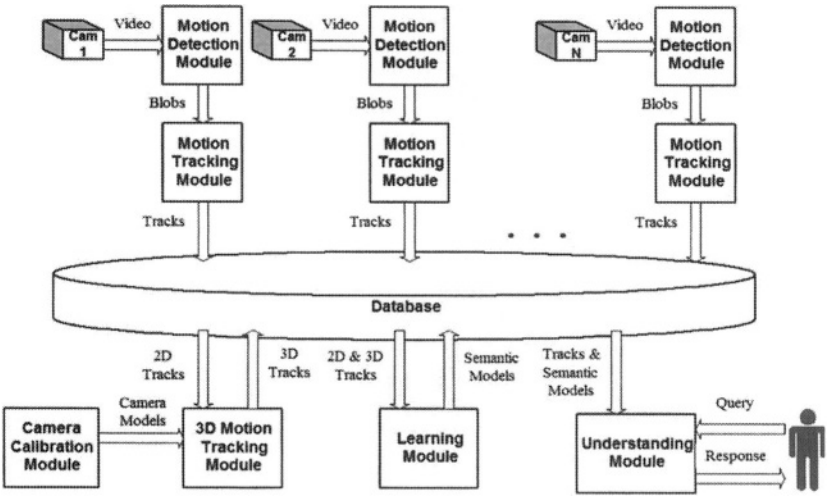


Figure 6.1. Architecture of the Kingston University Experimental Surveillance (KUES) System.

the current image from a reference or background image. This is highly efficient yet sensitive method for detecting changes in the images from fixed cameras, but relies on an accurate estimate of the background which itself may be changing due to variations in illumination (e.g. the diurnal cycle, clouds, artificial lighting), the vagaries of the weather (rain, snow etc), misleading optical effects (e.g. shadows, reflections, specularity) and a miscellaneous variety of irrelevant motion (e.g. wind-blown vegetation and flags, pictures on TV monitors).

A chromaticity-plus-intensity based Gaussian mixture model is used to model the background value for each pixel. The region-based representations of each object is tracked and predicted using a Kalman filter. To increase the robustness and accuracy of object tracking during grouping and occlusion, partial observations are used whenever available. A scene model and a Bayesian network are jointly used to interpret the occluded and exiting objects.

3.1 Motion Detection

Our system uses frame differencing for change detection, comparing each incoming frame with an adaptive background image and classifies those pixels of significant variation into foreground. The probability of observing values $\mathbf{I}$ at a pixel is modelled by a mixture of Gaussians [13]:

$$P(\mathbf{I}_k) = \sum_i \omega_k^{(i)}(\mathbf{I}_k, \mathbf{u}_k^{(i)}, \sigma_k^{(i)}) \tag{6.1}$$

where $\mathbf{u}_k^{(i)}$ is the temporal mean of the i-th distribution, $(\sigma_k^{(i)})^2$ is the trace of the covariance matrix, and $\omega_k^{(i)}$ is the weight reflecting the prior probability that the i-th distribution accounts for the data. At time k, every new pixel value is checked against the Gaussian distributions in a mixture model. For a matched distribution, the pixel measurement is incorporated in the estimate of that distribution and the weight is increased:

$$\mathbf{u}_k = (1-\rho)\mathbf{u}_{k-1} + \rho\mathbf{I}_k \tag{6.2}$$

$$\sigma_k^2 = (1-\rho)\sigma_{k-1}^2 + \rho||\mathbf{I}_k - \mathbf{u}_k||^2 \tag{6.3}$$

For unmatched distributions, their estimates remain the same but the weights are decreased. If none of the existing distributions matches the current pixel value, either a new distribution is created, or the least probable distribution for the background is replaced. The distribution(s), i_B , with the greatest weight is (are) identified as the *apriori* background model for the next frame. At time k, the set of foreground pixels identified is:

$$F_k = \{(r,c) : ||\mathbf{I}_k(r,c) - \mathbf{u}_{k-1}^{(i_B)}(r,c)|| > 2.5\sigma_{k-1}^{(i_B)}(r,c)\} \tag{6.4}$$

A major challenge for target detection in an outdoor surveillance environment comes from the flood of sunlight, to which many existing applications using intensity or *(R, G, B)* pixel representation are sensitive. Chromaticity-based representation for pixel values is a partial remedy. However, the relevant detection is noisy in poorly lit regions and may be undersegmented when part of a target has similar chromaticity to the background. Therefore, the intensity, $I = R + G + B$, and the chromaticity (r, g, b) are combined by using two separate mixtures of Gaussians to model the pixel observation [16].

Suppose $F_k^{(c)}$ and $F_k^{(i)}$ are the sets of the foreground pixels identified using chromaticity and intensity inputs, respectively. The set of final foreground pixels can be computed as the intersection between chromaticity-based foreground (dilated) and intensity-based foreground:

$$F_k = (F_k^{(c)} \oplus B) \cap F_k^{(i)} \tag{6.5}$$

where $\oplus$ denotes the morphological dilation with rectangular structuring element B. The fusion between two types of mixture models overcomes the disadvantages of using each model separately. In regions with lighting variation, few spurious chromaticity-based foreground regions are produced, which masks spurious intensity-based foregrounds due to the sensitivity to illumination changes; in poorly-lit regions, few spurious intensity-based foreground regions are detected, which masks the spurious chromaticity-based foregrounds caused by the sensitivity to noise; undersegmented chromaticity-based foregrounds, due to their similar chromaticity to the background, can be bridged or resized with the morphological dilation by B. The foreground pixels are filtered using a morphological closing (dilation-plus-erosion) operation and then clustered into regions using connected component analysis. Fig. 6.2 shows an example of foreground detection during a minor illumination change. The spurious

foreground at the top-left corner of the intensity-based result is caused by the illumination change and is suppressed in the chromaticity-plus-intensity result, yet the foreground objects equally well detected.

Figure 6.2. Foreground detection under an illumination change: (left) the original frame, (middle) intensity-based foregrounds, and (right) chromaticity-plus-intensity foregrounds.

In low light levels, when the colour quality of a CCTV camera may be severely compromised, the chromaticity-based detection can be disabled. For this, the average intensity of the input image and its variance are used to control the switching. For example, at sunset when the average intensity in a scene is very low, or on an overcast day when the average intensity is moderate but varies slowly, the chromaticity-based detection can be disabled and detection is wholly based on the intensity-based model, with a resultant saving in processing time [16].

The final step encodes the blobs detected by the connected component analysis, representing each with a measurement vector comprised of size, shape and appearance information: area, coordinates of the bounding box, blob centroid and colour histogram.

3.2 Scene Models

Because the camera is fixed, a scene model for static occlusions can be constructed for a specific camera position to support object tracking and reasoning through occlusion [5]. Whilst in this section we employ a manually-constructed scene model, chapter N describes an approach for automatically learning elements such models.. Three types of static scene occlusion are identified as (Fig. 6.3):

- *Border occlusion* (BO), outside the limits of the camera field-of-view.
- *Long-term occlusions* (LO) where objects may leave the scene earlier than expected, resulting in termination of a record in the object database. The long-term occlusion may exist at the border (e.g. buildings or vegetation) or in the middle of an image (e.g. at the entrance to a building). Without prior knowledge of these long-term occlusions, a target disappearing at an LO will be maintained in the object database for a certain of time and may then be mismatched with another target moving in front of the underlying occlusion.

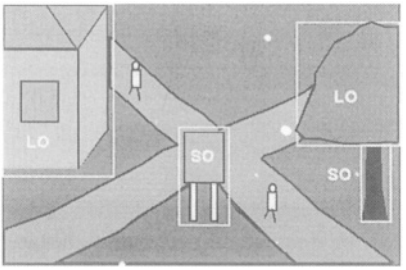


Figure 6.3. Scene model for static occlusions.

- *Short-term occlusions* (SO), where an object is temporarily occluded by a static occlusion, e.g. a tree or a road sign. Prior knowledge of these occlusions can be used to minimize tracking errors. A target moving behind a short-term occlusion is allowed to survive longer than a normal target without observation, depending on the physical dimensions of the underlying occlusion and the velocity of the target.

Each occlusion is characterised by its type (BO, LO or SO) and the region defined by its bounding box. The overlap of target predicted measurement with these static occlusions is used to decide target observability and reason about the status of targets without measurement. Currently a rectangular bounding box is used for each static occlusion to minimise the computational cost. A more accurate representation of an occlusion, e.g. using a convex polygon, is straightforward but not so important here, because the occlusion bounding boxes are only used for the prediction of termination and occlusion events. The determination of such events also depends on the result of tracking (e.g. an object fails to find a corresponding foreground region) since objects may pass in front of an LO or SO type occlusion.

3.3 Target Tracking

Each foreground region is represented by a foreground measurement vector:

$$\mathbf{f} = \begin{bmatrix} r_c & c_c & r_1 & c_1 & r_2 & c_2 \end{bmatrix}^T$$

where (r_c, c_c) is the centroid, r_1, c_1, r_2, c_2 represent the top, left, bottom and right bounding edges, respectively ($r_1 < r_2$ and $c_1 < c_2$). A foreground region may correspond to an object, a group of objects due to dynamic occlusion, or part of an object due to static occlusion. In this paper, we use $\mathbf{f}(i)$ to represent the i-th element of the vector $\mathbf{f}$, e.g. .

A Kalman filter based on a first-order motion model is used to track each object according to the object measurement vector (see section 7.1):

$$\mathbf{z} = \begin{bmatrix} r_c & c_c & r_1 & c_1 & r_2 & c_2 \end{bmatrix}^T$$

We distinguish object measurements from foreground measurements, because they are the same only for separate objects. Because our system aims to monitor pedestrians and vehicles, each target is assumed to move along a linear trajectory at constant velocity and with constant size. In practice, any minor violation of this assumption can be encoded in the process covariance matrix. The state vector used is:

$$\mathbf{x} = [\ r_c \quad c_c \quad \dot{r}_c \quad \dot{c}_c \quad \Delta r_1 \quad \Delta c_1 \quad \Delta r_2 \quad \Delta c_2\ ]^T$$

where $(\Delta r_1, \Delta r_2)$ and $(\Delta r_2, \Delta c_2)$ are the *relative* positions of the two opposite bounding box corners to the centroid. They not only incorporate height and width information, but also accurately represent the bounding box even when the centroid is shifted away from the geometric centre of the bounding box, e.g. due to asymmetry or self-shadows.

Each target is also characterised by its status

$$S \in \{new, terminated, updated, occluded, missing\}$$

where *new* is for objects entering the scene or splitting from existing objects, *terminated* for objects exiting the scene or being inactive for some time, *updated* for objects with some measurement, including objects merged or partially occluded, *occluded* for those completely behind static occlusions, and *missing* for those without observation and un-interpreted.

Each *new* object is initialised with the blob features and assigned a zero velocity and a large initial error covariance to encode the uncertainty of the unknown velocity. An *occluded* or *missing* object is updated with the *a priori* estimate with a linearly increasing error covariance. Once the error covariance becomes too large, i.e. the object has no observation for a certain number of frames, a *missing* object becomes a *terminated,* though an *occluded* object will be maintained if the invisible time is less than the expectation (occlusion width / velocity) plus a tolerance reflecting the error covariance.

3.4 Partial Observation

When tracking multiple objects in a complex scene, it is noted that the object measurement, $\mathbf{z}_k$, may be partly unavailable due to dynamic or static occlusion. Fig. 6.4 shows examples of partial observation, where the target is spatially overlapped with another target or a par of the scene. If these partial observations are input into the estimation process during grouping or occlusion, the tracker should be more robust and accurate than those without any observation [17].

We decide the observability of the objects on the basis of the predicted measurement, $\hat{\mathbf{z}}_k^-$, the foreground measurement, $\mathbf{f}_k$, and the scene model. The outcome is represented by the observability vector, $\mathbf{m}_k$, which has the same dimension as the object measurement vector, $\mathbf{z}_k$, with a one-to-one correspondence in their elements. Each element of rrifc has a binary value: *observable* (1) or *unobservable* (0). At the beginning of frame k, each object is set to *observable,* i.e. $\mathbf{m}_k(l) = observable, l \in [1, 6]$, and then subject to a 3-stage modification [18].

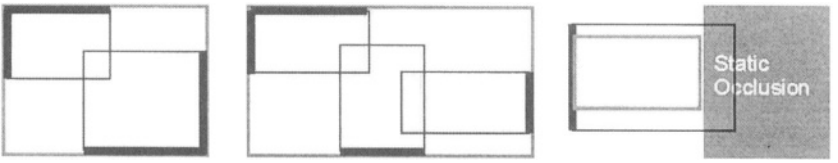


Figure 6.4. Partial observation during dynamic grouping (left and middle) and static occlusion (right).

1 Observability in grouping

For each tracked object and each foreground measurement, a match score is computed on the basis of the Mahalanobis distance, but it is set to zero if the predicted centroid of the object is within the bounding box of the foreground measurement. Each object selects its corresponding foreground measurement in term of minimum match score within a tolerance. For each foreground measurement, the objects that best correspond to it form a group of the objects that are most likely to be merged. The group may include multiple objects, one object, or be empty. For each object in the group, the observability of each bounding edge is modified according to whether this bounding edge is also the corresponding bounding edge of the group (foreground measurement).

2 Observability of foreground measurement

The observability of an object also depends on the observability of its associated foreground measurement. If the bounding box of the underlying foreground region touches the border of the field-of-view, its relevant bounding edge becomes *unobservable* and thus inhibits (masks) the relevant observability for the associated object bounding edge.

3 Observability in static occlusion

When an object is *partly* hidden behind a static occlusion, the measurement of some of its bounding edges become unreliable. For each object, each of the predicted bounding edges is checked. If either corner delimiting that edge is within a static occlusion defined in the scene model, that edge becomes *unobservable.* Once an object is determined as partially occluded by a static occlusion, a bounding box distance measure is used in the data association. It emphasises the minimum distance between corresponding bounding edges, rather than the average distance in the Mahalanobis distance. Therefore, an object may be matched with a much smaller foreground region at the border of a static occlusion.

After the observability of the four bounding edges for i-th object is determined, the observability of its centroid can be decided as *observable* only when all the four bounding edges are *observable.* An alternative but less strict definition considers (r_c, r_1, r_2) and (c_c, c_1, c_2) separately.

For a partially unobservable object, a measurement vector is constituted, whose members can be classified into two inter-correlated blocks (r_c, r_1, r_2) and (c_c, c_1, c_2). The inter-block variables are bound by the constant height (Δr_1 and Δr_2) and constant width (Δc_1 and Δc_2) assumption. Within each block, if all the variables are *unobservable,* the only clue for their measurements are the prediction; if part of its variables is *observable,* the *unobservable* measurements can be deduced from the *observable* measurements. Suppose the observability matrix, $\mathbf{M}_k$, is a diagonal matrix whose main diagonal is the observability vector $\mathbf{m}_k$. The measurement vector is estimated by:

$$\mathbf{z}_k = \mathbf{M}_k\mathbf{f}_k + (\mathbf{I} - \mathbf{M}_k)\mathbf{d}_k \tag{6.6}$$

where $\mathbf{d}_k$ is the deduced measurements of *unobservable* variables from *observable* measurements using some constraints on height and width. The height and width information in the *a priori* state estimate, $\hat{x}_k^-$, is used to compute $\mathbf{d}_k$. When one bounding edge is *unobservable* and its opposite is *observable,* this edge and the centroid are deduced from the opposite edge by assuming Δr_1 and Δr_2 (or Δc_1 and Δc_2) to be constant. When a pair of opposite edges are *observable* but the centroid is *unobservable,* the centroid is deduced by assuming the ratio $\Delta r_1/\Delta r_2$ (or $\Delta c_1/\Delta c_2$) to be constant. If all the variables in an inter-correlated block are *unobservable,* $\mathbf{d}_k = \hat{\mathbf{z}}_k^-$ for that block.

To reflect the lack of actual measurement data the corresponding element in the measurement covariance matrix, $\mathbf{R}_k$, is increased by λ ($\lambda > 1$) to reflect an increased uncertainty. Suppose $\mathbf{R}$ is the measurement covariance matrix for a completely *observable* object, the measurement covariance matrix for a partially observable object becomes:

$$\mathbf{R}_k = \mathbf{M}_k\mathbf{R}\mathbf{M}_k^T + (\mathbf{I} - \mathbf{M}_k)\lambda\mathbf{R}(\mathbf{I} - \mathbf{M}_k)^T \tag{6.7}$$

Because the centroid and bounding edges of each object are measured independently, $\mathbf{R}$ is a diagonal matrix; $\mathbf{M}_k$ is also diagonal, with each element being either 1 or 0. Hence, the equation above can be simplified as:

$$\mathbf{R}_k = [\mathbf{M}_k + \lambda(\mathbf{I} - \mathbf{M}_k)]\mathbf{R} \tag{6.8}$$

Using this equation, an *observable* variable of an object is estimated with a normal measurement variance, whilst an *unobservable* variable is estimated with a larger measurement variance. Therefore, the *observable* bounding edges contribute more to the object estimation than the *unobservable* bounding edges.

Fig. 6.5 shows two examples of object tracking through grouping. In the first example (top row), two previously separate pedestrians (No. 13 and 16) merge and share a large foreground region; then another pedestrian (No. 17) joins in the group, forming a larger foreground region; finally, the group of pedestrian splits. In the second example, a white van (No. 3) first passes by a newly stationary dark car (No. 2), heads toward and occludes a group of people (No. 4), decelerates and stops separately. With the use of partial observation, the position and size of each object are correctly estimated during grouping and all the objects are correctly re-tracked after splitting. To quantitatively evaluate the

performance of this algorithm, we have applied it to the PETS'2001 sequences and compared the results with those using the traditional blind tracking during occlusion. The new algorithm has advantages in terms of 32% decrease in tracking error and 63% improvement in path coherence [17].

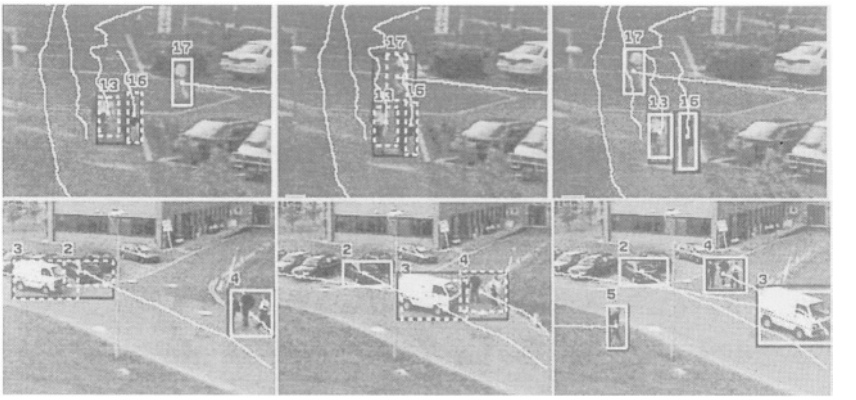


Figure 6.5. Tracking a group of targets with partial observation.

3.5 Target Reasoning

After matching *new* and *updated* objects to blob measurements, there remain some objects that fail to match any blob. This may arise from a variety of reasons: objects leaving the scene; the occlusion of objects by scene elements; the failure of foreground detection, or some unknown reason. The ambiguity can be partly relieved by using domain knowledge. For example, if it is known that the predicted position of an unmatched object overlaps a long-term occlusion, it is most likely that this object left the scene. This is the motivation of a rule-based system. However, there exist uncertainties in such domain knowledge: not all of the objects close to a long-term occlusion will leave the scene — they may walk in front of the occlusion; the foreground detection may fail at any position in a scene; the merging of objects near the occlusion is not reliably detected.

The uncertainties can be encoded in the conditional probabilities between the variables. The object status can then be inferred through a process of deduction. A Bayesian Network [12] is a framework for representing and using domain knowledge to perform probabilistic inference for a set of variables $\mathbf{X} = \{X_1, \ldots, X_N\}$ and has been used in motion analysis in [3]. A BN is a directed acyclic graph in which nodes represent random variables and arcs represent causal connections among the variables. Associated with each node is a conditional probability table given each possible state of its parents. In the case that a node has no parents, conditional probabilities degenerate to priors. By assuming conditional independence between some variables, the joint probability distribution for variable $\mathbf{X}$ to have the value, $\mathbf{x} = \{x_1, \ldots, x_N\}$, is

given by:

$$P(\mathbf{x}) = \prod_{i=1}^{N} P(x_i|pa(x_i)) \tag{6.9}$$

When evidence **e** are observed for a subset of the nodes, the estimate for any of the remaining nodes can be computed to maximise the posterior probabilities:

$$\hat{\mathbf{x}} = \text{argmax}_{\mathbf{x}} P(\mathbf{x}|\mathbf{e}) \tag{6.10}$$

The Bayesian network used for reasoning about unmatched objects is shown in Fig. 6.6. The variables EB ("Exit from BO"), EL ("Exit from LO"), OS ("Occluded by SO"), and MS ("Missing") are query nodes. The distance measures D(BO), D(LO) and D(SO), as well as the variable UM ("Unmatched object"), are evidence nodes. To simplify the representation and learning of the conditional probabilities, all quantities in the network are discretized into binary values. The distance measures used are the bounding box distance and include:

- D(BO) — the distance to the BO. D(BO) = 0 if the predicted bounding box of an object is completely inside the BO.
- D(LO) — the distance to the closest LO. D(LO) = 0 if the predicted bounding box of an object is inside an LO.
- D(SO) — the distance to the closest SO. D(SO) = 0 if the predicted bounding box of an object is inside an SO.

The network structure is determined using domain knowledge: UM ("Unmatched object") usually arises from an object exiting from the border occlusion (EB), exiting from a long-term occlusion (EL), occluded by a short-term occlusion (OS), or missing for unknown reasons (MS); when EB ("Exit from BO") is true, the prediction of the underlying object usually goes into the border occlusion so that D(BO) is small; when EL ("Exit from LO") is true, the prediction of the underlying object tends to overlay a long-term occlusion so that D(LO) is small; when OS ("Occluded by SO") is true, the prediction of the underlying object is often very close to a short-term occlusion so that D(SO) is small. The prior and conditional probabilities attributed to the variables in the network are listed as follows. To reduce the number of conditional probabilities to specify for node UM, the noisy-OR model is applied among its four parents EB, EL, OS and MS. This model assumes that (1) each parent has an independent chance of causing UM, (2) all the possible causes are listed (node MS works as the leak node explaining miscellaneous causes; once MS is true, it always gives rise to an unmatched object, i,e. $\mathrm{P(UM = T \mid MS = T) = 1)}$, and (3) whatever inhibits a parent from causing UM = T is independent of whatever inhibits another parent from causing UM = T. The independent inhibitor probabilities are $q_1 = \mathrm{P(UM = F \mid EB = T)}$, $q_2 = \mathrm{P(UM = F \mid EL = T)}$, $q_3 = \mathrm{P(UM = F \mid OS = T)}$, and the conditional probability table for node UM is as follows:

By using noisy-OR relationships, the variable UM, which depends on $n = 4$ parents, can be described using $O(n)$ parameters instead of $O(2^n)$ for the full

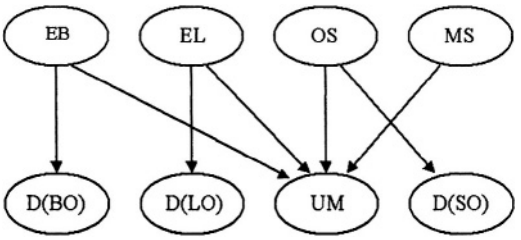


Figure 6.6. The Bayesian network for reasoning about unmatched objects.

Table 6.1. Conditional probabilities in Bayesian network.

EB	EL	OS	P(UM=F)	P(UM=T)
F	F	F	$1-p_4$	p_4
T	F	F	$q_1(1-p_4)$	$1-q_1(1-p_4)$
F	T	F	$q_2(1-p_4)$	$1-q_2(1-p_4)$
T	T	F	$q_1q_2(1-p_4)$	$1-q_1q_2(1-p_4)$
F	F	T	$q_3(1-p_4)$	$1-q_3(1-p_4)$
T	F	T	$q_1q_3(1-p_4)$	$1-q_1q_3(1-p_4)$
F	T	T	$q_2q_3(1-p_4)$	$1-q_2q_3(1-p_4)$
T	T	T	$q_1q_2q_3(1-p_4)$	$1-q_1q_2q_3(1-p_4)$

conditional probability table. The entire network is specified by 13 parameters, including 4 priors, 3 inhibitor probabilities and 6 conditional probabilities. Given the observed values for the evidence nodes, the posterior probabilities of any unmatched object caused by EB, EL, OS or MS are computed using the junction tree algorithm and the most probable explanation can be given.

These prior and conditional probabilities can be learned automatically over sample sequences taken at the same scene. For the four query nodes, this is to estimate the probability $P(x_i = T) = \theta_i$, given a set D of N observations. Suppose the likelihood to generate data D is a binomial distribution $P(D|\theta_i) = C(N, N_T)\theta_i^{N_T}(1-\theta_i)^{N-N_T}$ where $N_T = N(x_i = T)$ and θ_i has a Beta prior $P(\theta_i) = Beta(\alpha_T, \alpha - \alpha_T)\theta_i^{\alpha_T - 1}(1-\theta_i)^{\alpha-\alpha_T-1}$. The maximum *a posteriori* (MAP) estimate [9] for θ_i is:

$$\hat{\theta_i^{MAP}} = \frac{\alpha_T + N_T}{\alpha + N} \tag{6.11}$$

The hyperparameters α and α_T can be thought of as imaginary counts from our prior experience, equivalent to a sample size α . For the four evidence nodes the MAP estimate is:

$$\theta^{M\hat{A}P}_{x_i|pa(x_i)} = \frac{\alpha_T(x_i, pa(x_i)) + N_T(x_i, pa(x_i))}{\alpha(pa(x_i)) + N(pa(x_i))} \tag{6.12}$$

To estimate the state of the query nodes in learning probabilities, a decide-and-verify strategy is used on the assumption that foreground detection failures are very unlikely to occur and are independent. The EB or EL is thought of as true if an object with its prediction overlaying the BO or an LO cannot find an associated foreground region over a certain number of frames. The OS is believed true if an object with its prediction overlaying an SO cannot find an match over the expected invisible time plus a tolerance. The MS is considered true if an object cannot find a match over a certain number of frames for other reasons not trivial to automatically recognise. These reasons include foreground detection failure (targets moving beyond detection range, long-stationary targets absorbed into background, targets walking in front of a scene element with similar colour) and failure to build some LOs and SOs that actually exist in the scene. Therefore, an incomplete construction of the scene model will lead to a greater prior for the MS node. In addition, it is the local conditional probabilities for each LO or SO, rather than the global ones for all LOs or all SOs, that need to be learned. Such a local conditional probability represents how likely objects are to go behind a specific occlusion, while the EL's and OS's priors indicate the global occlusion density in the scene. When reasoning about the status of an object, the local conditional probabilities used are those of the nearest LO and SO to the object. Fig. 6.7 shows the tracking of multiple

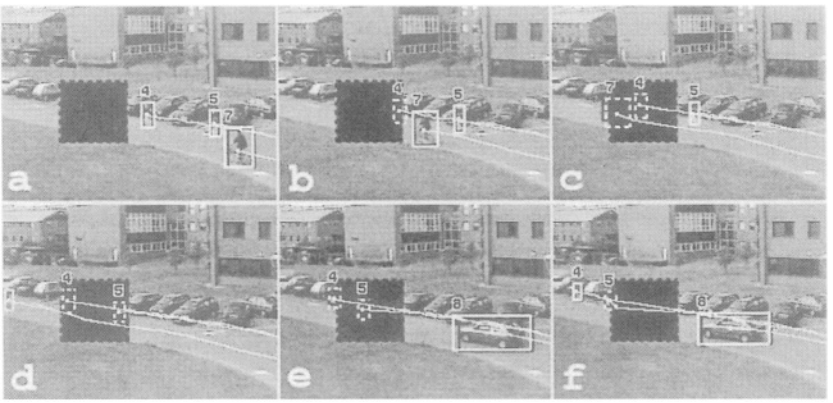


Figure 6.7. Tracking of objects completely behind a static occlusion.

targets completely occluded by a synthetic occlusion. In Figs. 6.7(b) and (c), when a pedestrian (No. 4) and a cyclist (No. 7) disappear behind the occlusion, they are classified as occluded objects by the target reasoning stage and their location estimate is predicted by the tracker. If their predicted emergence from the opposite side of the occlusion is delayed (i.e. the expected reappearance time elapses) but no match is found to any foreground region, these targets are

"held" at the border of the occlusion for a certain number of frames to cover the error covariance while anticipating a match (Figs. 6.7(c) and (d)). In this case, our algorithm correctly tracks each of the three targets. In all of the 23 objects passing behind the black occlusion added to the PETS'2001 sequences, only one object is incorrectly re-tracked (when two targets reappear at the same time), while a traditional algorithm without sensing the presence of any static occlusion reports three failures and a 59% increase in tracking error.

4. Multi view tracking

The task of multi view object tracking is comprised of many tasks. Initially, moving objects of interest must be identified in each camera. This represents a challenging problem, particularly in outdoor environments where lighting conditions cannot be controlled and image intensities are subject to large changes in illumination variation. Each camera in the surveillance network has an intelligent sensor, which employs a robust motion segmentation and object tracking strategy as was discussed in the previous section. It is assumed that each camera view is fixed and calibrated in a world coordinate system. The multi view object tracking framework should be able to integrate tracking information from each camera and reliably track objects between views. In addition the multi view tracker should be able to resolve both dynamic occlusions that occur due to object interaction, and static occlusions that can occur due to the scene constraints, for example trees that form occlusion regions.

The multi view tracker uses a training phase to learn information about the scene, which can facilitate the integration and object tracking process. This can include learning the relations between each view to allow feature correspondence to assign a unique label for an object even it is visible in several views simultaneously. In a typical surveillance environment the cameras are placed to maximise the field of coverage of the scene. As a consequence some cameras will have limited overlap, which increases the difficulty of tracking objects without loss of identity. Hence, the multi view tracker must be able to track objects between non-overlapping and spatially adjacent views. The system should be able to exploit spatial cues to maintain the identity of tracked objects. Since the object disappears from the field of view temporarily, to increase the likelihood of matching the object on reappearance it will be necessary to record attributes of the object at the time of its exit.

4.1 Homography Estimation

A homography defines a planar mapping between two camera views that have a degree of overlap. The homography mapping provides a mechanism for matching object features between overlapping camera views. Most surveillance scenes conform to the ground plane constraint, allowing the homography to be applied for matching objects between different camera views. The application of the homography is illustrated in Fig. 6.8 where it is used to correspond objects between three overlapping camera views. The black arrows show how

the homography can project the centroid of objects on the same ground plane between different views.

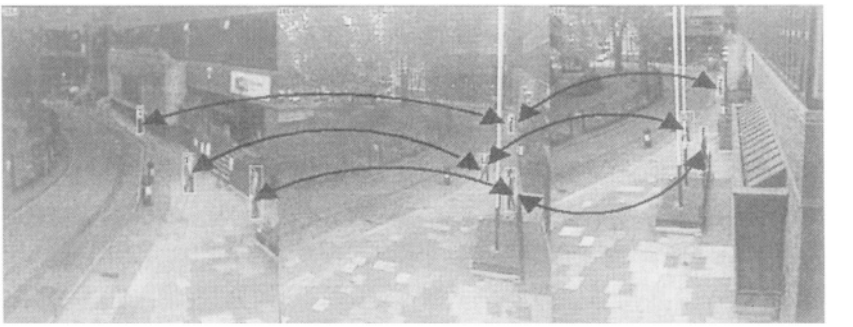


Figure 6.8. Example of viewpoint correspondence between three overlapping camera views.

4.2 Least Median of Squares

Given a set of correspondence points the homography can be estimated using the method described in section 7.2. The next step is to define a process that would allow a set of correspondence points to be determined automatically from a set of input data. Given a set of sparse object trajectories they can be used to provide training data for estimating the homography transformation between the two overlapping camera views. The object trajectories are taken during periods of low activity, in order to reduce the likelihood of finding false correspondence points. The trajectories take the form of a set of tracked object centroids that are found using the feature detection method described in section 3.3. A Least Median of Squares (LMS) approach is used to automatically recover a set of correspondence points between each pair of overlapping camera views, which can be used to compute the homography mapping. The LMS method performs an iterative search of a solution space by randomly selecting a minimal set of correspondence points to compute a homography mapping. The solution found to be the most consistent the set of object trajectories is taken as the final solution. Stein [14] used this method for registering ground planes between overlapping camera views. The LMS method was used since it is a robust alignment algorithm where the data contains a number of outliers

The homography relations between each overlapping camera are used to match detected moving objects in each overlapping camera view. The transfer error is the summation of the projection error in each camera view for a pair of correspondence points. It indicates the size of the error between corresponded features and their expected projections according to some translating function, the object centroid homography in our case. In Mikic, et al. [11] the 3D epipolar constraints were used as a basis for matching. Each method has its own advantages. The epipole line based approach can still function even if the two views do not share a common dominant ground plane but requires that

the camera geometry between the two views is known in advance and fairly accurate. The homography-based method assumes that each camera view shares a dominant ground plane.

The biggest advantage of the homographic method over the epipole based method is that the homography maps points to points, while the epipole approach maps points to lines, so a one dimensional search still needs to be performed to establish an object correspondence. The homographic method could be applied to all regions ofthe image assuming that we had 3D camera geometry along with terrain information of the scene, for example an elevation map. A graphical depiction is given for the feature matching in Fig. 6.9. The bounding box of each object is displayed. The white lines represent the epipole lines for each object centroid terminating at the ground plane. The black points represent the tracked centroid of each detected object. The white points in each bounding box represent the projection of the object centroid using the homography as a transformation. The two images in Fig. 6.9 shows an example of matching two vehicles The transfer error is used by the homography alignment and viewpoint correspondence methods for assessing the quality of a corresponded pair of centroids in two different camera views. The transfer error associated with a correspondence pair is defined as:

$$d(x', H^{-1}x'')^2 + d(x'', Hx')^2$$

where x' and x'' are projective coordinates in view 1 and view 2 respectively, H is the homography transformation from view 1 to view 2, and $d(a, b)$ is the Euclidean distance between a pair of projective coordinates a and b.

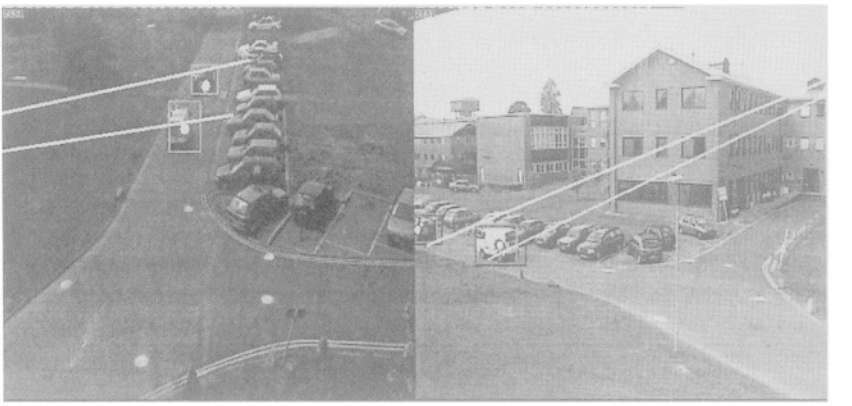


Figure 6.9. Feature matching using: epipole line analysis and homography alignment.

4.3 Feature Matching Between Overlapping Views

The LMS algorithm described in section 4.2 is used to determine a set of correspondence points, which were then used to compute the object centroid

homography. This homography can be used to correspond object tracks in the testing video sequences. Once the calibration data and homography alignment model are available we can use the relationship between both camera views to correspond detected objects. From observing the results of motion detection it is apparent that the object centroid is a more stable feature to track in 3D, since it is more reliably detected than the top or bottom of the object, particularly in outdoor scenes where the object may be a far distance from the camera.

To summarize the following steps are used for matching 2D object tracks, taken from different views, for a given image frame:

1 Create a list of all possible correspondence pairs of objects for each camera view.

2 Compute the transfer error for each object pair

3 Sort the correspondence points list by increasing transfer error.

4 Select the most likely correspondence pairs according to the transfer error. Apply a threshold so that correspondence pairs where Transfer Error$> \epsilon_{max}$ are not considered as potential matches.

5 Create a correspondence points list for each matching object

6 Map each entry in the correspondence points list using 3D line intersection of the bundle of image rays to locate the object in 3D.

7 For each object centroid which does not have a match in the correspondence pair list use the calibration information to estimate the location of the object in 3D

Two additional constraints are applied to the viewpoint correspondence is that we do not allow one to many mappings between observations in each camera view. This has the effect of reducing the number 'phantom' objects which can appear at the end of a dynamic occlusion. An example of viewpoint correspondence is shown in Fig. 6.10, the left image shows the original objects detected by the 2D object tracker, and the right image shows the observations remaining once viewpoint correspondence has been applied. It can be observed that the number of phantom objects near the lamppost in left camera view have been eliminated by the feature matching process. The viewpoint correspondence process has the affect of reducing the number of false objects that have been detected by the 2D object tracker.

4.4 3D Measurements

Given a set of corresponded object features along and camera calibration information it is possible to extract 3D measurements from the scene. Using multiple viewpoints improves the estimation of the 3D measurement. A 3D line intersection algorithm is employed to estimate each object's location in world coordinates. Using the calibrated camera parameters it is also possible to

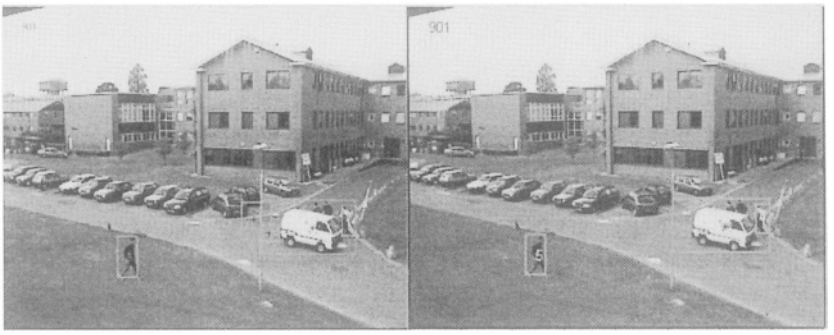


Figure 6.10. Example of feature matching between overlapping views.

estimate the uncertainty of the image measurement by propagating the covariance from the 2D image plane to 3D world coordinates. A 3D line intersection algorithm is used to estimate the location of an object in a least squares sense. Details of the method used to perform this process is given in section 7.3.

4.5 Tracking in 3D

Using the approach described in sections 4.2 and 4.3 we are able to merge the object tracks from separate camera views into a global world coordinate view. This 3D track data provides the set of observations, which are used for tracking using the Kalman filter.

The Kalman filter provides an efficient recursive solution for tracking the state of a discrete time controlled process. The filter has been applied in numerous tracking applications for visual surveillance. The 3D Kalman filter tracker assumes a constant velocity model. A summary is given of the state model used for tracking in following equations. At each state update step the observation covariance is set according to the measurement uncertainty determined by projecting a nominal image covariance from the image to the 3D object space.

State Model

$\mathbf{X}_t = [\ x \quad y \quad z \quad \dot{x} \quad \dot{y} \quad \dot{z}\ ]^T$ where, (x, y, z) is the spatial location in world coordinates, and $(\dot{x}, \dot{y}, \dot{z})$ is the spatial velocity in world coordinates.

State Transition Model **Observation Model**

$$A = \begin{bmatrix} 1 & 0 & 0 & T & 0 & 0 \\ 0 & 1 & 0 & 0 & T & 0 \\ 0 & 0 & 1 & 0 & 0 & T \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \qquad (6.13)$$

Where T is the difference of the time of capture of the current and previous image frame. The Kalman filter allows the system to reliably track the object state even if the objects disappear from the camera view for a five frames. This value was set based on empirical evidence.

3D Data Association. The 2D object tracks detected by the background subtraction are converted to a set of 3D observations using the 3D line intersection algorithm as discussed in section 4.3. Since the system is tracking multiple objects in 3D it is necessary to ensure that each observation is assigned to the correct object being tracked. This problem is generally referred to as the data association problem. The Mahalanobis distance provides a probabilistic solution to find the best matches between the predicted states of the tracked objects and the observations made by the system:

$$M_D = (H\hat{X}_k^- - Z_k)^T (H\hat{P}_k^- H^T + R_k)^{-1} (H\hat{X}_k^- - Z_k) \qquad (6.14)$$

An appropriate threshold can be chosen for the Mahalanobis distance by selecting a value that gives a 95% confidence for a match, assuming a chi-square distribution. The dimension of the observations for the 3D tracker is 3, hence the value threshold selected for M_D was 7.81. A Mahalanobis distance table is created between each tracked object and observation. The system then assigns each observation to each tracked object based upon the size of the Mahalanobis distance.

Outline of 3D Tracking Algorithm. The following is a summary of the steps used to update each tracked 3D object for a giving image frame:

1 For each 3D observation in frame T, create a Mahalanobis distance table and sort by the distance measure. Threshold the values such that each Mahalanobis distance $< \xi_{max}$

2 For each existing tracked object:

   (a) Select the observation which has the largest likelihood of being a match

   (b) Update the tracked object using this observation

3 For each existing tracked object not matched to an observation

   (a) If the tracked object has not been matched in K frames then mark it as deleted, else

   (b) Update the tracked object using the predicted state estimate

4 For each unmatched observation:

   (a) Create a new tracked object, using the observation to initialise the object state

   (b) Set the initial covariance of the object state.

The following two constraints are applied during the object state update process:

1 Each observation can be used to update only one existing tracked object.

2 A new tracked object can only be created when its initial observation does not match an existing tracked object

The first constraint prevents an object being updated during a dynamic occlusion when the observation is not consistent with its predicted trajectory. The second constraint prevents new tracks being created at the end of a dynamic occlusion when the objects separate.

Multi View Tracking Example. Fig. 6.11 shows how dynamic occlusions can be resolved in 3D when tracking objects between two camera views. The left figure shows a pedestrian being occluded by a vehicle, while the right image shows a group of pedestrians being occluded by a white van. The 3D ground plane map shows that the tracked objects can still be tracked in 3D during the dynamic occlusions that occur in both camera views.

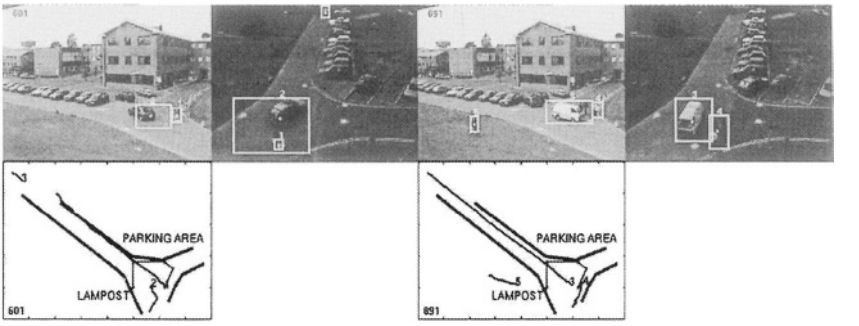


Figure 6.11. Examples of handling dynamic occlusion.

4.6 Non-Overlapping Views

In a typical image surveillance network the cameras are usually organised so as to maximise the total field of coverage. As a consequence there can be several cameras in the surveillance network that are separated by a short temporal and spatial distance, or have minimal overlap. In these situations the system needs to track an object when it leaves the field of view of one camera and the re-enters the field of view of another after a short temporal delay. For short time durations of less than two seconds the trajectory prediction of the Kalman filter can be used to predict where the object should become visible again to the system. However, if the object changes direction significantly of or disappears for a longer time period this approach is unreliable. In order to handle these cases the system uses an object handover policy between the pair

of non-overlapping cameras. The object handover policy attempts to resolve the handover of objects that move between non-overlapping camera views. The system waits for a new object to be created in the adjacent camera view. A data association method is applied to check the temporal constraints of the objects exit and re-entry into the network field of view.

Object Handover Regions. The object handover region models consist of a linked entry and exit region along with the temporal delay between each region. The major entry and exit regions in each camera are automatically learned by post-track analysis of tracking data. Given the set of entry and exit regions the learning component of the surveillance system also determines the spatial links between entry and exit zones in different camera views, as discussed in depth in Chapter 7. The temporal delay can be determined manually by observation, or by generating statistics from the data stored in the database. The temporal delay gives an indication of the transit time for the handover region for a specific object class, so the temporal delay for a pedestrian object class and a vehicle object class would be different based on the set of observations used to generate the statistics. Each entry and exit region is modelled as a Gaussian:

$$\langle (x, y), \Sigma \rangle \tag{6.15}$$

where (x, y) is the centre of the distribution in 2D image coordinates, and Σ is the spatial covariance of the distribution. The following convention is used to describe the major entry and exit regions in each camera view:

- X_i^k is the k-th exit region in the i-th camera view.
- E_j^l is the l-th entry region in the j-th camera view.

Given the set of major exit and entry regions in each camera the following convention is used to define the handover regions between the non-overlapping camera views: $H_{ij}^p = \langle X_i^k, E_j^l, t, \sigma \rangle$ is the p-th handover region between camera i-th and j-th camera views.

As previously discussed each handover region H_{ij}^p consists of a spatially connected exit and entry region pair (X_i^k, E_j^l), along with the temporal delay and the variance of the temporal delay (t, σ) between the exit and entry region. An example of object handover regions is visually depicted in Fig. 6.12. The black and white ellipses in each camera view represent the major entry and exit regions in each camera. The links represent the handover regions between each camera.

Object Handover Agents. The object handover mechanism only needs to be activated when an object is terminated within an exit region that is linked to an entry region in the adjacent camera view. Once the object leaves the network field of view and is in transit between the non-overlapping views the system cannot reliably track the object.

Handover Initiation

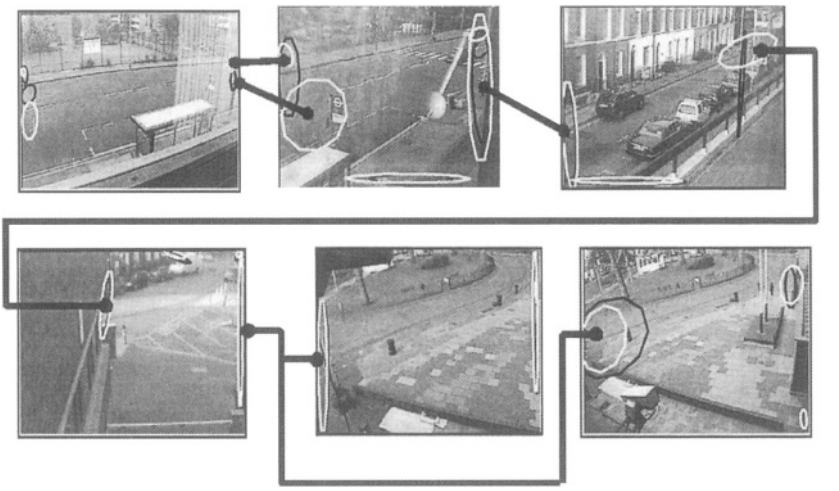


Figure 6.12. Handover regions for six cameras in the surveillance system.

The handover agent is activated when an object is terminated within an exit region X_i^k that is included in the handover region list. The handover agent records the geometric location, and time when the object left the field of view of the ith camera. Allowing the object handover agent to only be activated when an object is terminated in a handover region eliminates the case where an object is prematurely terminated within the field of view due to tracking failure caused by a complex dynamic occlusion. In addition, once the handover agent has been activated the handover region model can be used to determine the most likely regions where the object is expected to re-appear, hence reducing the computational cost of completing the handover process.

Handover Completion

The handover agent achieves completion when an object is created within the entry region E_j^l that forms a handover region with the exit region X_i^k, where the object was terminated in the ith camera view. The handover agent task is only complete if the new object satisfies the temporal constraints of the handover region. The new object location must be consistent with the temporal delay of the handover region and the transit time of when the object left and reappeared in the ith and jth camera view respectively.

Handover Termination

The handover agent is terminated once an object has not been matched after a maximal temporal delay, which can be determined by the statistical properties of the handover regions related to the exit region where the object left the field of view. The maximal temporal delay in a handover region is an important characteristic, since the surveillance regions are not constrained in such a way that an object must re-appear in the field of view once it enters a handover

region. It is possible that once the object has been terminated within an exit region it will not re-appear within the network field of view. When this case occurs it is not possible for the system to locate the object, since it is will not be visible by any of the cameras in the surveillance network.

The framework used for tracking object between non-overlapping views makes several assumptions. It is assumed that the temporal delay between the camera views is of the order of seconds for each object class. If the handover regions are located on the same ground plane and calibrated in the same world coordinate system then 3D trajectory prediction can be used to add another constraint to the data association between the handover object and candidate objects which appear in entry regions in the adjacent camera view. The 3D trajectory prediction is only valid if the object maintains the same velocity and does not significantly change direction once it has entered the handover region.

An example of the object handover reasoning process is given in Fig. 6.13. The identity of the vehicle is correctly preserved as it moves through four of the cameras in the surveillance network. The black lines represent the handover regions used to coordinate the tracking of objects between each of the views. The arrows on each line indicate the direction of the vehicle's motion between the camera views.

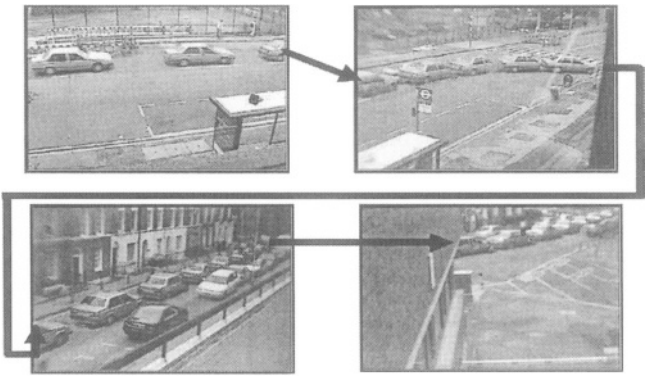


Figure 6.13. Object handover between four camera views.

5. System architecture

The surveillance system employs a centralised control strategy as was described in the introduction of this chapter. The multi view tracking server (MTS) creates separate receiver threads (RT) to process the data transmitted by each intelligent camera unit (ICU) connected to the surveillance network. Each ICU transmits tracking data to each RT in the form of symbolic packets. The system uses TCP/IP sockets to exchange data between each ICU and RT. Once the object tracking information has been received it is loaded into the surveillance

database that can be accessed for subsequent online or offline processing. Each ICU and RT exchanges data using the following four message types:

Background Image Message

This message format is used to transmit the background image from the ICU to the MTS during the initial start-up of the camera. The MTS uses the background image to visualise the tracking activity in 2D. The background image can be periodically refreshed to reflect changes in the camera viewpoint.

Timestamp Message

This message is used to transmit the timestamp of the current image frame processed by the ICU to the MTS. The MTS uses the recorded time-stamps to perform temporal alignment

2D Object Track Message

This message is generated for each detected object in the current image frame. The message includes details of the object's location, bounding box dimensions, and normalised colour components.

2D Object Framelet Message

Each detected object in the current frame is extracted from the image and transmitted to the MTS. The framelet is then stored in the surveillance database and can be plotted on the background image to visualise the activity within the field of view of the ICU.

5.1 Surveillance Database

The key design consideration for the surveillance database was that it should be possible to support a range of low-level and high-level queries. At the lowest level it is necessary to access the raw video data in order to observe some object activity recorded by the system. At the highest level a user would execute database queries to identify various types of object activity observed by the system. In order to support these real-time operational and reporting requirements we use a multi-layered database design, where each layer represents a different abstraction of the original video data. The surveillance database comprises three layers of abstraction:

- Image framelet layer
- Object motion layer
- Semantic description layer

This three-layer hierarchy supports the requirements for real-time capture and storage of detected moving objects at the lowest level, to the online query of activity analysis at the highest level. Computer vision algorithms are employed to automatically acquire the information at each level of abstraction. The physical database is implemented using PostgreSQL running on a Linux server. PostgreSQL provides support for storing each detected object in the database. This provides an efficient mechanism for real-time storage of each object detected by the surveillance system. In addition to providing fast indexing and retrieval of data the surveillance database can be customised to offer remote access via a graphical user interface and also log each query submitted

by each user. For the remainder of this chapter we focus on the image framelet and object motion layers of the surveillance database. The semantic description layer is described in the visual surveillance learning chapter.

Image Framelet Layer. The image framelet layer is the lowest level of representation of the raw pixels identified as a moving object by each camera in the surveillance network. Each camera view is fixed and background subtraction is employed to detect moving objects of interest. The raw image pixels identified as foreground objects are transmitted via a TCP/IP socket connection to the surveillance database for storage This MPEG-4 like coding strategy enables considerable savings in disk space, and allows efficient management of the video data. Typically, twenty-four hours of video data from six cameras can be condensed into only a few gigabytes of data. This compares to an uncompressed volume of approximately 4 terabytes for one day of video data in the current format we are using, representing a compression ratio of more than 1000:1.

In Figure 6.14 an example is shown of some objects stored in the image framelet layer. The images show the motion of two pedestrians as they move through the field of view of the camera. Information stored in the image framelet layer can be used to reconstruct the video sequence by plotting the framelets onto a background image.

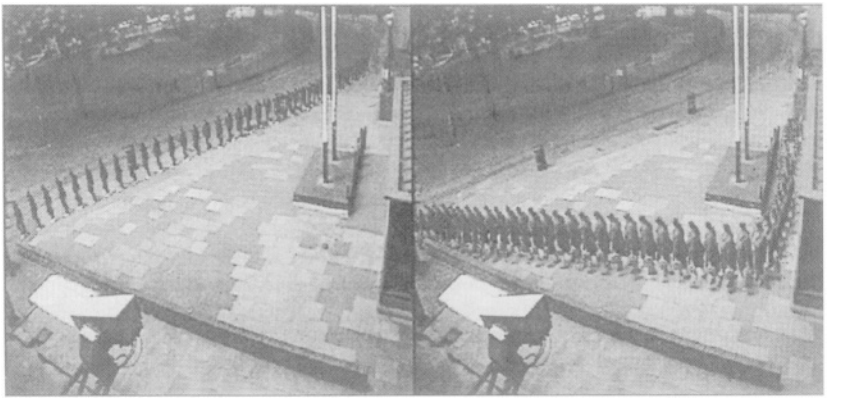


Figure 6.14. Example of objects stored in the image framelet layer.

The main attributes stored in the image framelet layer are described in Table 6.2. An entry in the image framelet layer is created for each object detected by the system. It should be noted that additional information, such as the time when the object was detected is stored in other underlying database tables. The raw image pixels associated with each detected object is stored internally in the database. The PostgreSQL database compresses the framelet data, which has the benefit of conserving disk space.

Table 6.2. Attributes stored in image framelet layer.

Field Name	*Description*
Camera	The camera view
Videoseq	The identification of the captured video sequence
Frame	The frame where the object was detected
Trackid	The track number of the detected object
Bounding_box	The bounding box describing the region where the object was detected
Data	A reference to the raw image pixels of the detected object

Object Motion Layer. The object motion layer is the second level in the hierarchy of abstraction. Each intelligent camera in the surveillance network employs a robust 2D tracking algorithm to record an object's movement within the field of view of each camera. Features are extracted from each object including: bounding box, normalized colour components, object centroid, and the object pixel velocity. Information is integrated between cameras in the surveillance network by employing a 3D multi view object tracker, which tracks objects between partially overlapping, and non-overlapping camera views separated by a short spatial distance. Objects in overlapping views are matched using the ground plane constraint. A first order 3D Kalman filter is used to track the location and dynamic properties of each moving object, which was discussed in section 4.

The 2D and 3D object tracking results are stored in the object motion layer of the surveillance database. The object motion layer can be accessed to execute offline learning processes that can augment the object tracking process. For example, a set of 2D object trajectories can be used to automatically recover the homography relations between each pair of over-lapping cameras, as was discussed in chapter 7. The multi view object tracker robustly matches objects between overlapping views by using these homography relations. The object motion and image framelet layer can also be combined in order to review the quality of the object tracking in both 2D and 3D. The key attributes stored in the object motion layer are described in Table 6.3 and Table 6.4.

In Fig. 6.15 results from both the 2D tracking and multi-view object tracker are illustrated. The six images represent the viewpoints of each camera in the surveillance network. Cameras 1 and 2, 3 and 4, and 5 and 6 have partially overlapping fields of view. It can be observed that the multi-view tracker has assigned the same identity to each object. Fig. 6.16 shows the field of view of each camera plotted onto a common ground plane generated from a landmark-based camera calibration. 3D motion trajectories are also plotted on this map in order to allow the object activity to be visualized over of the entire surveillance region.

6. Summary

We have considered in this chapter a number of the computer vision tasks that will underpin the operation of an automated visual surveillance system.

Table 6.3. Attributes stored in object motion layer (2D Tracker).

Field Name	*Description*
Camera	The camera view
Videoseq	The identification of the captured video sequence
Frame	The frame where the object was detected
Trackid	The track number of the detected object
Bounding_box	The bounding box describing the tracked region of the object
Position	The 2D location of the object in the image
Appearance	The normalized colour components of the tracked object

Table 6.4. Attributes stored in object motion layer (Multi View Tracker).

Field Name	*Description*
Multivideoseq	The identification of the captured multi video sequence
Frame	The frame where the object was detected
Trackid	The track number of the detected object
Position	The 3D location of the tracked object in ground plane coordinates
Velocity	The velocity of the object

Figure 6.15. Camera network on University campus showing 6 cameras distributed around the building, numbered 1-6 from top left to bottom right, raster-scan fashion.

The requirements placed on a real surveillance system are severe, with the need to operate over a wide range of varying illumination and weather conditions, whilst coping with the complexity of the perceptual challenges presented within the region under observation. Multiple cameras are prerequisite for most surveillance installations, and the system must be able to integrate the data de-

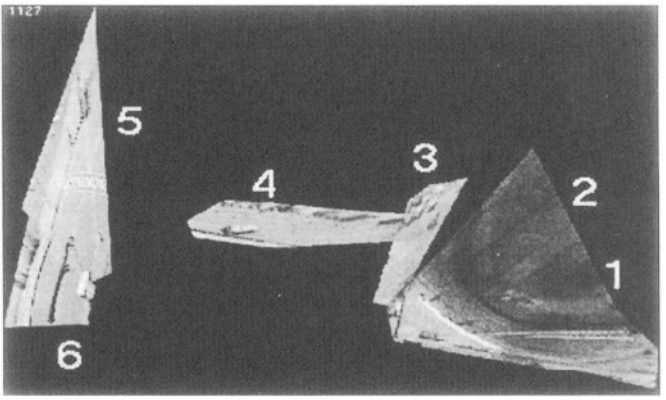


Figure 6.16. Reprojection of the camera views from Fig. 6.15 onto a common ground plane, showing tracked objects trajectories plotted into the views (white and black trails).

rived from all the cameras in order to best exploit their information gathering capabilities. Our approach is based on algorithms that can adapt to the changing conditions to ensure robust tracking of objects, minimising the effects of occlusion that would otherwise severely limit this robustness. In addition, we wish to minimise the amount of location-specific information that must be provided to the system, instead preferring to learn this information by observing the scene(s) and extracting geometric and semantic information that is used to construct models of the environment. These models are employed to augment the spatio-temporal reasoning of the tracking algorithms and will be essential aids for performing the subsequent stages of activity and behavioural analysis.

7. Appendix

7.1 Kalman Filter

A Kalman filter based on a first-order motion model is used to track each object according to the object measurement vector. The state transition and measurement equations are:

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{w}_{k-1} \quad (6.16)$$

$$\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k \quad (6.17)$$

where $\mathbf{w}_k$ and $\mathbf{v}_k$ are process noise and measurement noise, respectively, and $\mathbf{w}_k \sim N(\mathbf{0}, \mathbf{Q}_k)$, $\mathbf{v}_k \sim N(\mathbf{0}, \mathbf{R}_k)$. $\mathbf{A}$ is the state transition matrix, and $\mathbf{H}$ is the measurement matrix. The *a priori* estimate $\hat{\mathbf{x}}_k^-$, the *a posteriori* estimate

$\hat{\mathbf{x}}_k^+$ and the predicted measurement $\hat{\mathbf{z}}_k^-$ are iteratively computed by:

$$\hat{\mathbf{x}}_k^- = \mathbf{A}\mathbf{x}_{k-1}^+ \tag{6.18}$$

$$\hat{\mathbf{z}}_k^- = \mathbf{H}\hat{\mathbf{x}}_k^- \tag{6.19}$$

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{z}_k - \hat{\mathbf{z}}_k^-) \tag{6.20}$$

where $\mathbf{K}_k$ is the Kalman gain matrix that is sought to minimise the *a posteriori* error covariance $\mathbf{P}_k^+$ in a least-square sense and can also be computed with $\mathbf{P}_k^+$ and the *a priori* error covariance $\mathbf{P}_k^-$ iteratively:

$$\mathbf{P}_k^- = \mathbf{A}\mathbf{P}_{k-1}^+\mathbf{A}^T + \mathbf{Q}_{k-1} \tag{6.21}$$

$$\mathbf{K}_k = \mathbf{P}_k^-\mathbf{H}^T[\mathbf{H}\mathbf{P}_k^-\mathbf{H}^T + \mathbf{R}_k]^{-1} \tag{6.22}$$

$$\mathbf{P}_k^+ = [\mathbf{I} - \mathbf{K}_k\mathbf{H}]\mathbf{P}_k^- \tag{6.23}$$

7.2 Homography Estimation

A homography mapping defines a planar mapping between two overlapping camera views:

$$x' = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}} \tag{6.24}$$

$$y' = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}} \tag{6.25}$$

Where (x, y) and (x', y') are image coordinates for the first and second camera views respectively. Hence, each correspondence point between two camera views results in two equations in terms of the coefficients of the homography. Given at least four correspondence points allows the homography to be evaluated. It is most common to use Singular Value Decomposition (SVD) for computing the homography. The homography matrix can be written in vector form:

$$\mathbf{H} = [\ h_{11}\ \ h_{12}\ \ h_{13}\ \ h_{21}\ \ h_{22}\ \ h_{23}\ \ h_{31}\ \ h_{32}\ \ h_{33}\ ]^T \tag{6.26}$$

Each pair of correspondence points, (x, y) and (x', y'), results in two equations in terms of the coefficients of the homography matrix:

$$[\ x_i\ \ y_i\ \ 1\ \ 0\ \ 0\ \ 0\ \ -x_ix_i'\ \ -yx_i'\ \ -x_i'\ ]\mathbf{H} = 0 \tag{6.27}$$

$$[\ 0\ \ 0\ \ 0\ \ x_i\ \ y_i\ \ 1\ \ -xy_i'\ \ -yy_i'\ \ -y_i'\ ]\mathbf{H} = 0 \tag{6.28}$$

Given N correspondence points a $(2N \times 9)$ matrix M that can be constructed and then used to minimise $||M\mathbf{H}||$ subject to the constraint $||\mathbf{H} = 1||$. The value of the homography matrix can then be estimated by using Singular Value Decomposition.

7.3 3D Measurement Estimation

Given a set of corresponded objects in each camera view a 3D ray is projected through the centroid of the object in order to estimate its location. Using the camera calibration model it is possible to map the 2D object centroid to a 3D line in world coordinates.

Given a set of N 3D lines $\mathbf{r}_i = \mathbf{a}_i + \lambda_i \mathbf{b}_i$ a point $\mathbf{p} = (x, y, z)^T$ must be evaluated which minimises the error measure:

$$\xi^2 = \sum_{i=1}^{N} d_i^2 \tag{6.29}$$

Where d_i is the perpendicular distance from the point $\mathbf{p}$ to the line $\mathbf{r}_i$, assuming that the direction vector $\mathbf{b}_i$ is a unit vector then we have:

$$d_i^2 = |\mathbf{p} - \mathbf{a}_i|^2 - ((\mathbf{p} - \mathbf{a}_i) \cdot \mathbf{b}_i)^2 \tag{6.30}$$

Fig. 6.17 provides an explanation of the error measure from a geometric viewpoint. The point $\mathbf{a}_i$ is a general point located on the line, and $\mathbf{b}_i$ is the unit direction vector of the line. The distance d_i^2 is the perpendicular distance between an arbitrary point $\mathbf{p}$ and the line $\mathbf{r}_i$. The origin of the world coordinate system is defined by $\mathbf{O}$. Evaluating the partial derivatives of the summation of all d_i^2 with respect to x, y and z results in the equation for computing the least squares estimate of $\mathbf{p}$:

$$\xi^2 = \sum_{i=1}^{N} d_i^2 = \sum_{i=1}^{N} \{|\mathbf{p} - \mathbf{a}_i|^2 - ((\mathbf{p} - \mathbf{a}_i) \cdot \mathbf{b}_i)^2\} \tag{6.31}$$

Rearrangement of above equation leads to:

$$\frac{\partial \xi^2}{\partial x^2} = \sum_{i=1}^{N} \{2(x - a_{ix}) - 2(\mathbf{p} - \mathbf{a}_i) \cdot \mathbf{b}_i b_{ix}\} \tag{6.32}$$

$$\frac{\partial \xi^2}{\partial y^2} = \sum_{i=1}^{N} \{2(y - a_{iy}) - 2(\mathbf{p} - \mathbf{a}_i) \cdot \mathbf{b}_i b_{iy}\} \tag{6.33}$$

$$\frac{\partial \xi^2}{\partial z^2} = \sum_{i=1}^{N} \{2(z - a_{iz}) - 2(\mathbf{p} - \mathbf{a}_i) \cdot \mathbf{b}_i b_{iz}\} \tag{6.34}$$

$$\frac{\partial \xi^2}{\partial x^2} + \frac{\partial \xi^2}{\partial y^2} + \frac{\partial \xi^2}{\partial z^2} = 0 \tag{6.35}$$

Using matrix notation an equation can be derived to minimise the geometric error function for all N lines.

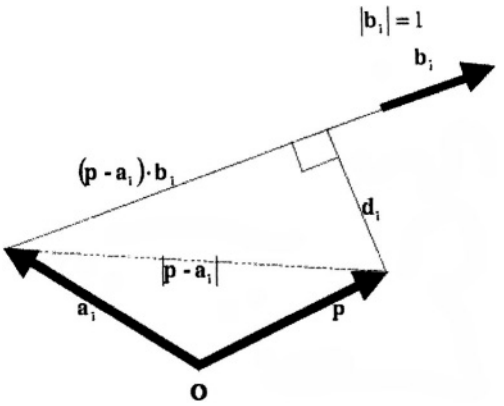


Figure 6.17. Geometric view of the minimum discrepancy.

Acknowledgments

This work was partly undertaken with support from the Engineering and Physical Science Research Council (EPSRC) under grant number GR/M58030.

References

[1] Black J, Ellis T, "Multi Camera Image Tracking", *PETS2001,* Kauai, Hawaii, December 2001.

[2] Black J, Ellis T, "Multi Camera Image Measurement and Correspondence", *Measurement,* 32, 2002, pp. 61-71.

[3] Buxton H, Gong S, "Visual surveillance in a dynamic and uncertain world", *Artificial Intelligence,* 78: pp. 431-459, 1995.

[4] Ellis T, "Co-operative computing for a distributed network of security surveillance cameras", *IEE European Workshop. Distributed Imaging* (Ref. No.1999/109). IEE, London, UK; 1999; 136, pp. 10/1-5.

[5] Ellis T, Xu M, "Object detection and tracking in an open and dynamic world", *PETS2001,* Kauai, Hawaii, December 2001.

[6] Ellis T, "Performance Metrics and Methods for Tracking in Surveillance", *PETS2002,* Copenhagen, pp. 26-31, May 2002.

[7] Ellis T., Makris D. and Black J., "Learning a multi-camera topology", Proc. *IEEE Joint Int. Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance,* 2003.

[8] Hartley R, Zisserman A, "Multiple View Geometry in Computer Vision", Cambridge University Press, 2001.

[9] Heckerman D., "A tutorial on learning with Bayesian Networks", *Technical Report,* MSR-TR-95-06, Microsoft Research, 1995.

[10] Lee, Romano R, Stein G, "Monitoring Activities from Multiple Video Streams: Establishing a Common Coordinate Frame". *IEEE Trans, on PAMI,* Vol. 22, No. 8, August 2000, pp 117-123.

[11] Mikic I, Santini S, Jain R, "Video Integration from Multiple Cameras", *DARPA Image Understanding Workshop,* Monterey, CA November 1998.

[12] Pearl J, Probabilistic Reasoning in Intelligent Systems: Network of Plausible Inference, Morgan Kaufmann, San Mateo, CA, 1988.

[13] Stauffer C, Grimson W E L, "Adaptive background mixture models for real-time tracking", *Proc. CVPR'99,* 1999, pp. 246-252.

[14] Stein G, "Tracking from Multiple View Points: Self-calibration of Space and Time". *DARPA IU Workshop,* 1998, pp 1037-1042.

[15] Tsai R, "An efficient and Accurate Camera Calibration Technique for 3D Machine Vision", *IEEE Conf. on CVPR,* June 1986, pp 323-344.

[16] Xu M, Ellis T, "Illumination-invariant motion detection using colour mixture models", *Proc. BMVC,* pp. 163-172, Manchester, Sept. 2001.

[17] Xu M, Ellis T, "Partial observation vs. blind tracking through occlusion", *Proc. BMVC,* pp. 777-786, Cardiff, 2002.

[18] Xu M, Ellis T, "Tracking occluded objects using partial observation", *Acta Automatica Sinica,* special issue on Visual Surveillance of Dynamic Scenes, 29(3), pp. 370-380, 2003.

Chapter 7

LEARNING AND INTEGRATING INFORMATION FROM MULTIPLE CAMERA VIEWS

Mapping an Ambient Environment

D.Makris, T.Ellis and J.Black
Digital Imaging Research Centre (DIRC), Kingston University, UK
{d.makris,t.ellis,j.black}@kingston.ac.uk
Keywords: Activity models, scene recognition, visual learning, video surveillance.

1. Introduction

The primary goal of visual surveillance and monitoring is to understand and interpret the activities of objects of interest (typically people and vehicles) within an environment. The current deployment of such systems mainly fulfils a role of security surveillance, providing remote viewing and recording of video information from CCTV cameras to help protect people and property, detecting illegal, anti-social or invasive actions. However, as such systems become more pervasive in urban environments, their potential is to be used to provide a much wider range of interactive services that can both facilitate and anticipate the needs and wants of people.

A fully functioning system would be capable of recognising and predicting the behaviour of individuals and groups, employing specific models representing the activity patterns. This should also enable the system to understand the typical or expected behaviours within a particular environment, and then to be capable of identifying unusual or atypical behaviours. To do so, the system would benefit from contextual knowledge relating to the environment. Such knowledge might be both local and global. For instance, consider the activities associated with a ticket office in a railway station, and its local environs. The office will generate particular types of activity as people enter the scene from (often) well-defined regions and move towards the office. At busy times people will need to queue to access the ticket seller(s), undertaking their transaction then departing to some other part of the station.

It is also desirable for this high level knowledge to be derived by the system itself, rather than be given explicitly by a human operator. Therefore, the system

can be installed with a minimum effort ("plug'n'play") and it can adapt to any changes of its environment. Such a capability is very useful, especially for large scale distributed surveillance systems where the calibration of newly installed cameras or recalibration of old cameras can be a time-consuming and skilled task.

For pedestrians we can begin by identifying a number of 'primitive' activities such as walking, sitting and queuing that can be inferred from tracking data. Whilst these activities can be derived for individual objects, it is valuable to be able to recognise typical activities, commonly performed by many objects, and to relate these to scene-dependant contextual knowledge. It is interesting to note that many of the behaviours that we might want to identify as part of a surveillance or monitoring task occur when the objects are stationary (or near-stationary), so the detection of 'stopped' objects is a significant event in the tracking process.

The interaction of people with the environment is constrained by the spatial geometry. Most modern man-made environments are ergonomically designed to facilitate efficient access to popular locations, anticipating flow-rates and densities to minimise the effects of over-crowding, or to ensure that there are no isolated areas where individuals may feel vulnerable or threatened. Whilst it is possible to incorporate geometrical models of an environment into a surveillance recognition system, such models would need further augmentation to express the typical activities of people interacting with the environment. Furthermore, these activities are not stationary, in the sense that at different times of the day, week or year, they can change. In the case of the ticket office example, there will be certain times when the office is closed and when people enter the scene, their actions will be quite different.

In this chapter we develop models to represent the common activities that are observed within an outdoor scene. The models will be learnt from an analysis of trajectory data extracted from the scene over a long observation period (10-20 hours). The models combine spatial and probabilistic information to create a representation that can be used to describe both the spatial geometry associated with the activity, as well as a statistically-derived likelihood of usage. The models are used to encode regions in the imaged scene where various primitive activities occur: objects entering or leaving the camera view (entry and exit zones); the common paths or routes taken by objects moving through the scene; and stop zones, where objects come to rest for some minimum period.

The models can be used to support a variety of different uses in the online surveillance system, such as annotation and atypical behaviour detection. In annotation, individual trajectories can be described using only a small number of parameters associated with a particular activity or set of activities: for example: "object #27 entered the scene at entrance C at 16:51:20, moved along path 4, stopped at location L2 from 16:51:56-16:52:11, then proceeded to exit the scene from exit A at 16:52:44". Such annotation provides a meta-data layer in the trajectory database, allowing a more efficient means to describe objects and create global usage statistics of activities. In addition, the annotation can be combined with specific contextual scene knowledge (e.g. that location L2 corresponds to the region in front of the ticket office) to semantically enrich the

annotation and facilitate human-centric queries to the database (e.g. extract all visits to the ticket office between 11:30-12:30). The typical behaviour of objects is encoded into the probabilistic component of the model, and can be used in combination with the spatial component to identify an action that is unusual, at least across the learning set from which the models were constructed.

We also employ the entry/exit models for each camera view to automatically learn the topology of an arbitrary network of video cameras observing an environment. The topology is learnt in an unsupervised manner by temporally correlating objects transiting between adjacent camera viewfields, establishing the correspondence of links between the entry and exit zones of cameras in the network. A significant benefit of the method is that it doesn't rely on establishing explicit correspondence between trajectories, and results in a measure of inter-camera transition times, which can be used to support predictive tracking across the camera network.

The trajectory data used to learn these models is extracted by querying the tracking database described in chapter 6.

1.1 Semantic Scene Model

A semantic description of activity is usually given in relation to semantic elements of the scene, e.g.: "John entered the house from the front door, walked along the corridor, sat at the desk and then left from the back door". However, to allow automatic description of such activities (video annotation) from visual surveillance, semantic labels, like "front door", "corridor", "desk" must be defined. Ideally, these features would be automatically recognised by the visual surveillance system.

In the context of this chapter, semantics are defined in relation to the activity of targets. For instance, doors are characterised as a feature associated with entry/exit events, a desk is an element of interest that targets may stop nearby and a pavement is a common path that pedestrians move along.

A semantic model of the scene is introduced. The semantic labels of the scene regions are associated with the activities that are performed on these regions. The model must provide both a spatial and probabilistic representation in order to characterise the target activity in terms of spatial features of the scene and express the level of usage and the associated uncertainty, supporting a predictive capability. The model includes regions associated with a particular semantic interpretation, such as entry/exit zones, paths, routes, junctions and stop zones.

Figure 7.1 shows a simplified depiction of an outdoor scene, consisting of a number of interconnected pathways. The seat icon (I, J) represents regions where targets may stop, while other labels are associated with entry/exit regions (A, C, E, G, H) and junctions (B, D, F) where targets moving along the pathways may change their routes. The segments between entry/exit zones or/and junctions (AB, CB, BD, DF, FG, ...) represent paths, while routes are represented by the sequence of paths between an entry zone and an exit zone (ABDFH, CDBFE, EDFG,...).

Entry zones are regions where targets enter the scene. Similarly, exit zones are regions where targets leave the scene. An entry zone and an exit zone may

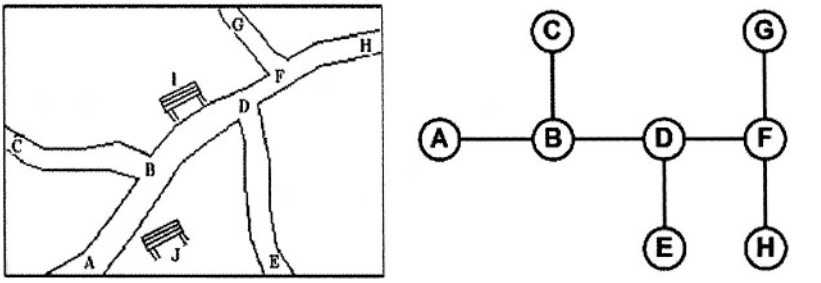


Figure 7.1. (a) Topographical map of the scene, (b) topological map of the scene. Entry/exit zones and junctions are the nodes of the network.

be coincident, e.g. as in most pedestrian environments, or not e.g. in road traffic environments, where traffic is constrained by road traffic regulations. Typically, entry/exit zones are either scene-based, e.g. doors and gates, or view-based, e.g. located parts on the boundaries of the camera view. The distinction between scene-based and view-based entry/exit zones can be useful when information from multiple cameras is integrated.

Junctions are the areas where two or more pedestrian pathways or roads meet. At junctions, there is an uncertainty about the future motion of a target, as it can follow more than one path.

Whilst in common English, paths and routes have similar meanings, they are distinguished in the context of this chapter in the following manner. Paths are segments of either pedestrian pathways or roads in between entry/exit zones and junctions. A target route is the complete history of a target activity around the scene, from its entry zone to its exit zone, through various paths and junctions. More precisely, paths should be referred to as path segments, but the above given definitions are kept for the sake of simplicity.

Stop zones are defined as the regions where targets are stationary or almost stationary, for some minimum period of time. For example, pedestrians are stationary when they stop in order to sit, rest, queue, wait to access a resource, merely observe the scene or just wander around. Stop zones are included in the scene model for two reasons: firstly, a stop zone is usually related to a physical scene feature, such as a bus stop, an ATM machine, a park seat, a shop window, a cashier, a computer, a printer, etc. Secondly, although the majority of research in video surveillance has focused on detecting and tracking motion, it is actually when targets stop and interact with each other or with these fixed elements of the scene that the system is more likely to be interested in them.

Two different presentations of the scene model are suggested: topographical and topological. The scene model is naturally represented by a topographical map (Figure 7.1a) based on either an image plane(s) or a ground map representation. Ground map representations have an advantage over image-based representations not only because they can represent the physical features of the

scene in proper proportion, but also because they allow integration of information from multiple cameras.

The scene model can be also represented by a topological map, i.e. an abstract network of nodes and connections, as shown in Figure 7.1b where entry/exit zones and junctions comprise the nodes of the network connected by paths. Augmented network representations can be constructed to include additional semantic feature, such as paths and stop zones.

The two different representations are used to illustrate two different aspects of the model. More specifically, the topographical map visualises the spatial characteristics of the scene elements, as interpreted by a human, whereas the topological map can be the basis of a probability network that can be used for a probabilistic analysis of the activity.

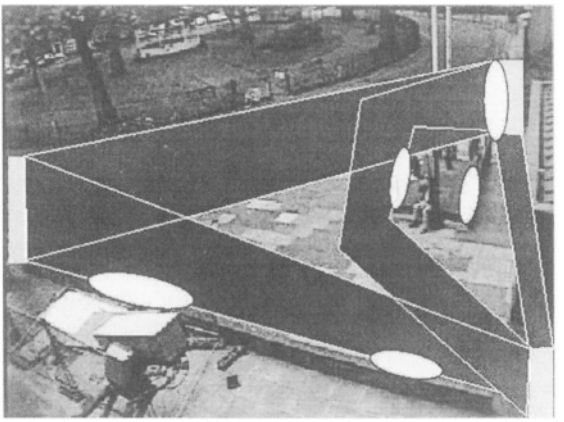


Figure 7.2. A manually derived semantic description for regions exhibiting common activities in the observed scene. The boxes at the edges correspond to the entry and exit areas of the scene, the closed polygons to commonly used paths and the ellipses to areas where pedestrians normally stop.

Spatio-probabilistic representations are learnt for the different scene semantics. Entry, exit and stop zones are related to single-point events, therefore sets of single points are used as training data. Routes, paths and junctions are related to motion that is represented by sequence of points (trajectories), therefore trajectory sets are used as training data.

2. Learning point-based regions

Targets enter and exit the scene from either the borders of the image (view-based feature), or at doors or gates (scene-based feature). The first and the last successfully tracked positions of an object (in other words the first and the last point of its observed trajectory) are used to indicate the entry and the exit event. The distribution of trajectory start coordinates for a given entry zone

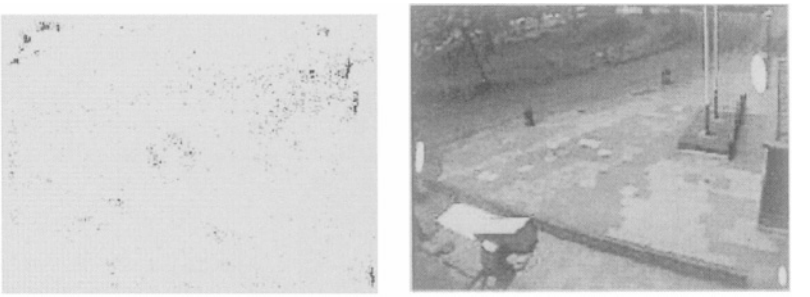


Figure 7.3. (a) Entry-point dataset (4250 samples) with both types of noise present, (b) Three entry zones derived by the multi-step algorithm. Dense clusters in the top left are eliminated. Clusters are represented as ellipses at one standard deviation.

is influenced by the speed of targets across the image plane as they enter the camera viewfield. This is dependent on the target's actual speed in the scene, the video frame capture rate and the direction of motion with respect to the camera view. Clustering many such observations allows a region of the scene to be associated with a given zone.

We choose to model the shape of the entry/exit zones using Gaussian Mixture Models (GMMs), because they compactly represent the variability and the uncertainty of the observed entry/exit events. Figure 7.3 depicts the entry/exit point datasets and fitted GMMs for a typical outdoor scene.

The entry-point/exit-point datasets can be contaminated by two different types of noise: a) tracking failure noise is caused by the failure of the tracking algorithm to continuously track objects. It is usually distributed over all the activity areas; b) semi-stationary motion noise is caused by the apparent motion of objects (trees, curtains, computer screens). It is usually densely-distributed within small areas of the image.

We conjecture that trajectory start and finish coordinates are spatially correlated with localized image regions, whilst tracking failure is spatially uncorrelated. Hence, the regions with dense observations are more likely to correspond to real entry/exit zones, whilst tracking failure noise is identified by wide Gaussian distributions over the activity areas. In contrast, although the semi-stationary motion noise can be identified by tight Gaussian distributions, they can be characterized as exhibiting very short trajectory paths, typically starting and finishing within a small area. These criteria can be used to filter the noise clusters from the signal (entry/exit zone) clusters.

We use a multi-step learning method that is based on the Expectation-Maximization (EM) algorithm [4] to learn the clusters. One of the advantages of EM is that it can successfully distinguish overlapped distributions. Therefore, if noise is overlaid onto signal, then it is possible to generate separate clusters for the signal and the noise, subject to different statistics. The two types of

noise that are present in the trajectory data are also found in the entry/exit-point datasets and they have distinguishable statistical characteristic.

The multi-step algorithm first discards the semi-stationary noise and then the tracking failure noise. The actual number of entry/exit zones in the scene is unknown, however EM is a parametric method and the model order must be predefined. In order to overcome this problem, the number of entry/exit zones in the scene is initially (conservatively) overestimated and the number of the signal clusters is estimated by eliminating the noisy ones. A summary of the algorithm is given below:

1 The EM algorithm with model order N is applied to the entry-point dataset E and a GMM is derived.

2 If all the points of a trajectory belong to a single GM, as derived in the previous step, the trajectory is considered semi-stationary. A new cleaned entry-point dataset **E'** is formed that does not contain the entry points of the semi-stationary trajectories.

3 The EM algorithm with model order N' is applied to the clean entry-point data-set **E'**.

4 Gaussian clusters are classified as either signal clusters or noise clusters, according to a density criterion. More specifically, if $omega_i$ is the prior probability of a cluster i and Σ_i is its covariance matrix, where i=1.. N', then a measure of the density d_i is given by:

$$d_i = \frac{\omega_i}{\pi\sqrt{|\Sigma_i|}} \tag{7.1}$$

A threshold value T is defined by the clean entry-point dataset **E'**:

$$T = \frac{\alpha}{\pi\sqrt{|\Sigma|}} \tag{7.2}$$

where α is a user-defined weight and Σ is the covariance matrix of the dataset **E'**.

The clusters derived at the steps (1) and (3) are characterised according to their density. High-density Gaussian clusters correspond to either entry zones or semi-stationary motion noise, while low-density clusters correspond to tracking failure noise. The algorithm eliminates semi-stationary motion noise at step (2) and tracking failure noise at step (4).

Results for a camera viewing a road traffic environment are shown in Figure 7.3. Two entry zones and two exit zones were identified that are non-coincident, because they correspond to two unidirectional road lanes. (Although in this scene pedestrian traffic is normally present, the dataset was derived over a weekend, when pedestrian traffic was very light.)

In Figure 7.4, the clusters at the left side of the images are wider than the ones at the right side. The is due to the higher image plane speed of the vehicles

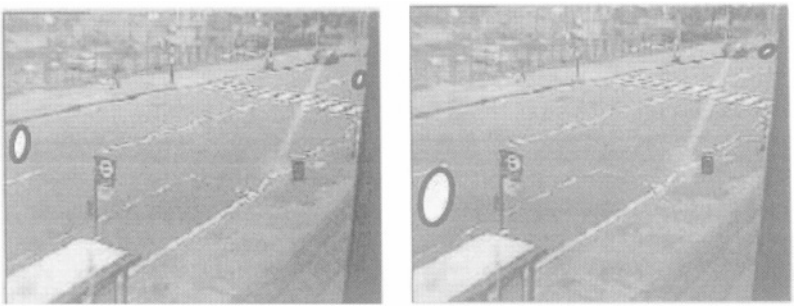


Figure 7.4. (a) Two detected entry zones in a road traffic environment (b) Two detected exit zones in a traffic environment road.

at the left side (closer to the camera), which, in combination with a low frame rate (5fps), results in a wider distribution of the first/last tracked positions for the vehicles. This difference of the distributions of the entry/exit zones is actually desirable, as the cluster explicitly determines where the targets should be initialised/terminated, according to their entry/exit speed and the system frame rate.

Stop zones are defined as regions where the targets are stationary or almost stationary. A variety of different areas can be characterised as stop zones, such as where people rest, wait for the opportunity to continue their journey (e.g. at a pedestrian crossing or a road traffic junction), access a particular resource (e.g. an automatic teller machine or at a bus stop), or merely observe the scene. Targets may also become stationary when they meet and interact with other targets, for example two people meeting in a park and sitting on a bench to chat or a vehicle waiting for a pedestrian to walk across a pedestrian crossing.

A stop event is detected when a target's speed becomes lower than a predefined threshold. A stop-event dataset is formed by checking the target trajectories for stop events. Speed is estimated in ground plane coordinates because the apparent speed on the image plane may be strongly affected by the perspective view of the camera. Therefore the stop-event dataset is formed using ground plane coordinates (see Figure 7.5).

As with the entry/exit zones, a GMM is used to model the spatio-probabilistic characteristics of the stop zones and EM is applied to learn them.

3. Learning trajectory-based regions

3.1 Route model

In road traffic environments, vehicles must follow specific predefined routes. Similarly, in pedestrian environments, people normally walk on well-prescribed pathways. Even in cases where no predefined routes exist, the structure of the

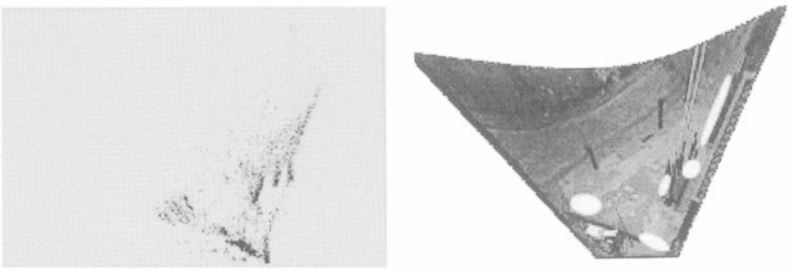


Figure 7.5. (a) Stop events (9455) on the ground plane. Stop events were detected when targets' speed drop lower than 0.25m/sec. (b) Five signal stop zones as derived by the EM algorithm.

scene affects the behaviour of pedestrians and normal route-patterns of activity exist, which is verified by results presented in this section.

A route model that is both consistent with the human interpretation and fulfils the requirements for probabilistic analysis is required. A route can be described intuitively by its start and end areas, its main axis between the start and the end and its boundaries along the main axis. Additionally, quantitative information about the usage along and across the route is required to describe the statistics of typical usage.

The scene is assumed to contain multiple routes that may have overlapped sections. A single route model must encode the following properties, using both spatial and probabilistic representation:

- The main axis of the route.
- The terminators (start and end points) of the route.
- A description of the width along the route.
- Indication of the level of usage of the route, both along and perpendicular to the direction and in comparison to the other routes.

The route model that we have developed (see Figure 7.6) consists of a discrete representation defined by a central spline axis, composed of a sequence of equidistant nodes that represents some average of the route. The constant distance between adjacent nodes is referred to as the resample factor R of the model. In addition, two bound splines around the central axis form an envelope and represent the width of the path. A route has two terminator nodes (start and end) that typically correspond to entry/exit zones of the scene. Finally, a weight factor represents the usage frequency of the route.

This route model allows explicit representation of the spatial extent of routes and therefore it is consistent with the semantic representation requirement. In addition, the probabilistic representation of the usage, both along and across the route, and the discrete nature of the model allow direct deployment of a probabilistic network (e.g. a HMM).

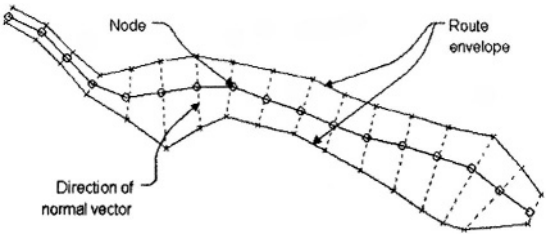


Figure 7.6. Spatial representation of the route model.

3.2 Learning algorithm

The input data of the algorithm is a set of trajectories, derived by a motion tracking algorithm that estimates the location of the centroid of the moving objects, from a single fixed camera. It is desirable to learn paths using representative trajectories unconstrained by tracking failure. For this reason, short trajectories or trajectories with many sudden changes of direction are filtered. Further validation of the trajectory dataset is based on knowledge of the entry/exit zones. Specifically, trajectories are accepted only if they start from a valid entry zone and terminate at a valid exit zone.

The model order of the scene routes is not defined explicitly, but it is determined implicitly by the algorithm parameters and the dataset. The first trajectory of the dataset initialises the first route model. Other route models will also be initialised automatically by trajectories that do not match an existing model.

Theoretically, there is no restriction in the number of route models of the scene. In practice, route models with very low usage are discarded through the learning process, for computational efficiency.

A summary of the learning algorithm is as follows:

1 The first trajectory of the dataset initialises the first route model

2 Each new trajectory is compared with the existing route models.

3 If a trajectory matches a route model, then the route model is updated.

4 If a trajectory does not match to any route model, a new model is initialised.

5 The updated route model is resampled, so inter-node distances are kept equal to R.

6 Each updated route model is compared with the other route models.

7 If two route models are sufficiently overlapped, they are merged.

The algorithm requires two parameters: a) the resample factor R which defines the level of detail for each route model. Very small values for the

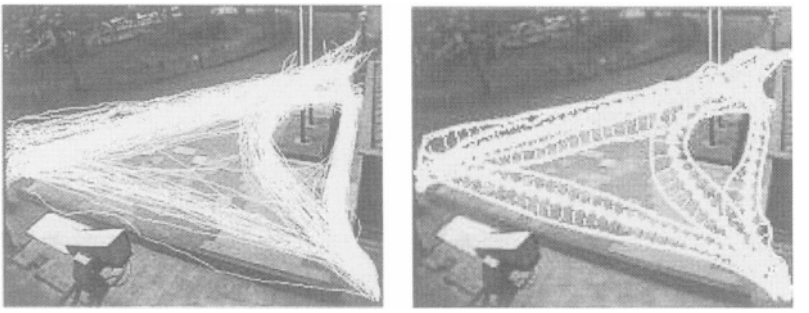


Figure 7.7. (a) Dataset of 752 trajectories, (b) Set of five route models, as derived by the route learning algorithm, for R=20 pix and T=30 pix.

resample factor are not recommended, because this selection can make the algorithm computationally expensive without significant benefit. b) a distance threshold T which defines the minimum allowable gap between different routes. Its recommended value is related to the quantity of learning data; specifically the less data the larger the value of T.

3.3 Segmentation to paths and junctions

Intuitively, a junction is the area where two routes cross. A more rigorous definition is adopted here: a junction is the region of intersection of two routes, where route directions differ by more than an angle ω. This definition reflects the fact that while a target is on a junction, some uncertainty is raised about its future direction.

Paths are considered as route parts between junctions and/or entry/exit zones. Paths may also relate to route overlapping. If route parts are overlapped and the route directions are similar along the overlapped route parts, their union represents a path. For instance, the upper path of Figure 7.8b is formed by two route parts with similar direction.

Accumulative statistics could be used to identify the areas where target directions are similar (paths) or different (junctions). However, from the above definitions, it can be concluded that junctions and paths are closely related to the geometry of the scene route models; therefore, they can be easily extracted from a set of route models, using computational geometry and constraints on the direction of the routes.

The route models of Figure 7.7 are visualised as polygons in Figure 7.8(a). Junctions are detected in regions where the direction of the routes differs by more than 5° and results are shown in Figure 7.8(b). Junctions are used to split the route models into paths. Figure 7.8b also visualises the segmentation of routes to paths and junctions.

The partitioning of routes to paths and junctions is performed not only to identify semantic features, but also to support activity analysis. For example,

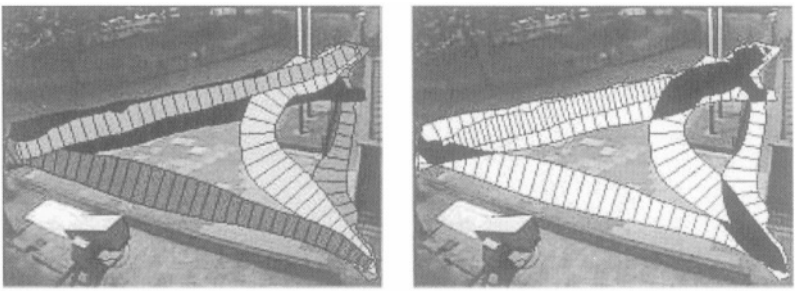


Figure 7.8. (a) Five routes detected by the route learning algorithm. (b) Segmentation of the scene route models to paths (white areas) and junctions (black areas).

entry/exit zones, paths and junctions can be considered as primitives of routes and rare, complicated routes can be described as sequences of these primitives. Also, junctions are regions of interest for a long-term prediction module, because a targets' motion within junctions reflects the targets' intention to move towards specific regions.

4. Activity analysis

Target activity can be related to elements of the scene model. For example, a target will enter the scene by an entry zone, follow a path, then reach a junction and take another path, stop at a stop zone and finally exit the scene through an exit zone. If it is detected outside the scene model, its activity will be characterised as atypical.

The discrete character of the scene model, as illustrated in a topological representation, allows discrete-state models like Markov Chains and Hidden Markov Models (HMM) to be applied. Both Markovian models can be used for activity modelling and analysis. The suggested approach is HMM-like for two reasons: firstly, HMMs can distinguish observations and states and model the uncertainty of correspondences of observations to states using membership functions. Secondly, the probabilistic nature of each of the scene elements allows the required membership functions to be easily determined for each of the states of the model.

Two network representations can be derived by the scene model. The first consists of scene elements like entry/exit zones, paths, junctions and stop zones. The second consists of all the nodes of all the scene route models. HMMs can be overlaid onto both types of network and can be used for activity analysis.

A Route-Based Hidden Markov Model (RBHMM) [6] describes a HMM that is superimposed on the set of all the nodes of the route models of a scene and is used to detect atypical activities. For instance, while Figure 7.9a depicts a common trajectory, Figures 7.9b-c depict trajectories that are atypical, according to the RBHMM model of the scene.

Figure 7.9. Three trajectories are shown. The left trajectory is a very common one. The middle one contains two rather atypical examples ('X' symbols). The right one is a very uncommon one of somebody 'climbing' (actually a tracking association error).

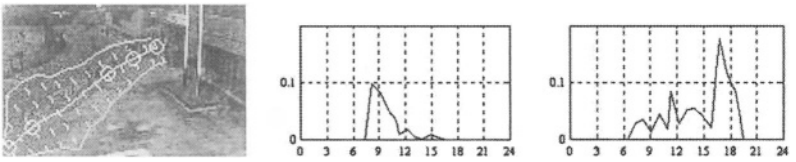


Figure 7.10. (a) Route model. (b) Probability that a pedestrian will move along the route, towards the entrance. (c) Probability that a pedestrian will move along the route, away the entrance. The probabilities are given for a 24-hour period. x-axis indicates the time of the day.

However, deciding whether an event is typical or atypical must involve information of the time at which it occurs. For instance while the trajectory of Figure 7.9 can be assumed as typical at 9am, it should be considered as atypical if it occurred at midnight.

Therefore, the RBHMM should be time-variant to reflect the variability of the expected activity over time. Figure 7.10 illustrates this variability of usage of a specific route. Figure 7.10b shows of the probability that a pedestrian will use the route to move towards the entrance, while Figure 7.10c shows the probability that a pedestrian will use the route to move away the entrance. At 8-9am, almost all trajectories occur towards the entrance of the University, whilst around 5pm, people tend to be leaving the building.

5. Integration of information from multiple views

Multi-camera surveillance systems cover wide-area scenes and aim to track targets within this scene. The key issue in these systems is to effectively integrate information from multiple cameras in order to provide complete histories of targets' activities within the environment, sampled by multiple cameras. To integrate information in the spatial domain, a world coordinate system is required. Usually, a ground plane coordinate system [16] is used which is consistent with the assumption that all the scene activity is coplanar. A ground plane map is used to illustrate results in ground plane coordinates. For example, Figure 6.12 of a later chapter illustrates a ground map constructed from geometric calibration models and views from six cameras. Four different field

of views (FOV) are defined on the ground plane, to determine how the scene is viewed by a camera surveillance network:

- Visible FOV defines the regions that a single camera images, excluding occluded areas and obscured areas where activity cannot be interpreted.
- Camera FOV encompasses all the regions within the camera view, including occluded regions
- Network FOV is the union of all the visible FOVs of all the cameras of the network.
- Virtual FOV covers the network FOV and all the gaps in between the visible camera FOVs, within which targets may exist.

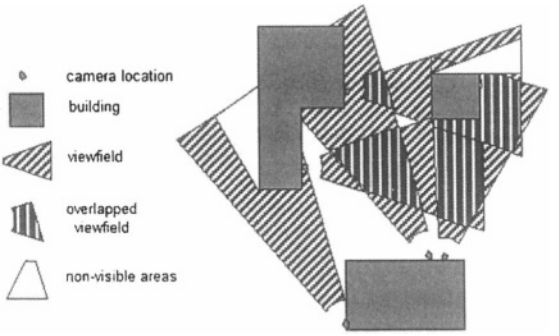


Figure 7.11. Visible FOVs of a network of cameras.

The visible FOVs of different cameras are related with different types of associations, depending on their spatial locations and how activity is seen. Specifically, two FOVs may be a) overlapped, i.e. they have common parts and a target may be seen simultaneously by the two cameras, or b) adjacent, i.e. they do not have common parts, but they are relatively close, such that targets exiting one FOV may enter the other FOV, c) distant, i.e. they are far apart and there is no direct relationship of the targets activities in these two views. This chapter uses the term "camera network topology" to represent the set of all the relationships of the cameras of the network.

Activity scene models can be applied to both the individual camera views and the common ground plane. Trajectory data has been converted to ground plane coordinates, and used as input to the entry/exit zones learning algorithm and the route learning algorithm. Results are illustrated in Figure 7.12.

Two issues are related to the ground plane approach of constructing integrated models for the entire covered scene. Firstly, the method requires explicit geometric calibration of all the cameras of the system. Secondly the model covers only the network FOV, failing to represent activity on the gaps of the

virtual FOV. The next section introduces a technique that deals with these two issues.

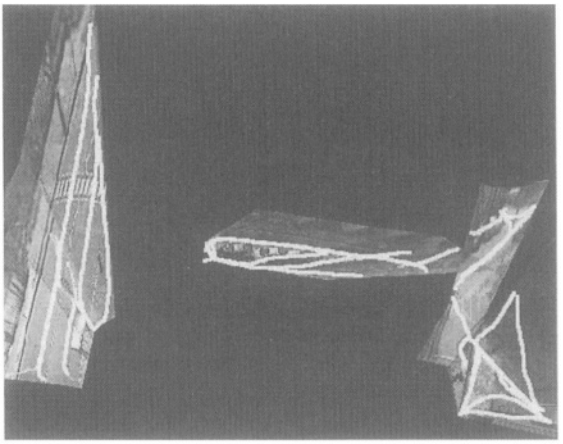


Figure 7.12. Routes learnt on a ground plane map.

5.1 Multiple Camera Activity Network (MCAN)

In the previous section, semantic scene models have been derived for individual camera FOVs and for the network FOV. However, it is desirable to extend the scene model to the entire virtual FOV, to cover activities that occur within the "blind" areas of the system. Although these activities are not directly viewed by the system, reasonable assumptions are derived.

Lets first assume that no geometric information exists that allows localisation of the camera FOVs. Cameras with overlapped visible FOVs can be identified [1] [15] and calibrated using a homography. A common semantic scene model can be derived for each set of cameras with overlapped FOVs. However, possible gaps between cameras do not allow the semantic scene model to be established for the whole system, and isolated scene models are derived instead [14].

To overcome the lack of a common scene model for the virtual FOV, the isolated scene models must be linked. However, no spatial linking is achievable, due to the lack of geometric calibration. Instead, a probabilistic-temporal linking of the isolated views is proposed.

All the entry/exit zones of the camera FOVs are represented collectively as a network of nodes.The links of the network represent transitions between the entry/exit zones, either visible (through the Network FOV), or invisible (through the "blind" areas). A markovian chain or a HMM can be overlaid on the topology representation, to create a probabilistic framework for activity analysis and long-term predictions.

Visible links are learnt by the route learning algorithm using trajectories derived by a single-camera tracker or overlapped multiple camera tracker and are physically represented in spatial terms, according to the route model

However, the challenge is to identify the invisible links and this is the focus of the method proposed in this section. Invisible links are estimated in temporal terms and more specifically by pdfs that show the distribution of the target transition periods through the blind areas.

Theoretical formulation. A MCAN is formulated by the set of all entry/exit nodes that are detected within all the cameras of the system. No information is provided regarding the spatial relationship of these nodes. It is required to identify the directional links of this network expressed in probabilistic-temporal terms, which represent target transitions from one node to the other.

A graph model (shown in Figure 7.13) is used to represent a possible link between two nodes, i and j. Targets disappear from the node i with rate $n_i(t)$ and appear at the node j with rate $m_j(t)$. A third virtual node k represents everything out of the nodes i and j. Targets transit from node i to node j in time τ with probability $\alpha_{ij}(\tau)$, otherwise they transit to the virtual node k with probability α_{ik}.

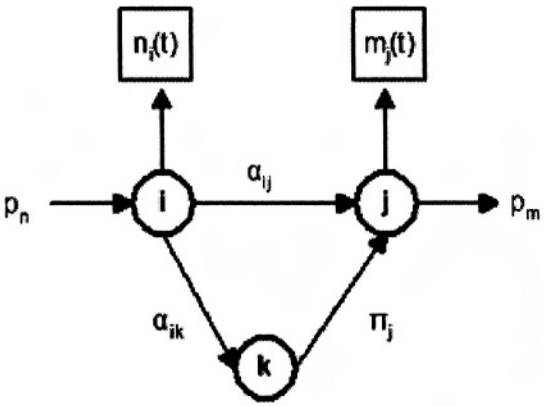


Figure 7.13. Graph model showing the probabilistic links between two nodes (i and j) and the virtual node k.

Transition probabilities fulfill the equation:

$$\alpha_{ij}(\tau)d\tau + \alpha_{ik} = 1 \tag{7.3}$$

Also, targets from the node k move to node j with rate π_j; i.e. new targets can be "generated" at node k, and are detected on entering at node j.

The surveillance system is able to observe the signals $n_i(t)$ and $m_j(t)$. The two signals are assumed individually and jointly stationary. Therefore, the

cross-correlation function is defined:

$$R_{ij}(\tau) = E[n_i(t)m_j(t+\tau)] \tag{7.4}$$

If it is assumed that the two signal $n_i(t), m_j(t)$ are taken digital values from the set 0, 1, then $p_n = pn_i(t) = 1 = En_i(t)$ and $p_m = pm_j(t) = 1 = Em_j(t)$.

The cross-correlation $R_{ij}(\tau)$ and the covariance $C_{ij}(\tau)$ defined as:

$$C_{ij}(\tau) = R_{ij}(\tau) - p_n p_m \tag{7.5}$$

and are used to identify possible links. If

$$C_{ij}(\tau)d\tau = 0 \tag{7.6}$$

then the two signals are uncorrelated and, because according to the proposed graph model, their relationship can be only linear, they are independent [12] and no real link should exist between them. Otherwise, the two signals are dependent and a valid link i®j must exist. In this case, the transition probability is estimated by the formula:

$$\alpha_{ij}(\tau) = \frac{C_{ij}}{p_n(1-p_n)} \tag{7.7}$$

Summarising, the MCAN is defined by the nodes, expressed as Gaussian distributions on the separate camera views and directional links between nodes, defined by transition probabilities that depend on the transition time.

The topology of the camera views is determined by the set of valid links and their transition times. If a link is detected between the zones of two cameras, the two cameras are either adjacent or overlapped. If the transition time between the exit zone i and the entry zone j is approximately zero, then the two zones of the two cameras are overlapped. If the transition time is positive, then the targets move from one zone to the other through an invisible path. Finally, if the transition time is negative, then the targets move from one zone to the other through a path that is partially or entirely visible by the two cameras.

6. Database

The surveillance database supports several activities ranging from real-time storage of tracking data to allowing a user to recall certain types of object activity. In the previous chapter on distributed multi-view tracking we described how these requirements were implemented using a hierarchical database structure [2]. The image framelet layer stores the video associated with detected objects, and the object motion layer contains the tracking data of each object observed by the system. In this chapter we describe the layers of the database that facilitate the reporting requirements of the database.

Semantic Description Layer. The object motion layer provides input to a machine-learning algorithm that automatically learns a semantic scene model, which contains both spatial and probabilistic information. Regions of activity

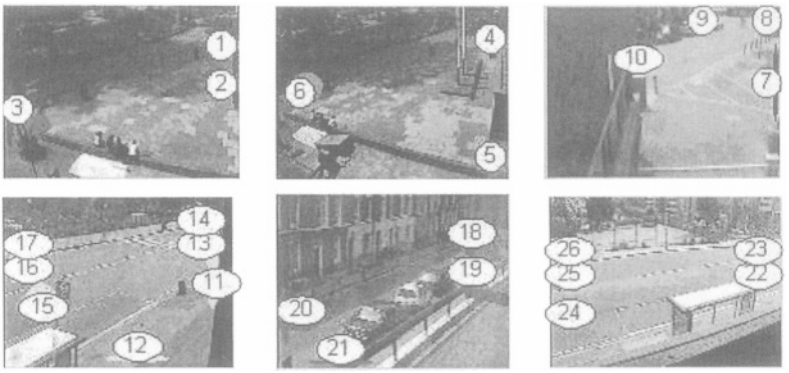


Figure 7.14. The detected entry/exit zones for the six cameras of the network. The zones are numbered as nodes of the activity network.

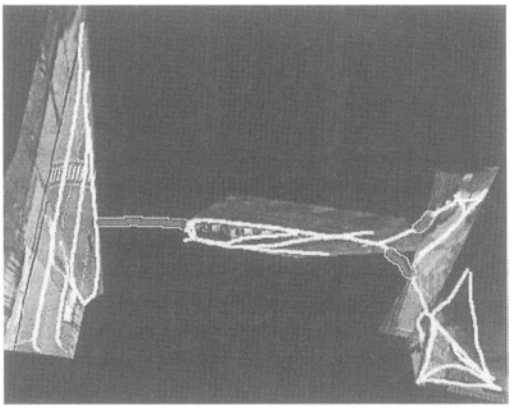


Figure 7.15. Invisible links (gray lines) detected using the MCAN and visible routes (white lines) detected using the route learning algorithm, as shown in Figure 7.12.

can be labelled in each camera view, for example entry/exit zones, paths, routes and junctions. These models can also be projected on the ground plane as is illustrated in Figure 7.15. These paths were constructed by using 3D object trajectories stored in the object motion layer. The gray lines represent the hidden paths between cameras. These are automatically defined by linking entry and exit regions between adjacent non-overlapping camera views. These semantic models enable high-level queries to be submitted to the database in order to detect various types of object activity. For example we can generate spatial queries to identify any objects that have followed a specific path between an

entry and exit zone in the scene model. This allows any object trajectory to be compactly expressed in terms of a routes and paths stored in the semantic description layer.

Table 7.1. Attributes stored in semantic description layer (entry/exit zones).

Field Name	*Description*
Camera	The camera view of the entry or exit zone
Zoneid	The identification of the entry or exit zone
Position	The 2D centroid of the entry or exit zone
Cov	The covariance of the entry or exit zone
Poly_zone	A polygonal approximation of the entry or exit zone

Table 7.2. Attributes stored in semantic description layer (routes).

Field Name	*Description*
Camera	The camera view of the route
Routeid	The identification of the route
Nodes	The number of nodes in the route
Poly_zone	A polygonal approximation of the envelope of the route

Table 7.3. Attributes stored in semantic description layer (route nodes).

Field Name	*Description*
Camera	The camera view of the route node
Routeid	The identification of the route
Nodeid	The identification of the route node
Position	The central 2D position of route node
Position_left	The left 2D position of the route node
Position_right	The right 2D position of the route node
Stddev	σ Gaussian distribution of object trajectories observed at the route node
Poly_zone	Polygon representation of region between this route node and its successor

The main properties stored in the semantic description layer are described in Table 7.1, Table 7.2 and Table 7.3. Each entry and exit zone is approximated by a polygon that represents the ellipse of the region. Using this internal representation in the database simplifies spatial queries to determine when an object enters an entry or exit zone. The polygonal representation is also used to approximate the envelope of each route and route node, which reduces the complexity of the queries required for online route classification that will be demonstrated in the next section. An example of the routes, routenodes, entry and entry regions is shown in Figure 7.16. The black and white ellipses indicate entry and exit zones, respectively. Each route is represented by a sequence of

nodes, where the black lines represent the main axis of each route, and the white lines define the envelope of each route.

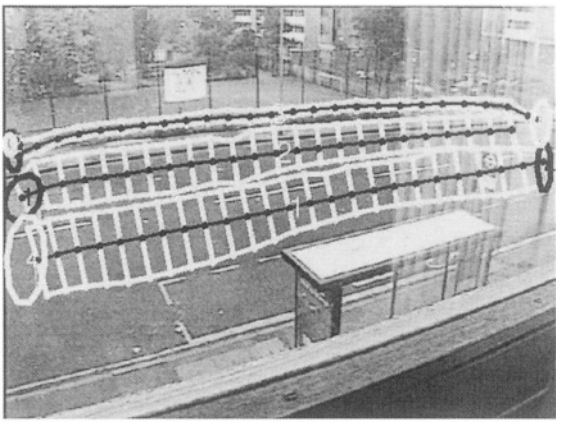


Figure 7.16. Example of routes, entry and exit zones stored in semantic description layer.

6.1 Metadata Generation

Table 7.4. Metadata generated (object_summary).

Field Name	*Description*
Videoseq	The identification of the capture video sequence in the image framelet layer
Trackid	The trackid of the object
EntryTime	The time when the object was first detected
ExitTime	The time when the object was last seen
EntryPosition	The 2D entry position of the object
ExitPosition	The 2D exit position of the object
Path	A sequence of points used to represent the object's 2D trajectory
Appearance	The average normalized colour components of the tracked object

Metadata is data that describes data. The multi-layered database allows the video content to be annotated using an abstract representation. The key benefit of the metadata is that it can be more efficiently queried for high-level activity queries when compared to the low level data. It is possible to generate metadata online when detected objects are stored in the image framelet and object motion layers. Initially, the video data and object trajectory is stored in the image framelet and object motion layers. The object motion history is then expressed in terms of the model stored in the semantic description layer to produce a high-level compact summary of the object's history. The metadata contains information for each detected object including: entry point, exit point,

Table 7.5. Metadata generated (object_history).

Field Name	*Description*
Videoseq	The identification of the captured video sequence in the image framelet layer
Trackid	The trackid of the object
Routeid	The identification of the route
EntryTime	The time the object entered the route
Entrynode	The first node the object entered along the route
EndTime	The time the object left the route
ExitNode	The last node the object entered along the route

time of activity, appearance features, and the route taken through the FOV. This information is tagged to each object detected by the system. The key properties of the generated metadata are summarised in Table 7.4 and Table 7.5. Each tracked object trajectory is represented internally in the database as a path geometric primitive, which facilitates online route classification.

Visual Queries. One application of the surveillance database is to support object activity related queries. The data stored in the surveillance database provides training data for machine learning processes that learn spatial probabilistic activity models in each camera view. By integrating this information with tracking data in the surveillance database it is possible to automatically annotate object trajectories. The image framelet layer of the database contains the low-level object pixel data that was detected as a moving object by the single view camera. This layer is used to support video playback of object activity at various time intervals. The second layer is comprised of the object tracking data that is captured by the single view tracking performed by each intelligent camera. The data consists of the tracked features of each object detected by the system. The tracked features stored as a result of single view tracking include: bounding box dimensions, object centroid, and the normalised colour components. Data is extracted from the object motion layer in order to learn spatial probabilistic models that can be used to analyse object activity in the scene. The semantic description of the scene allows the information in the object motion layer to be expressed in terms of high-level meta-data that can support various types of activity based queries. The query response times are reduced from several minutes to only a few seconds.

An example of the results returned by an activity query is shown in Figure 7.17. The semantic description of the scene includes all the major entry and exit regions identified by the learning process. The major entry and exit regions are labelled on each image. The first example in Figure 7.17a shows a sample of pedestrians moving between entry region B and exit region A. The second example in Figure 7.17b shows a sample of pedestrians moving between entry region C to exit region B. The activity based queries are run using the meta-data layer of the database, resulting in considerable savings in terms of execution time, compared to using the object motion, or image framelet layers.

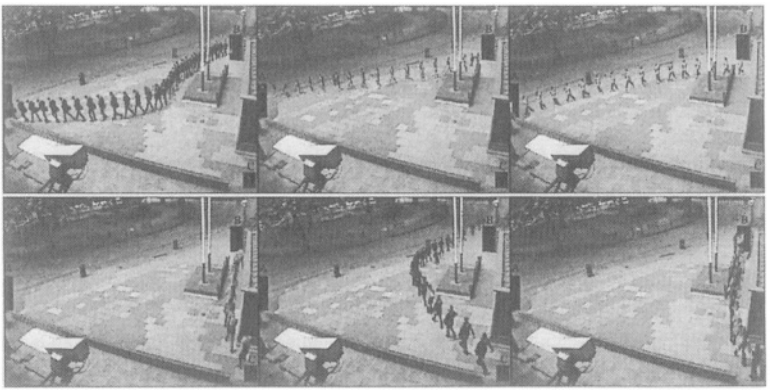


Figure 7.17. Visualisation of results returned by spatial temporal activity queries.

Figure 7.18 illustrates how the database is used to perform route classification for two of the tracked object trajectories. Four routes are shown that are stored in the semantic description layer of the database in Figure 7.18. In this instance the first object trajectory is assigned to route 4, since this is the route with the largest number of intersecting nodes. The second object trajectory is assigned to route 1. The corresponding SQL query used to classify routes is shown below. Each node along the route is modeled as a polygon primitive provided by the PostgreSQL database engine. The query counts the number of route nodes with which the object's trajectory intersects. This allows a level of discrimination between ambiguous choices for route classification. The '?#' operator in the SQL statement is a logical operator that returns true if the object trajectory intersects with the polygon region of a route node. Additional processing of the query results allows the system to generate the history of the tracked object in terms of the route models stored in the semantic description layer. A summary of this information generated for the two displayed trajectories is given in Table 7.6. It should be noted that if a tracked object traversed multiple routes during its lifetime then several entries would be created for each route visited.

```
select routeid, count(nodeid)
from routenodes r, objects o
where camera=2
and o.trajectory ?# r.polyzone
and o.videoseq =87
and o.trackid =1
```

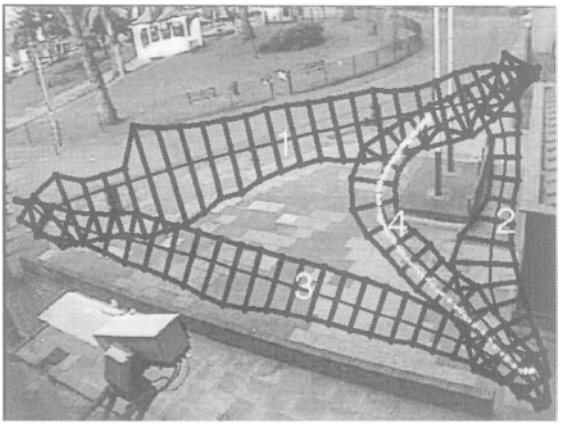


Figure 7.18. Example of online route classification.

Table 7.6. Results returned by the SQL query to the database.

Videoseq	*Trackid*	*Start Time*	*End Time*	*Route*
87	1	08:16:16	08:16:27	4
87	3	08:16:31	08:16:53	1

7. Summary

This chapter described a methodology for learning activity-based semantic scene models from observing activity, in the application area of automatic visual surveillance. More specifically, a semantic scene model was introduced, consisting of features like entry/exit zones, stop zones, routes, paths and junctions.se The semantic labels are learnt by unsupervised algorithms that exploit the vast amount of motion observations that can be gathered by surveillance systems.

Entry/exit zones and stop zones are semantic labels associated to single-point events. These regions are modelled by GMMs and learnt by an EM-based algorithm. Routes, paths and junctions are semantic labels assocaited to sequence-of-point events (trajectories). Route models can be learnt automatically from a set of trajectories. Then, the routes are segmented to paths and junctions using computational geometry.

The activity of a scene can analysed using a Route-Based Hidden Markov Model (RBHMM) that is a HMM superimposed on the scene routes. RBHMM are proposed to be time-variant so they can encode the variability of the activity with respect to the time of the day.

Scene models from multiple camera views are intergrated using the Multiple-Camera Activity Network (MCAN). This model allows tempo-probabilistic linking of camera views and does not require any manual camera calibration. Instead, the network is learnt though an automatic correlation-based algorithm.

The methodology that was described in this aimed to provide surveillance systems with a high-level knowledge of their environments. Also, it was inspired by the idea of autonomous surveillance systems that can be self-calibrated and can adapt to changes of their environments.

The database stores several different representations of the tracking data, which supports spatial-temporal queries at the highest level, to the playback of video data at the lowest level. In the earlier chapter on object tracking the image framelet and object motion layers of the database were discussed. In this chapter the semantic description layer of the database was described along with its applications for performing online route classification and metadata generation. The advantage of using a hierarchical database is that the metadata can be utilized to give much faster response times to various object activity queries than would be possible when querying the original tracking data.

Acknowledgments

This work was partly undertaken with support from the Engineering and Physical Science Research Council (EPSRC) under grant number GR/M58030.

References

[1] James Black, Tim Ellis, “Multi Camera Image Tracking”, Second International Workshop on Performance Evaluation of Tracking and Surveillance, PETS2001, Kauai, Hawaii, December 2001.

[2] James Black, Tim Ellis, Dimitrios Makris, "A Hierarchical Database for Visual Surveillance Applications", IEEE International Conference on Multimedia and Expo, ICME2004, Taipei, Taiwan, June 2004.

[3] Dempster, N. Laird, D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm", Journal of the Royal Statistical Society, B-39, pp.1-38, 1977.

[4] Tim Ellis, Dimitrios Makris, James Black, "Learning a Multicamera Topology", Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, VS-PETS2003, pp. 165-171, Nice, France, October 2003.

[5] Dimitrios Makris, Tim Ellis, "Finding Paths in Video Sequences", British Machine Vision Conference, BMVC2001, pp.263-272, Manchester, UK, September 2001.

[6] Dimitrios Makris, Tim Ellis. "Spatial and Probabilistic Modelling of Pedestrian Behaviour", British Machine Vision Conference, BMVC2002, pp.557-566, Cardiff, UK, September 2002.

[7] Dimitrios Makris, Tim Ellis, "Path Detection in Video Surveillance", Image and Vision Computing, vol.20(12), pp.895-903, October 2002.

[8] Dimitrios Makris, Tim Ellis, "Automatic Learning of an Activity-Based Semantic Scene Model", IEEE Int. Conf. on Advanced Video and Signal Based Surveillance, AVSS2003, pp. 183-188, Miami, FL, July 2003.

[9] Dimitrios Makris, Tim Ellis, James Black, "Learning Scene Semantics", Early Cognitive Vision Workshop", ECOVISION 2004, Isle of Skye, May 2004.

[10] Dimitrios Makris, Tim Ellis, James Black, "Bridging the Gaps between Cameras", IEEE Conference on Computer Vision and Pattern Recognition, CVPR2004, Washington DC, USA, June 2004.

[11] Dimitrios Makris, "Learning an Activity-Based Semantic Scene Model", PhD Thesis, City University, London, 2004.

[12] Athanasios Papoulis, "Probability, Random Variables and Stochastic Processes", Third Edition, McGraw-Hill, 1991.

[13] Lawrence R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", Proc. of the IEEE, Vol. 77, no. 2, pp. 257-286, February. 1989.

[14] Stauffer, K. Tieu, "Automated multi-camera planar tracking correspondence modelling". IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2003, vol.1, pp 259-266, Madison Wisconsin, June 2003.

[15] G.P. Stein, "Tracking from Multiple View Points: Self-calibration of Space and Time", Image Understanding Workshop, Montery, CA, November 1998.

[16] T. N. Tan, G. D. Sullivan, K.D. Baker, "Recognising Objects on the Ground-plane", Image and Vision Computing, vol.12, pp. 164-172, 1994.

Chapter 8

FAST ONLINE SPEAKER ADAPTATION FOR SMART ROOM APPLICATIONS

S.Kadambe
HRL Laboratories, LLC, USA
skadambe@hrl.com

Keywords: Smart room, remote monitoring, online speaker adaptation, stochastic matching, maximum likelihood

1. Introduction

In smart room applications such as remote monitoring of (a) meetings, (b) conference presentations and (c) progress of a trainee an automatic speech recognition (ASR) engine is used to extract excerpts from the speech of different people attending the meeting or conference or the training and to convert it to text so that it can be transmitted to the remote location. The performance of an ASR engine has to be robust to noise, to different room acoutics and to different speakers for it to meet the user's satisfaction. In this chapter, we are concentrating on making an ASR engine robust to different speakers which includes non-native speakers i.e., the speakers whose first language is different from the language for which an ASR engine is trained for.

In order to make an ASR engine robust for non-native speakers, the acoustic models it uses have to be adapted for that particular speaker. Generally, a separate adaptation training speech data is collected from different speakers and the acoustic models are adapted for that particular speaker using the training data associated with that person. However, this is not a practical solution since in a meeting or in a conference presentation scenarios, speakers can be dynamically changing as people will be coming in and out of a meeting or a presentation. This implies that anybody's speech excerpts could be extracted at any given point of time. This necessitates a need for on-line adaptation without using any adaptation training data - physically it is not possible to collect any training data in these scenarios. Some of the techniques developed hether to [1]- [7] can be modified for on-line adaptation with some degradation in performance. However, a practical solution is to adapt the models on-line without the requirement of adaptation training data with no or minimal degradation in

performance. In this chapter, such a technique that uses modified maximum likelihood stochastic matching is described.

The adaptation conceptually corresponds to projecting the trained models to the testing environment. This kind of projection can be performed at signal, feature and model spaces as shown in Figure 8.1. Most of the adaptation

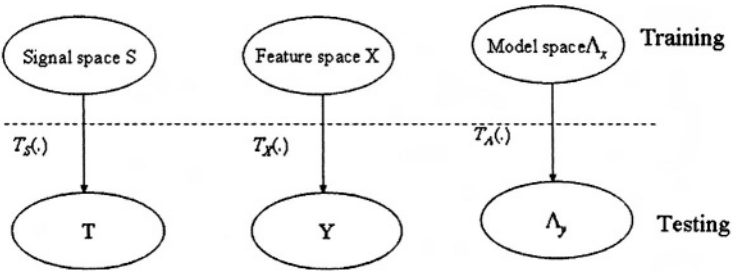


Figure 8.1. The conceptual diagram of adaptation

techniques developed so far perform the projection in the model space and are based on linear or piece-wise linear transformation. These techniques are computationally expensive, need separate adaptation training data and are not applicable for the derivatives of cepstral coefficients. The technique described in this chapter also applies the projection in the model space but is applicable for the derivatives of cepstral coefficients and does not require separate adaptation training data as most of the other adaptation techniques do and hence, is a very practical approach. To achieve near real time speech recognition, the adaptation process should not take much time. Therefore, for quick adaptation, the described technique adapts the acoustic models utterance by utterance and, chooses the models and the mixture components of the chosen model that need to be projected to match the testing environment. The main idea behind adapting the models utterance by utterance is to have the flexibility and ability to adapt to new speakers as they change dynamically. The organization of the chapter is as follows: section 2 describes the proposed adaptation technique; section 3 gives the implementation details with respect to a continuous speech recognizer; section 4 provides the experimental details and results and section 5 concludes and discusses the future research directions.

2. Description of the proposed on-line adaptation technique

If the bias (noise, speaker, etc.) in signal space is convolutive, the bias in feature space using Cepstral features will be additive: $y_t = x_t + b_t$ where b_t corresponds to bias or distortion. Let the distortion model λ_B be a single Gaussian density with a diagonal covariance matrix and is of the form: $p(b_t) = N(b_t : \mu_b, \sigma_b^2)$. Under these assumptions the structure of the parameter set λ_y

of the training model space remains the same as that of the training set λ_x. This implies that the means and variances of λ_y are derived by adding a bias:

$$\mu_y = \mu_x + \mu_b \tag{8.1}$$

$$\sigma_y^2 = \sigma_x^2 + \sigma_b^2 \tag{8.2}$$

where $(\mu_{\mathbf{x}}, {\sigma_{\mathbf{x}}}^2)$ corresponds to parameters of the training model λ_x and $(\mu_{\mathbf{y}}, {\sigma_{\mathbf{y}}}^2)$ to λ_y. These equations define the model transformation $G_\eta(.)$ with the parameters to estimate $\eta = (\mu_{\mathbf{b}}, \sigma_{\mathbf{b}}^{\mathbf{2}})$. The bias parameters estimation problem can be written as:

$$\begin{aligned} \eta' &= (\mu'_{\mathbf{b}}, \sigma_{\mathbf{b}}^{\mathbf{2}'}) \\ &= \underset{\mu_{\mathbf{b}}, \sigma_{\mathbf{b}}^2}{\operatorname{argmax}} \sum_S \sum_C p(\mathbf{Y}, S, C|\eta, \Lambda_X) \end{aligned} \tag{8.3}$$

Here, S is a set of all possible selected models for a given set of trained models Λ_X and C is a set of all selected mixture components of chosen models. The auxiliary function in a maximum likelihood sense (ignoring the prior word probabilities of the language model) that is used in solving the above optimization problem iteratively with the Expectation Maximization (EM) algorithm is:

$$\begin{aligned} Q(\eta'|\eta) &= -\sum_{t=1}^{T}\sum_{n=1}^{N}\sum_{m=1}^{M}\gamma_t(n,m)\sum_{i=1}^{D}\xi(m,n,i) \\ &\textit{where} \\ \xi(m,n,i) &= [\frac{1}{2}\log(\sigma_{n,m,i}^2 + \sigma_{b_i}^{2'} + \frac{(y_{t,i} - \mu'_{b_i} - \mu_{n,m,i})^2}{2(\sigma_{n,m,i}^2 + \sigma_{b_i}^{2'})})] \end{aligned} \tag{8.4}$$

Here T corresponds to total observation time, N corresponds to number of chosen acoustic models for adaptation (in our case this corresponds to a boundary model since the ASR engine that we use is based on segments and which are modeled as boundaries), M is the chosen number of mixture components for each chosen model and D is the total number of bias parameters that is estimated for each model. Taking above equation's derivative with respect to η' does not result in a closed form solution for $\sigma_{b_i}^{2'}$; however, in [1] the authors show an alternate approach. In this paper, we use that approach to derive the estimates of the bias parameters. Equations corresponding to estimated bias parameters are provided below.

$$\mu'_{b_i}(n) = \frac{\sum_{t=1}^{T}\sum_{m=1}^{M}\gamma_t(n,m)E(b_{t,i}|y_{t,i}, model = n, c_t = m, \eta, \Lambda_x)}{\sum_{t=1}^{T}\sum_{m=1}^{M}\gamma_t(n,m)} \tag{8.5}$$

$$\sigma_{b_i}^{2'}(n) = \frac{\sum_{t=1}^{T}\sum_{m=1}^{M}\gamma_t(n,m)E(b_{t,i}^2|y_{t,i}, model=n, c_t=m, \eta, \Lambda_x)}{\sum_{t=1}^{T}\sum_{m=1}^{M}\gamma_t(n,m)} - \mu_{b_i}^{\prime 2} \tag{8.6}$$

Because our proposed approach uses a bias for the mean and variance parameters for each boundary model that is modeled by a GMM, the estimated bias parameters equation differs from the one derived in [1]. From the above two equations, it can be seen that We have one set of bias parameters for each boundary model n. Readers familiar with HMM's may think of these GMM's or *boundary models* as *states,* though in our case the boundary models (GMM's) are concatenated to segments, which represent the states in the FST.

The derivation of the conditional expectations E that are needed in the estimation of bias parameters in the above two equations are derived by modifying the algorithm that is described in [2] for the additive noise model. Using this algorithm together with our Cepstral data, leads to a convolutive noise model, which is appropriate for the purposes of channel equilazation and speaker adaptation. The conditional expectations are defined as

$$E(b_{t,i}|y_{t,i}, model = n, c_t = m, \eta, \Lambda_X) = \mu_{b_i} + \frac{\sigma^2_{n,m,i}}{\sigma^2_{n,m,i} + \sigma^2_{b_i}}(y_{t,i} - \mu_{n,m,i} - \mu_{b_i}) \tag{8.7}$$

$$E(b^2_{t,i}|y_{t,i}, model = n, c_t = m, \eta, \Lambda_X) = \frac{\sigma^2_{b_i}\sigma^2_{n,m,i}}{\sigma^2_{n,m,i}+\sigma^2_{b_i}} + \{E(b_{t,i}|y_{t,i}, model = n, c_t = m, \eta, \Lambda_X)\}^2 \tag{8.8}$$

Every chosen boundary model and its associated chosen mixture components have its own mean and variance bias which are added according to equations 8.1 & 8.2. To account for the reliability or accuracy of the estimated bias parameters, the update equations 8.1 & 8.2 are modified as:

$$\mu_y = \mu_x + \alpha_\mu \mu_b \tag{8.9}$$

$$\sigma^2_y = \sigma^2_x + \alpha_\sigma \sigma^2_b, \tag{8.10}$$

where α_μ and α_σ can be interpreted either as a stepsize or a weight indicating the reliability or accuracy of the bias. A small stepsize means a slower convergence to the optimum solution and/or could indicate a low reliability of the biases μ_b or σ^2_b. The stepsize could be implemented in an adaptive sense. A new idea of reliability measurement has been discussed and implemented as described below.

For each boundary model, the number of mixture components accounted for in equations 8.5 & 8.6 are counted. The more data i.e., the more mixture components has been collected for a boundary, the higher the reliability of its bias. This leads to a model dependent $\alpha(n)$ which is kept same for both mean and variance update in this study; however, we plan to use different functions for mean and variance in our future study. This $\alpha(n)$ is given by:

$$\alpha(n) = \frac{\text{number of mix. comps. used in bias formula for model n}}{\text{sum of all mix. comps. used for all models}} \tag{8.11}$$

Note that the equations 8.5 & 8.6 are in the form of weighted sum where the weight $\gamma_t(n,m)$ is the joint likelihood (or probability) of producing a symbol $\mathbf{Y}$ at time t, in model n, using the mixture m, given the model (trained) parameters Λ. That is:

$$\gamma_t(n,m) = p(\mathbf{Y}, model = n, c_t = m|\eta, \Lambda_X). \tag{8.12}$$

The calculation of this depends on the framework in which an ASR engine is implemented. The ASR engine that we consider in our study uses the Finite State Transducer (FST) framework. A description of how $\gamma_t(n,m)$ can be implemented in this frame work is provided in the next section.

Note that the technique described here is a modified version of the technique described in [1]. The modifications are: (a) in the bias parameters estimation (eqs. 8.5 & 8.6) – since a chosen set of acoustic models are adapted, these parameters are estimated for each of the chosen models and for a chosen set of mixture components, (b) in the computation of the conditional expectation (eqs. 8.7 & 8.8) and (c) in the update equations (eqs. 8.9 & 8.10) – we use the weighted update equations and the weight determines the accuracy or reliability of the estimated bias parameters.

The overall block diagram of the proposed technique is as shown in Figure 8.2. In short, the proposed on-line, fast speaker/environment adaptation corre-

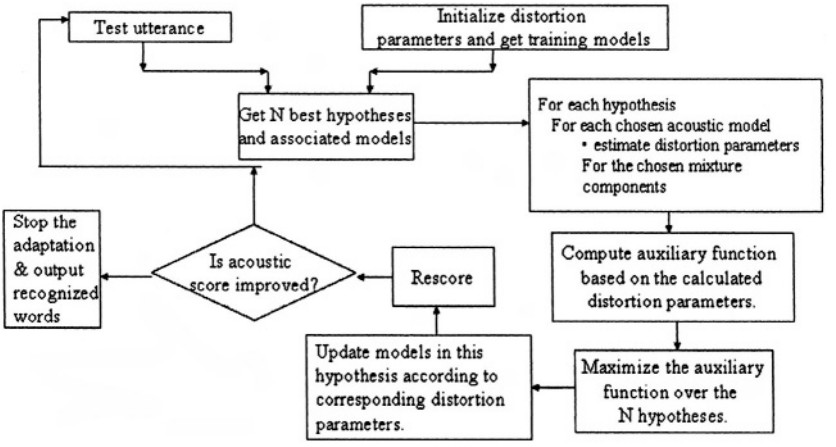


Figure 8.2. The block diagram of the proposed adaptation technique

sponds to estimating η' such that the acoustic likelihood score increases. The iterative procedure is stopped when the acoustic score of the current iteration is same as the previous iteration. The update equations (8.9 & 8.10) or the transformation $G_\eta(.)$ is applied only to the chosen models and their associated chosen mixture components. The models are chosen based on low confidence by using confidence measures where as the mixture components are chosen based on a threshold value.

The reason for choosing acoustic models with low confidence is as follows. The acoustic models with low confidence imply that the trained acoustic models did not match well with the test data. This further indicates that they are the outliers which need to be adapted so that they match better with the test data. Therefore, it makes sense to choose those models with low confidence to adapt.

The threshold value is kept constant for all different speakers. The threshold value is selected empirically which is described in section 4. In that section, the chosen value is also provided.

Note that this type of selection of acoustic models and the mixture components is not done in [1].

For the evaluation and verification of the proposed technique, it is implemented as an additional module in MIT's Summit speech recognizer and was tested using a set of non-native speakers' speech data. Note that even though this has been implemented as part of MIT's speech recognition engine, it can be implemented as part of any other speech recognition engine. Some of the details of (a) MIT's speech recognition engine and (b) the implementation of our proposed approach are provided in the following section.

3. Implementation details of proposed approach

In brief, SUMMIT is part of a spoken dialogue based system that is developed by MIT's Spoken Language Processing (SLS) group [8]. This system from lowest to highest level of processing consists of the SUMMIT, SAPPHIRE and GALAXY systems. The proposed speaker adaptation module has been implemented by making changes to SUMMIT and SAPPHIRE code.

The MIT SUMMIT System is a segment-based probabilistic speech recognition system. Using the Mel-frequency representation of a given signal, acoustic landmarks of varying robustness are located and embedded in an acoustic network. The sequence of landmarks/boundaries represents time. The segments in the network are then mapped to phoneme hypotheses. The result is a phonetic network, in which each arc is characterized by a vector of probabilities for all the possible candidates. Each segment can have one or more "landmarks" or 'boundaries' each of which is represented by a boundary model (GMM). During recognition, the phonetic network is matched to a pre-compiled pronunciation network to determine the best word hypothesis. An N-gram language model is used with the Viterbi search algorithm. SUMMIT uses an FST framework where each speech segment is represented by a state. The Viterbi training is used, resulting in a hypothesis based N-best scoring for recognition.

As mentioned in the previous section the implementation of $\gamma_t(n, m)$ (equation 8.12) depends on the ASR engine. For SUMMIT it can be implemented as follows.

3.1 Calculation of γ in an FST framework

The FST framework allows representation of information sources and data structures used in recognition which includes context-dependent units, pronunciation dictionaries, language models and lattices within a uniform structure.

In particular, a single composition algorithm is used to combine both information sources such as language models and dictionaries in advance and, acoustic observations and information sources dynamically during recognition. During speech preprocessing each utterance is split in to a number of landmarks or boundaries which delineate main changes in speech features. A boundary vector is composed of several adjacent frame feature vectors and is modeled by a boundary model – GMM. In SUMMIT currently about 1500 different boundary models are used; 92 of which are non-transitional. During recognition, SUMMIT groups the boundaries to segments/phones by making use of language model and pronunciation dictionary restrictions which have been set in advance. Up to 200 hypotheses are calculated using an FST model where each state corresponds to a segment/phone. The output consists of N best hypotheses. Each hypothesis can have a different number of segments, different segments or phones, segments differing in lengths, etc. In the current version of SUMMIT, transition probabilities are not used. All phone to phone and boundary to boundary transitions are assumed equally likely and have been assigned a log-based score of zero.

To compute $\gamma_t(n,m)$, the probability of the mth mixture component of model n producing the observation $\mathbf{o}$ at time t, given the past and the future path through the models, a probability $P(\mathbf{o}_t|model = n, mixt = m, \lambda)$ can be defined as

$$P_t(n,m) = \frac{w_{n,m}\mathcal{N}(\mathbf{o}_t; \mu_{n,m}, C_{n,m})}{\sum_{j=1}^{M} w_{n,j}\mathcal{N}(\mathbf{o}_t; \mu_{n,j}, C_{n,j})}, \tag{8.13}$$

where $w_{n,m}$ is the weight of the mixture component m of model n. With this probability and the fact that all boundary transitions are equally likely, we can write:

$$\gamma_t(n,m) = \underbrace{\prod_{ti=1}^{t-1}\left(\sum_{k=1}^{M_{s(ti)}} P_{ti}(s(ti),k)\right)}_{\alpha} \cdot P_t(n,m) \cdot \underbrace{\prod_{ti=t+1}^{T}\left(\sum_{k=1}^{M_{s(ti)}} P_{ti}(s(ti),k)\right)}_{\beta}, \tag{8.14}$$

where $M_{s(ti)}$ is the number of mixture components of the hypothesized model at time ti and T is the total number of boundaries.

The first product term in the above equation corresponds to forward probability α and the second product term corresponds to backward probability β. These two probabilities are computed as follows: At each boundary there is a model for each of the N-best hypotheses. The same model may be selected by more than one hypothesis. For the N-best hypotheses and series of landmarks there is a set of unique models for the utterance. In the adaptation only this set of models are considered. At each time boundary or landmark, forward and backward probabilities are obtained by calculating the probability to be in a model m at time landmark t by summing the mixture log probability score along each hypothesis to the boundary t and the associated partial score with the model m for that hypothesis. The log probabilities were then re-scaled

(to prevent machine overflow) and then converted to a relative probability by exponentiation. These numbers were then scaled to add to 1.0 when summed over the N-best hypotheses (i.e. converted to probabilities.). Finally if a model occurred for more than one hypothesis, the results were combined so that it occurred only once at t.

4. Experimental details and results

After implementing the proposed fast on-line adaptation technique as part of SUMMIT as an additional module called "Adaptation", it was tested using non-native speakers' data.

Basically, the adaptation is performed with an inner loop adjusting the models and an outer loop to re-process the same utterance several times as shown in Figure 8.2 to make sure there is convergence. After the EM calculation, new scores are calculated by running the recognizer SUMMIT as shown in Figure 8.2 with the updated models. If the update increases the total probability score, the models are updated again. If not, the previous set of models (the best) are restored.

First, a threshold value was set empirically for the selection of number of components of a Gaussian mixture associated with a chosen acoustic model to be adapted. This threshold value corresponds to the coefficient or the weight or the probability associated with each mixture component. This weight in SUMMIT is in the form of 10^{-d}. For different values of d, an experiment of adaptation was conducted by considering N-best hypotheses of ASR output where N was set to 2, 3, 5 and 10. For each d value, word recoginition accuracy (WA) was computed for twenty seven utterances. In Table 8.1, the threshold value, the number of hypotheses that were used in the adaptation and WA are tabulated. From this table, it can be seen that for all threshold values, there is an improvement in WA as compared to no adaptation (the first row in the Table 8.1); however, the maximum improvement in WA is for the threshold value $d = 6$ and for $N = 2$. Hence, these value were selected and was kept the same for all the speakers for whose speech utterances the proposed algorithm was tested. Next, the adaptation experiment was conducted by considering ten non-native

Table 8.1. Example of threshold selection

threh	nbest	WA
0	-	87.5
10^{-3}	2	88.4
10^{-4}	2	89.2
10^{-5}	2	90.1
10^{-6}	2	90.9
10^{-7}	2	89.2
10^{-4}	3	88.4
10^{-5}	3	88.8
10^{-6}	3	88.8
10^{-7}	3	88.8

speakers and twenty four utterances from each speaker. The acoustic models and the number of mixture components associated with each model to adapt were selected using the confidence scores and the threshold value, respectively, as mentioned before. The adaptation procedure as described in Figure 8.2 was applied to each utterance by discarding all the models that were adapted for the previous utterance. Such an adaptation procedure was applied since the next utterance could be from a different speaker especially, in the scenarios considered in this study. If the scenario is a dialogue between one speaker and an ASR engine, it makes sense not to restart the adaptation process for each utterance. However, we are interested in making an ASR engine robust to anybody at any time and any where. Hence, we are adapting utterance by utterance.

Since the EM iteration seems to converge in at most two trials the adaptation can be achieved in close to real-time. In Table 8.2, the recognition time for different utterances is provided with and without adaptation. From this, it can be

Table 8.2. Recognition time of an utterance with (w) and without (w/o) adaptation

Utterance length in secs	Recog. time w/o adaptation in secs	Recog. time w adaptation in secs
1.42	0.422	0.65
3.0	0.838	0.975
1.39	0.414	0.62
3.64	0.977	1.046
1.8	0.637	0.844
7.1	2.344	2.808
3.5	0.955	1.028
3.7	1.038	1.207
1.52	0.441	0. 667
2.76	0.933	1. 092

seen that the recognition time with adaptation is not significantly much higher than the recognition time without adaptation. For the selection of mixture components, a thresholding technique was used as mentioned before. This threshold value was selected empirically and then kept constant for all speakers. From the above discussion it can be seen that the selected threshold value is $d = 6$ and $N = 2$. The experimental results of Table 8.3 indicate that this empirically chosen threshold did not get affected by the speaker variability. Note that no separate adaptation training data was used. The recognition accuracies that we obtained for twenty four utterances from each of 10 non-native speakers are tabulated in Table 8.3. The total number of words in twenty four utterances varied from speaker to speaker. Hence, this is also tabulated in this table. From this table it can be seen that an average of 7.4 % improvement in word accuracy can be obtained across ten non-native speakers. Note that in systems that are used for remote monitoring, the N-best hypotheses output of an ASR engine is further processed by applying natural language processing techniques to reduce the error in extracting the excerpts from speech in the form of text. The recognition accuracy improvement of an ASR engine by 4 to 7 % will

Table 8.3. Word recognition accuracy (WRA) for 24 utterances

Speaker	WRA w/o adaptation in %	WRA w. adaptation in %	Relative Improvement in %	# of words
1	87.5	90.9	3.9	232
2	61.5	65.6	6.7	195
3	87	90	3.3	192
4	77.4	84	7.9	186
5	72.4	77.2	6.0	225
6	74.9	80.4	6.8	200
7	65.4	69.2	5.5	285
8	66.7	74.6	10.6	246
9	75.2	87.7	14.25	488
10	63.6	69. 9	9.01	855

improve the overall performance of the system by 10 to 15 % [8] which would definitely helps in improving the user's satisfaction.

5. Conclusions

The results indicate that the proposed on-line fast adaptation technique adapts the required models fast and improves the recognition accuracy significantly. The main advantages of this technique are (a) it does not need separate adaptation training data which is impractical to obtain since the ASR systems can be used by anybody, anytime and anywhere especially, in the smart room applications and (b) it performs adaptation in near real-time since it adapts only a small set of models that need to be adapted for any given utterance and the EM algorithm converges within two iterations. Future work warrants the usage of mixture Gaussian components for the distortion model to further improve the recognition accuracy.

References

[1] A. Sankar and C-H. Lee, "A maximum likelihood approach to stochastic matching for robust speech recognition," IEEE Trans. On Speech and Audio Processing, 4:190-202, May 1996.

[2] R. C. Rose, E. M. Hoftsetter and D. A. Reynolds, "Integrated models of signal and background with application to speaker identification in noise," IEEE transactions on Speech and Audio processing, 2(2):245-257, April 1994.

[3] Hui Jiang and Li Deng, "A robust compensation strategy for extraneous acoustic variations in spontaneous speech recognition," IEEE Transactions on Speech and Audio Processing, 7(1):9 -17, Jan. 2002.

[4] J. McDonough, T. Schaaf, and A. Waibel, "On maximum mutual information speaker-adapted training", ICAASP 2002, 1:601-604, 2002.

[5] Bowen Zhou and J. Hansen, "Rapid speaker adaptation using multi-stream structural maximum likelihood eigenspace mapping," ICASSP

2002, 4:4166-4169, 2002.

[6] J.-T. Chien, "Online unsupervised learning of hidden Markov models for adaptive speech recognition," IEE Proceedings on Vision, Image and Signal Processing, 148(5): 315 -324, October 2001.

[7] Shaojun Wang and Yunxin Zhao, "Online Bayesian treestructured transformation of HMMs with optimal model selection for speaker adaptation," IEEE Transactions on Speech and Audio Processing, 9(6): September 2001.

[8] V. Zue, et al, "JUPITER: A telephone-based conversational interface for weather information," IEEE trans. On speech and audio processing, 8(1): 85-96, January 2000.

Chapter 9

STEREO-BASED 3D FACE RECOGNITION SYSTEM FOR AMI

S.Ramalingam and D.Ambaye
Department of Computing, Middlesex University, UK
{s.ramalingam,d.ambaye}@mdx.ac.uk

Keywords: Ambient Intelligence, AmI, Non-Intrusive Verification and Authentication, NIVA, computer vision, face-recognition, Linear/Fisher Discriminant Analysis (LDA/FDA), Image Indexing, Stereo-Imaging.

1. Introduction

Ambient Intelligence (AmI) is the notion of technology embedded into our surroundings at work, home and leisure contexts that can be designed to make our lives more safe, more effective, less arduous and more enjoyable.

A key attribute of AmI technologies is that they should be both ubiquitous and innocuous. They are technologies embedded into our natural surroundings, present whenever needed, enabled by simple and effortless interactions, attuned to all our senses, adaptive to users and context and autonomously acting. Leading commentators suggest that the near future will see AmI applications in every day situations ranging from safe driving systems, smart buildings and home security to smart fabrics or e-textiles. It is also becoming of interest to a wide range of commercial and law enforcement applications. The key point being that they are poised to become part of every day life as we will know it.

Undoubtedly, one of the most important enablers of AmI is intelligent vision. The ability to detect, and recognise people or objects in the environment is a key pre-requisite for many AmI applications. The range of possible applications for intelligent vision is steadily expanding as advances in Ambient Intelligence (AmI) is the notion of technology embedded into our surroundings at work, home and leisure contexts that can be designed to make our lives more safe, more effective, less arduous and more enjoyable.

Although very reliable biometric methods exist, such as fingerprint analysis, retinal or iris scans, these methods are highly intrusive in respect to the overall capture to recognition cycle. For instance, such systems require users to be subjected to unnatural and repetitive identification processes. In some instances the additional process time overheads placed on the user can also be an issue. However, a personal identification system based on face images that

are frontal or partial in view can be less intrusive, more natural in environment [21] and faster. An appropriately designed face recognition technology system can enable user-friendly and fast access to an ATM machine or a computer, to control entry into restricted areas, to recognize individuals in specific areas (banks, stores).

A common issue for most face recognition systems is achieving consistent performance levels under non-standard conditions. Differing applications can bring varying levels and types of constraints which the system must cope with. It must operate under a variety of conditions, such as varying illuminations and facial expressions, it must be able to handle non-frontal facial images regardless of gender, age or race. This can have an impact on relative performances in terms of speed and accuracy. As would be expected, differing face recognition approaches currently available satisfy such constraints to varying degrees.

The Non-Intrusive Verification and Authentication (NIVA) project, of which the work described in this paper is a part, explores these and other issues in relation to face recognition in an AmI context. The project has the primary objective of establishing best practices on how to develop and implement face recognition systems that are more user-friendly and effective. It is currently investigating a range of approaches to face recognition and verification. The approach described in this paper, stereo-based 3D face recognition, is of interest because it offers particular characteristics which, taken together, could improve the usability and effectiveness of AmI applications. For instance, some potential advantages include:

- Intrinsic high levels of accuracy in recognition.
- Ease of adaptability to varying conditions in the environment (eg. pose, intensity).
- Authentication/verification in real-time with free movement to enable a more naturalistic user interaction.

Nevertheless, stereo-based face recognition systems suffer from the following disadvantages that they have not yet gained the usability in applications:

- Calibration of the stereo camera is a research problem in building robust recognition systems.
- 3D algorithms have been developed for tracking but not recognition, possibly due to the difficulty in discriminating face objects from non-face objects.
- Being 3D, any system has to cope with the handling of volumetric data and is a serious problem for large sized databases.
- Not being cost-effective as a hardware component.

We attempt to overcome these issues through the use of a commercially available stereo-camera system that is easy to set up and integrate with the vision system. We have demonstrated an easy means of handling 3D data in feature space, and used for recognition that has real-time and robust performance.

2. Face Recognition: Review

Face recognition deals with the identification or verification of one or more persons in the database. Identification verifies an unknown query image as existing in the database or not. Verification authenticates claimed identity. Face recognition systems fall into two groups, namely those that use static images and others that use video sequences. The actual technique of image processing, interpretation and recognition will actually depend on the application for which it is used. Hence a third degree of classification exists based on the application itself.

In the following sections, we review some of the recent face recognition systems based on the use of still images, 3D images as well those that employ retrieval and indexing mechanisms as part of the recognition process. We then briefly describe the proposed system that employs 3D imaging with indexing mechanisms.

2.1 Face Recognition from Still Images

A recent review paper [21] gives a thorough survey of face recognition that exist both as research and commercial systems. In most applications the images are available only in the form of single or multiple views of 2D intensity data, so that the inputs to the computer face recognition algorithms are visual only. For this reason, the literature reviewed in [21] is restricted to studies of human visual perception of faces. A summary of this detailed review is described in the following section.

Using Principal Component Analysis (PCA) many techniques have been developed:

- Eigenfaces which uses a nearest neighbour classifier.
- Feature-line-based methods, which replace the point-to-point distance with the distance between a point and the feature line linking two stored sample points.
- Fisherfaces which use linear/Fisher discriminant analysis.
- Bayesian methods, which use a probabilistic distance metric and SVM methods, which use a support vector machine as the classifier.

In face recognition applications utilising higher order statistics, independent component analysis (ICA) is argued to have more representative power than PCA. ICA is a generalisation of PCA, which de-correlates the higher order moments of the input in addition to the second order moments.

More recent methods have been concerned with both representation and recognition, so a robust system with good generalisation capability can be built. Key lessons learned in FRVT[11] were:

1 Given reasonable controlled indoor lighting, the current state of the art in face recognition is 90 per cent verification at a 1per cent false accept rate.

2 Face Recognition in outdoor images is a research problem.

3 The use of morphable models can significantly improve nonfrontal face recognition.

4 Identification performance decreases linearly in the logarithm of the size of the gallery.

5 In face recognition applications, accommodations should be made for demographic information since characteristics such as age and sex can significantly affect performance.

2.2 Face Recognition from Image Retrievals

In recent years, there has been an interesting trend in the application of relational database indexing and retrieval mechanisms being applied to image recognition. This has risen from the ever-increasing need for managing large set of images. In particular, vision applications such as face recognition and verification typically use a large database of face images as part of the system.

Automatic image retrieval systems are feasible in very limited domains [10]. Typically image retrieval mechanisms require human assisted and knowledge based assisted schemes for image understanding. We need to perform image understanding process before we perform indexing process. Current image database systems have been classified into two general groups : 1) Databases with no image understanding capabilities or 2)Vision systems with image repositories. The first type relies on alphanumeric descriptions of images stored in a database. It was pointed out in current research that these types of systems lack the ability to accurately interpret and retrieve complex images [2]. On the other hand traditional computer vision systems have addressed the latter problem, but rely on image repositories with little concern about efficient insertion, indexing and querying common in database optimization.

Indexing is vital for response critical real-time applications, be it banking or identifying a criminal during the short span of time spent in the que to the check-in desk of an air-port. With vision applications, indexing has so far been an implicit mechanism embodied within the vision system [7]. With the application of database indexing and query mechanisms, it is now possible to treat the recognition part as being separate from the retrieval part. However what makes such a system difficult is dealing with the understanding- and complexity of multi-dimensional image feature data that will be selected and used for indexing. This is a challenge for a conventional database system.

In [13] syntactic descriptions of appearances is made use of for retrieval of face images. Images are indexed based on the feature vectors, which are constructed using derivative filter outputs. The image database is a general purpose consisting of 1561 images. Given a query, the system retrieves a subset of classified objects. The system does not rank a retrieved object. The feature vectors are of fixed length. Matching is performed in two stages: in the first stage the user marks selected regions within the query image for which invariant vectors are calculated. The second stage involves spatial fitting mechanism to

derive corresponding query points. Image interpretation in this case is very computationally intensive and only possible to be performed off line. As well as the previous system, human intervention makes the system more error-prone in the query description process or an expert in the domain is required for query interpretation. The results do not inspire confidence.

In [20] face image retrieval system is presented for criminal identification. Once again the system employs human interface for image and text processing through description and visual browsing. The system relies on the presence of a domain expert during the process of interpretation. A special neural network in combination with a Kohonen's [6] MAP and Wu's LEP [19] model is used to generate a self-organized index tree. This is followed by an abstract facial image icon for retrieval. The system is tested on 100 facial images, but no results were reported.

An image retrieval mechanism involving subjectivity imprecision and uncertainty is attempted in [4]. Queries are mapped on to the database associated with statistical qualitative features, such as long, short, oval, square chin. The features once again are described in interactive manner. Elicitation of semantic attributes is a human assisted process.

Effective indexing of images [8] include R-Trees, inverted lists, weighted centre of mass, hash table indexing, two -level signature files, etc. In [9] face image retrieval technique using HMMs is presented for indexing video images. In particular it addresses the issues of illumination related to principle component analysis. The technique uses a set of local features and creates an HMM model for each local feature. To index a new image the system first needs to identify the cluster to which it belongs using the Viterbi algorithm [12]. This technique is tested on a face database of 100 depths images with various facial expressions, illuminations and occlusions. In addition 30 video sequences consisting of 25 images each of frontal faces are tested. HMM is used to find a match between the query and the database, the recognition is performed in the Eigen space. The paper reports high recognition rate based on individual features.

2.3 3D Face Recognition

True non-3D face recognition algorithms deal with 2D intensity images or 3D images derived from 2D projections in effect. Face recognition based on still images or captured frames in a video stream can be viewed as 2D image matching and recogniton; range images are not available in most commercial/law enforcement applications. Face recognition based on other sensing modalitites such as sketches and infrared images is also possible. Recently there has been a few succesful applications developed for robot navigation and face tracking with stereo imaging [5].

In [3], a system called PersonSpotter using stereo imaging is described. This system is able to capture, track, and recognize a person walking toward or passing a stereo CCD camera. It has several modules, including head a head tracker, pre-selector, landmark finder, and identifier. The head tracker determines the image regions that change due to object motion based on simple

image differences. A stereo algorithm then determines the stereo disparities of these moving pixels. The disparity values are used to compute histograms for image regions. Regions within a certain disparity interval are selected and referred to as silhouettes. Two types of detectors, skin color based and convex region based, are applied to these silhouettes. The outputs of these detectors are clustered to from regions of interest which usually correspond to heads. To track a head robustly, temporal continuity is exploited in the form of the thresholds used to initiate, track, and delete an object.

To find the face region in an image, the preselector uses a generic sparse graph consisting of 16 nodes learned from eight example face images. The landmark finder uses a dense graph consisting of 48 nodes learned from 25 example images to find landmarks such as the eyes and the nose tip. Finally, an elastic graph matching scheme is employed to identify the face. A recognition rate of about 90 per cent was achieved; the size of the database is not known.

2.4 NIVA System Overview

With the backdrop of the issues described in previous sections, this paper proposes an algorithm of indexing and retrieval that is an integral part of the recognition system and is based on multi-dimensional feature vectors . Further, the system is suitable for both 3-D and 2-D set of face images. Section 3 below describes important aspects of the proposed system. The NIVA Vision System incorporates an automatic mechanism that will enable easy indexing and retrieval from a face database. What is proposed is simplifying the features to produce a suitable index tree that narrows down search to a smaller subset of the original database. The indexed database can now be effectively used for recognition. This is a two step process that uses scores of match for retrieval and recognition. Any ambiguities of indexing are resolved through the recognition process by using appropriate distance metrics.

3. NIVA 3D Vision System

The proposed system comprises of two major modules namely, the Small Vision System (SVS) and the proposed vision system. The NIVA vision system architecture is shown in Fig. 9.1. The system is developed in Matlab6 integrated with the SVS stereo camera system. The SVS takes care of image capture, pre-processing, calibration, rectification, disparity estimation, and filtering. The database is assumed to be normalized with respect to lighting given that all images were captured under normal lighting conditions over an 8 week period, and pose variations were restricted to 2-D. Normal expressions were allowed during the capture. The NIVA VISION vision system consists of the modules classified as follows:

1 **Feature Space Representation:** The module transforms the database of face images into feature space. The query face is projected onto the feature space for further matching and indexing. The outputs from this module become the input to the next module, image indexing.

2 **Image Indexing:** This module applies the standard LDA and performs a classification. A rough set is produced with a score of match between the query and the database.

3 **Face Recognition:** The FDA is now applied on the rough set. Face is recognized with a very high accuracy.

3.1 NIVA 3D Stereo-based Face Database

The face recognition system takes as input two images from two slightly different views of the same face from the stereo camera and produces a three-dimensional image called the disparity map. The database considered here is a part of the student face database developed earlier[16]. The images were captured under normal room lighting conditions. The commercial SVS stereo camera was used to capture images and determine the disparity maps. This database was developed over a period of 2 months under normal room lighting conditions. The size of the database is 10 images of 200 student individuals from the Asian community. Each student as a subject was asked to sit before a PC on top of which was mounted the stereo camera. The subject was asked to turn from left to right going through marked dots along the wall at 10 equidistant points. This works out to approximately 18 deg bpose variation. Normal expressions were allowed. The captured image outputs appear as shown in Figure 9.2.

In the current work, a stereo imaging based 3D face database of 40 student subjects with 10 views of each is used. Stereo image pairs and disparity information are stored as 8-bit bitmaps. The image sizes vary but a typical size would be 80x80 pixels. The varying size of the images for a specific lens parameter is an advantage in discriminating one individual's features from the other.

4. Face Recognition in NIVA

Face recognition in NIVA VISION is a two step process. The first step is to obtain the candidate set of images. This step filters out the images to avoid unnecessary searches. The candidate set is processed for further matching and images are ranked according to their degree of similarity. In this paper, we link the retrieval mechanism to a vision system for face recognition. Work in progress of a database indexing and retrieval mechanism as applied to a vision system is described here. In particular, we are considering a 3D stereo-based face image retrieval system. Stereo-vision provides depth perception by merging information captured from multiple images at different viewpoints.

This work is an extension to the work on stereo face recognition using discriminant eigenvectors[15]. We first discuss the face recognition system for the sake of understanding the characteristics of the face database, followed by the image retrieval mechanism in Section 5. The face recognition system consists of the modules of performing a linear discriminant analysis on a set of sampled signatures on the 3D face image. These modules are explained below.

4.1 Fisher/Linear Discriminant Analysis

Recently, practical face recognition systems have been developed based on eigenface representations. Systems using Linear/Fischer discriminant analysis as the classifier have also been very successful. Such classifiers perform LDA training via scatter matrix analysis. For an M class classification, the within- and between-class scatter matrices S_w and S_b respectively, are computed as given by equations 9.1 and 9.2 as follows:

$$S_w = \sum_{i=1}^{M} Pr(\omega_i) C_i \tag{9.1}$$

$$S_b = \sum_{i=1}^{M} Pr(\omega_i)(m_i - m_o)(m_i - m_o)^T \tag{9.2}$$

where $Pr(\omega_i)C_i$ is the prior class probability and usually replaced by $1/M$ in practice with the assumption of equal probability. S_w and S_b show the average scatter C_i of the sample vectors x of different classes ω_i around their respective means mi:

$$C_i = E[(x - m_i)(x - m_i)^T | \omega = \omega_i \tag{9.3}$$

Similarly, S_b represents the scatter of the conditional mean vectors m_i around the overall mean vector m_o. Various measures are available for quantifying the discriminative power, a commonly used one being the ratio of the determinant of the between- and within-class scatter matrices of the projected samples:

$$V_{opt} = \arg\max \left| \frac{V^T S_b V}{V^T S_w V} \right| = [\lambda_1, \lambda_2, ..., \lambda_k] \tag{9.4}$$

Let us denote the optimal projection matrix which minimizes V_{opt} by V; then V can be obtained by solving the generalised eigenvalue problem:

$$S_b \zeta_i = \zeta_i S_w \lambda_i \tag{9.5}$$

The Fisher-face method uses a subspace projection prior to LDA to avoid the possible singularity in S_w. This is the approach followed in this paper. Let a training set of N face images represent M different subjects. The face images in the training set are two dimensional arrays of disparity values, represented as vectors of dimension n. Different instances of a person's face are defined to be in the same class and faces of different subjects to be from different classes.

For the scatter matrices defined in Equations 9.1 and 9.2, the matrix cannot be found directly from Equation 9.4 because in general the matrix S_w is singular. This stems from the fact that the rank of S_w is less than $N - M$, and in general the number of pixels in each image(n) is much larger than the number of images in the learning set N. In [1], the fisherfaces method avoids S_w being singular by projecting the image set onto a lower dimensional space so that the resulting within class scatter is non-singular. This is achieved by using

Principal Component Analysis (PCA) to reduce the dimension of feature space to $N - M$ and then applying the standard linear discriminant on the resulting separation matrix defined in Equation 9.5 to reduce the dimension to M – 1.

$$V = V_{fisher} V_{pca} \tag{9.6}$$

$$V_{pca} = \arg\max |TV^T CV| \tag{9.7}$$

$$V_{fisher} = \arg\max \frac{|V^T V_{pca}^T S_b V_{pca} V|}{|V^T V_{pca}^T S_w V_{pca} V|} \tag{9.8}$$

Equation 9.6 forms the feature vector for cluster analysis. Hence every sample in the set of N face images is projected onto this feature vector corresponding to the columns of V_{fisher} and a set of features is extracted for each sample image in the training set. Alternatively, average of feature vectors may be determined for each class. This provides a generalised feature vector for each class and minimises the number of searches during matching.

4.2 Face Classification in NIVA

Using Euclidean distance in the feature space performs the recognition task. In[1], a weighted mean absolute/square distance with weights obtained based on the reliability of the decision axis was used. We stick to the Euclidean distance measure:

$$\mathcal{D}(\mathcal{T}, \mathcal{E}) = \sum_{v=1}^{k} \frac{(\mathcal{T}_v - \mathcal{E}_v)^2}{\sum_{\mathcal{E} \in S} (\mathcal{T}_v - \mathcal{E}_v)^2} \tag{9.9}$$

where $\mathcal{T}$ and $\mathcal{E}$ are the projections of the test image and example image respectively on vector v . S is the set of image instances. Therefore, for a given face image $\mathcal{T}$, the best match is given by

$$\mathcal{E}' = \arg\min_{\mathcal{E} \in S} \mathcal{D}(\mathcal{T}, \mathcal{E}) \tag{9.10}$$

4.3 Pattern Vectors

The LDA algorithm described above is applied on the set of signatures derived from the disparity images. These signatures are a result of sampling the disparity depth image along the y-axis. This procedure is similar in approach to wavelet decompositions along possible directions[15] and can be repeated for other directions here. The result of such sampling is a set of 25 signatures across the depth image (Figs.2b and 3). Fig.3 gives an image representation of the set of all signatures obtained for a face with varying intensity levels. These depth signatures could be obtained at fiducial points on the face deriving local features and then we can apply eigenspace analysis. Here, we consider equidistant sampling points along the height of the face and use any features obtained for the eignspace analysis.

For each such signature, a set of higher order central moments[14] are obtained as features (Figs. 4 and 5). This results in a feature vector of dimension 40 subjects × 10 views × 25 signatures × 6 moments. These features are used not only as a discriminating feature amongst the individual faces but also used as automatic indices for dynamic partitioning of the database. Such a mechanism enables lower response times of match and/or retireival.

5. NIVA Dynamic Indexing to Database and Recognition

In an earlier paper, we define a ***static*** database indexing mechanism for face recognition [18] that takes the approach of a relational database system for indexing. In that paper, a cluster analysis is performed along each dimension of the feature space and a conditional check is made in the multi-dimensional feature space for classifying an image to a partition. Such a partition typically includes replication of images under different partitions. An index tree structure gives the possible candidate partitions that could be searched for at any time. This is an offline process.

At the end of this process, what the system learns is what are the common features between the various images in the database and those that make them different. If so, how could we use this knowledge to partition the database appropriately. Such knowledge is particularly useful for a vision system that recognizes faces just as in the human process of registering and recognizing faces. When a query is posed, index matching is first performed which prunes the database tree. At the second level, the actual database elements are matched as in a recognition system. The system is interfaced with an RDBMS (relational database management system). This system is currently under development progress.

In this paper, we describe the NIVA indexing mechanism that ***dynamically*** partitions the database. That is, the partitioning is biased by the query and is not a pre-process. Instead it takes place on the fly. Hence each time a new query is posed, the partitioning looks different. The same result as above is obtained but with an index tree that has a better hit rate and smaller size of target sets. The target set is now pruned by using the Fisher's Discriminant Analysis (FDA) and a distance metric as described in Section 4. The result is higher efficiency with real-time performance and can adapt to an incremental learning very easily. (This is an improvement the static database partitioning).

6. NIVA Implementation of Indexing and Recognition

In this section, we describe in detail the indexing mechanism that is crucial for a high hit rate. The key to success of an efficient index tree is through the use of multivariate analysis on the feature set that provides the discriminating ability.

Much work in visual object recognition deal with different views of objects which are analysed in a way that allows access to view-invariant descriptions. Generalisation from one profile viewpoint to another is poor, though generalisation from one three-quarter view to the other is very good. Fortunately, for

face recognition the differences in the 3D shapes of different face objects are not dramatic. This is especially true after the images are aligned and normalised. Using a statistical representation of the 3D heads, PCA was suggested as a tool for solving the parametric SFS problem. The inherent nature of the 3D volumetric data and the model-based approach adopted in NIVA enable sufficient overlap of information in the sample set. The usefulness of higher order moments in vision have been demonstrated in earlier papers [14]. We use second and higher-order moments as a feature set and build a view invariant description across the canonical views. The same set of features is also used to index the database.

6.1 Feature Space

Let M be the set of unique faces in the database and N be the number of samples per face. Also let Z be the number of signatures per image. In NIVA, $M = 40$ and $N = 10$ and $Z = 25$, typically. Let $I = \{I_i \mid i = 1, ..., M.N\}$ be the set of images; each image is described by 25 signatures, so $I = \{I_i \mid i = 1, ..., Z\}$ is the set of signatures of the i-th image. The signatures are characterized by the set of 6 central moments, which are real numbers. We know that the first central moment is zero. Hence $I_{ij} = \{m_{ij}^k \mid k = 2, ..., 6\}$ is the set of the central moment values for the j-th signature of the i-th image from the database. In the discussion that follows index i refers to the image number in the database, index j to the signature's number, and index k to the moment's number.

Let the central moments be represented by the set S given by:

$$S_j^k = \{m_{ij}^k \mid i = 1, ..., M.N, j = 1, ..., N\} \tag{9.11}$$

which defines the k-th moment of the j-th signature of all images.

The sets

$$S^k = \bigcup_{j=1}^{Z} S_j^k \tag{9.12}$$

describe the partitioning of the k-th moments of all signatures of all images, and, finally, the set

$$S = \bigcup_{k=2}^{7} S^k \tag{9.13}$$

contains all the information about all the moments of all signatures for all images in the database. For computational purpose, the set S could be thought of as a 3-dimensional matrix of dimension 400 × 25 × 6.

6.2 Query Processing

The query Q is a single image to be matched with the ones from the database; the information about the query is then contained in 25 signatures, each of which

is characterized by 6 central moments. Then, we have the following:

$$Q_j^k = \{m_{Qj}^k \mid \forall j \in N, \forall k \in Z\} \tag{9.14}$$

$$Q = \bigcup_{k=1}^{Z} Q_j^k \tag{9.15}$$

Equations 9.14- 9.15 take the same definitions as in Equations 9.11- 9.15 except that it is for a specific query.

Let $\overline{S}^k$ and $\overline{Q}^k$ be the means of moments and signatures respectively taken over the samples of individual face images of the database and the query respectively. These contain a generalised representation for every subject in the database. For computational purpose, the set S is a 3-dimensional matrix of dimension 40 × 25 × 6.

The next step is a projection of the central moments of the query onto the feature space. This is achieved by applying the LDA on the feature set. The LDA classifier requires as input, the training set given by Equation 9.12, the query given by Equation 9.15 and a group vector to identify the group to which each sample of the training set belongs. The classifier determines the group into which each sample is classified and by computing the distance metric given in Equations 9.9 and 9.10. The result is a classified subset of the original database, based on evidence accumulation using the distance metric:

$$F_j = \{f_{ij} = \sum_{k=2}^{2} \delta(\overline{S}^k, \overline{Q}^k) \mid \forall j \in N, \forall k \in Z\} \tag{9.16}$$

$$\delta(x, y) = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{if } x \neq y \end{cases} \tag{9.17}$$

and f_{ij} is a frequency count of the moments of the j-th signature of the query Q and the image I_j . The sets F_j are aggregated to produce set F:

$$F_j = \{f_{ij} = \sum_{j=1}^{Z} f_{ij} \mid i = 1, ..., M\}, \tag{9.18}$$

where f_i is the accumulated matches between all the moments of the query and i-th image from the database.

NB: The result of such a matching procedure is a vector with integer entries, describing only the frequency of the match.

6.3 Step 1: Image Indexing

Equation 9.18 gives a matching score for all the images in the database as a first cut. What we need to extract is a subset of candidate images close to match with the query. This is achieved by applying a simple threshold on the scores as follows, assuming equal probability of occurrence of $j \in Z$:

Let

$$C = |F|$$
$$C \subseteq I$$

Let

$$\delta = \sum_i \frac{f_i}{C} \tag{9.19}$$

be the threshold applied for selection. Then we have candidate sets for recognition satisfying the threshold condition as follows:

$$F^0 = \{f_i | f_i \geq \delta\} \tag{9.20}$$

The cardinality of this set is expected to be small and is passed on to the FDA module of the NIVA system.

6.4 Step 2: Face Recognition

We now use the set defined in Equation 9.20 as the most likely set for of matches for the query and perform the Fisher's Discriminant Analysis (FDA) for recognition. This module is as described in Section 3.1.

A key feature of the two stage process of indexing and recognition is that, we start with a generalised representation of the models and hence the indices are deduced from these representations. However, for the recognition process, since the number of possible candidates is narrowed down to a small subset we can afford to have individual representations in terms of the various views of the models. We infer from the test results that the generalised representations are not only compact but provide high discriminative ability so as to index into a narrow candidate subset. On the other hand, the view expansions enable to pin-point the specific pose(s) of a query.

7. Testing and Results

In general, vision systems are tested in a controlled environment against the following criteria of validation, generalisation and rejection. Validation tests are conducted to verify if the system recognizes seen instances of an object. Generalises checks if the system has sufficient generalisation ability to recognize unknown instances of known objects. Rejection simply tests if the system is capable of rejecting objects that do not belong to the database, which is a difficult task. In NIVA vision, three main categories of classification were performed based on representations of the database, query and recognition as shown in table below: Table 9.1a indicates that the database at the time of indexing always maintains the compact representation. The query may take either the compact or individual view representations. Note that the compact representations in fact are the generalised representations derived from the canonical views. Likewise the database representation for recognition through FDA may take either of the representations. Whatever is the condition, the recognition performance is tested to be the same.

Table 9.1a. Feature Representations Used for Testing.

Test	Query	Index	Recognition
1	Generalised	Generalised	Generalised
2	Generalised	Generalised	View Based
2	View based	Generalised	View Based

Table 9.1b. Smaple-set sizes for the databases and Query.

Test	NIVA Module	Sample-sets	Representation
Indexing	Query	8,10	Generalised, View Based
Indexing	Database	8,10	Generalised
Recognition	Query	View Based	View Based
Recognition	Database	8,10	Generalised, View Based

Table 9.1b indicates the possible combinations used in NIVA's testing module. Typically, 10 samples/face are used in the training set both for indexing and for recognition. Other combinations, with fewer samples are also tested. Typically 8 samples in any order are chosen. In this case, the queries specifically include the left out samples. This tests the ability of the system to generalise with fewer samples. Every view of the face is used in testing.

Again, whatever might be the sample size, the representation could be one of two combinations, namely, generalisation (compact) or view-dependent (each view is treated as a sample).

7.1 Indexing and Recognition Performance

NIVA's performance is measured at two levels, namely indexing and recognition. The recognition performance is indeed influenced by that of the indexing module. In this respect, certain performance combinations take place. For instance, hits refer to direct recognition of the query as a result of indexing. There is no doubt on the match. Hence there is no need to proceed with the recognition process. Misses refer to the state when the indexing mechanism completely misses out the right match. No further recognition is carried out. Very few cases such as these occur. Other cases are enumerated below.

Resolving Cases.

1 The hit list has two top ranks: This is resolved automatically through FDA.

2 Query is ranked 1 in the indexed list, but other faces also have same ranking: resolved automatically through FDA. Frequency of occurrence is high.

3 Not in the top of the list. Along with one or more indices of same rank: resolved automatically through FDA. Frequency of occurrence is high.

4 Misses in the index list. No automatic resolution as yet. However, if threshold level is brought to lower that than what is fixed, it might be possible.

5 In the indexed hit list, but missed during recognition: Never occurs.

Ranked indexing. Table 9.2 shows ranks along the first column. These mean that the query exists in the indexed list with the rank specified. This is then followed by the recognition process.

Tables 9.2- 9.4 give part of the results carried out. It is to be noted that the recognition performance 9.3 is 100 per cent in all cases. If the indexing mechanism missed out the actual match, then recognition fails. This brings down the overall performance of the NIVA VISION system as indicated by the overall performance in Table 9.4. However, as long as the query is in the index list, whatever might be its ranking, FDA recognizes the query without fail. The performance is in real-time.

Table 9.2. Indexing Performance.

Test → Ranks ↓	Sample-sets for Database and Query 10/10	10/1-8	10/3-10	1-8/1-8
	Test 1	Test2	Test3	Test4
Hits	42.5	37.5	40.0	37.5
1	37.5	37.5	40.0	45.0
2	7.5	15.0	10.0	12.5
3	7.5	5.0	5.0	5.0
5	-	-	2.5	-
Miss	5.0	5.0	2.5	-
Total	95	95	97.5	100

Table 9.3. Recognition Performance on Indexed Subset.

Test →	Sample-sets for Database and Query 10/10	10/1-8	10/3-10	1-8/1-8
	100	100	100	100

Table 9.4. Overall (NIVA) Performance.

Test →	Sample-sets for Database and Query 10/10	10/1-8	10/3-10	1-8/1-8
	95	95	97.5	100

7.2 Conclusion and Future Work

This paper has described a two-stage process of 3D face recognition through indexing and FDA matching suited to AmI contexts. The system known as NIVA Vision has an indexing mechanism that has proved to be very efficient in narrowing down the matching sets, in computational capacity, and response time. The system has been tested on a moderate size database with 400 images in all. The simplicity of the algorithms has contributed to the performance of system. The use of higher order statistics has provided high degree of discrimination between classes. The results are very promising for pursuing further research in stereo-based face recognition.

Future work will focus on optimization and validation type activities including the following:

1 This work forms the basis for testing on an extensive 3D face database [16] that has been constructed with stereo-camera. Future work includes testing the system with video sequence that is part of the above database.

2 The suitability of extending the system to meet the needs of real-time remote authentication [17] application will be explored.

3 The indexing and recognition mechanisms proposed in [18] will be improved integrating the system to a relational database system, to make it suitable for handling very large databases.

4 Validation of this approach relative to the needs of particular AmI contexts, including e-learning and information systems transactions.

AmI contexts place high requirements in terms of performance, response and usability on vision systems. The NIVA system is a useful step towards improving both the effectiveness and usability of face recognition in AmI contexts.

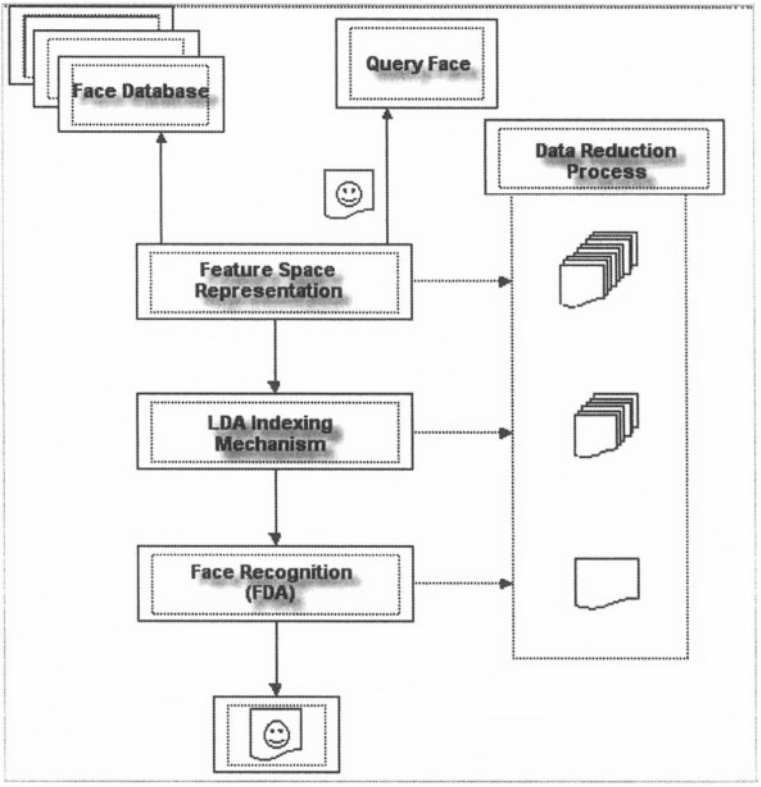


Figure 9.1. NIVA VISION Architecture

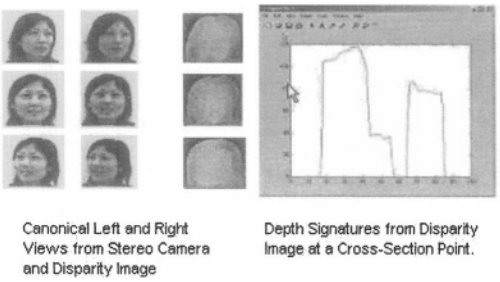


Figure 9.2. Depth Signature from Disparity Image at a Cross-Section Point

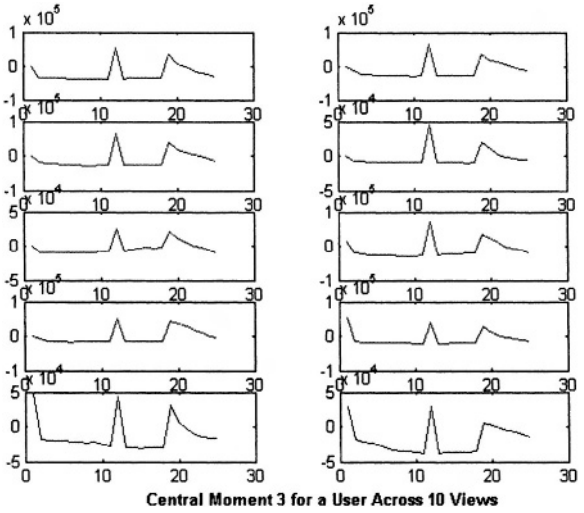


Figure 9.3. Central moments (order=2) on a set of signatures on the disparity image

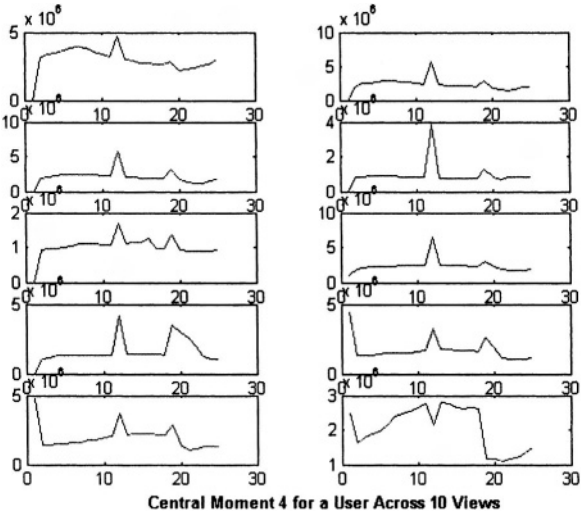


Figure 9.4. Central moments (order=3) for the same user

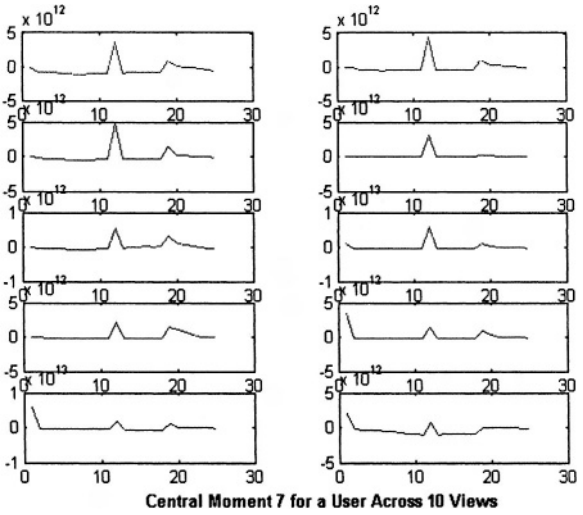


Figure 9.5. Signature Differences for 2 Users (First User)

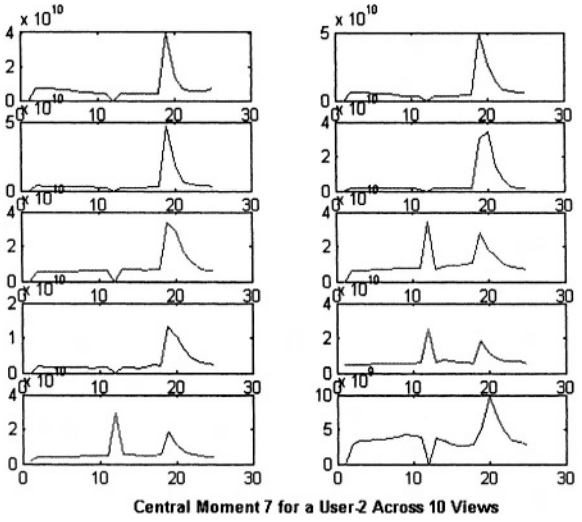


Figure 9.6. Signature differences for 2 users (second User)

References

[1] Belhumeur, P.N., Hespanha, J. P., and David J. Kriegman. (1997). " Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection," IEEE Transactions on Pattern Analysis and Machine Intelligence, v. 19 n.7, pp.711-720.

[2] Bach J., Paul S., and Jain R. (1993). "A Visual Information Management System for the Interactive Retrieval of Faces," IEEE Trans. Knowledge and Data Engineering, Vol.5, No.4, p.619-628.

[3] Douglas Decarlo , Dimitris Metaxas. (2000). "Optical Flow Constraints on Deformable Models with Applications to Face Tracking," International Journal of Computer Vision, v.38 n.2, p.99-127.

[4] Gudivada, V. V. Raghavan, and G. S. Seetharaman. (1994). "An approach to interactive retrieval in face image databases based on semantic attributes," In Third symposium on Document Analysis and Information Retrieval, p.319–335.

[5] Konolige K., 1997,"Small Vision Systems: Hardware and Implementation". Eighth International Symposium on Robotics Research. Hayama, Japan. October 1997. http://www.ai.sri.com/ konolige].

[6] Kohonen T. (1990). "The Self -Organising Map," in Proceedings of IEEE:78, Number 9 , p. 1464-1480.

[7] Kirby M., and Sirovich L. (1990). "Application of the Karhunen-Love Procedure for the Characterization of Human Faces," IEEE Trans. PAMI, No.12, Vol. l, p.103-108.

[8] Lee, S.Y., Shan, M.K., and Yang, W. P. (1989). "Similarity Retrieval of Iconic Image Database". Pattern Recognition. Vol.22. No.6. Nov. 1989. pp.675-682.

[9] Martinez A. (1999). "Face Image Retrieval Using HMMs," in Proceedings of IEEE Workshop on Content-Based Access of Images and Video-Libraries.

[10] Mulhem, P., Lim. J. (2002). "Symbolic Photograph Content-Based Retrieval," in Proceedings of the third international conference on Information and knowledge management. p.94-101.

[11] Phillips, P. J., Grother, P. J., Micheals, R. J., Blackburn, D. M., Tabassi, E., and Bone, J. M. (2003). "Face recognition vendor test 2002: Evaluation report." NISTIR 6965. Available online at http://www.frvt.org.

[12] Rabiner L.R.(1989). "A tutorial on hidden Markov models and selected applications in speech recognition," Proc. IEEE 77(1):257-285.

[13] Ravela, S. and Manmatha, R. (1997). "Image Retrieval by Appearance". SIGIR'97. In Proc. of the ACM, pp.278-285.

[14] Soodamani, R. and Liu, Z.Q. (1998)."Object Recognition by Fuzzy Modelling and Matching," Fuzzy Systems Proceedings. IEEE World Congress on Computational Intelligence, vol.1, p. 165-170.

[15] Soodamani, R. and Ronda, V. (2001). "Stereo Face Recognition using Discriminant Eigenvectors." Electrical and Computer Engineering Series. Advances in Signal Processing, Robotics and Communications. WSES Press. p. 164-169.

[16] Soodamani, R. (2001). "Inhouse Project, Stereo Imaging Based Face Recognition - Database Construction," Centre for Signal Processing, Singapore: Nanyang Technological University.

[17] Soodamani, R and Zheng, S. (2002). "Face Recognition Web-based Security," Projects showcased in CoE Technology Week 2002, Singapore: Nanyang Technological University.

[18] Soodamani, R., Vladlena, B. and David, A. (2004). "Image Retrieval and Indexing for Stereo-based 3D Face Database," Technical Report, Middlesex University, 2004.

[19] Wu J.K. (1990). "LEP-Learning based on Experiences and Perspectives," Paris: ICNN-90.

[20] Wu. J. K.,Ang Y. H., Lam, P.C., Moorthy S.K., Narasimhalu A.D.(1993). "Facial Image Retrieval , Identification, and Inference System," Proceedings of the first ACM international conference on Multimedia ACM p. 1-9.

[21] Zhao W., Chellappa R., Phillips P. J. and Rosemfeld A. (2003) "Face Recognition: A Literature Survey". ACM Computing Surveys. Vol. 35. No. 4. December 2003. pp. 399-458.

Chapter 10

SECURITY AND BUILDING INTELLIGENCE

From people detection to action analysis

G.L.Foresti[1], C.Micheloni[1], L.Snidaro[1], P.Remagnino[2]

[1] *Department of Computer Science (DIMI), University of Udine, Italy*
{Foresti,Michelon,Snidaro}@dimi.uniud.it

[2] *Digital Imaging Research Centre (DIRC), Kingston University, UK*
p.remagnino@kingston.ac.uk

Keywords: Ambient intelligence, people detection and counting, tracking, event classification and association

1. Introduction

The events occurred over the last few years have increased the importance of security. Computer vision and image processing play a paramount role in the development of surveillance systems and they are commonly used to interpret video data. Video or visual surveillance is employed in applications such as traffic monitoring [12, 19] and the automatic understanding of human activity [6]. Generally, monitoring means keeping under control a wide area by deploying a heterogeneous network of sensors including static cameras [8], ominidirectional cameras [9], motorised pan-tilt-zoom (PTZ) cameras [16] and on-board cameras (mounted on vehicles, for instance). When more cameras are used, data and information fusion algorithms must be devised; this effectively can be seen as a cooperation among sensors to solve the common tasks of monitoring a public or private area [3, 13].

Ambient Intelligence (AmI) introduces a new transparent communication layer to support machine intelligent algorithms, ultimately capable of customizing the living environment to best suit the person habits and to improve their quality of life. AmI fosters "environments able to recognize and respond to the presence of individuals in an invisible way", as stated by the IST Advisory Group (ISTAG) in the Final Report "Scenarios for Ambient Intelligence in 2010" [10]. Devising such systems is a major endeavor, that only a multidisciplinary team of researchers and technologists can make possible.

This chapter is concerned with the development of algorithms for the intelligent building. In particular, the focus is on all those technological innovations embedded in a building that enable a continuous response and adaptation to changing conditions, increasing the comfort and security of its occupants, and allowing for a more efficient use of resources. Resource management is a crucial aspect usually addressed to maximize the return of investment and to fulfill the objectives of the organization who owns it [5], [11].

In this work, we describe a security system that follows some of the criteria of AmI, to supply custom information from the monitoring of a wide area such a university building. In particular, a distributed surveillance system able to track people either on or through the floors of the building has been studied. The system is composed of a network of sensors strategically placed to detect people inside the monitored environment. Events are automatically generated once a person takesa predetermined action as entering or exiting the building, taking the stairs etc. A correlation process among the set of events allow to determine the path designed by single persons from the time instant of their ingress till the exiting from the building.

By knowing, for each time instant, the position of a person we can perform a customization of the information supplied by selecting both the event and the type of communication. In particular, we propose a framework in which an identified person can be updated with useful information anywhere inside the building. As an Example, once the event classification module has determined an important action, such as *"person A is looking for person B",* the system, by knowing where *"person B"* is, can send the information *"Person A is looking for you"* to person B by selecting the proper receiving device, i.e. laptop, desktop, PDA, etc. Finally an AmI system, currently under development, is described. It implements an intelligent infrastructure to monitor the training of student nurses in a academic environment, where hospital ward simulations take place and their monitoring is aimed at simplifying the task of the instructor and implement a novel training method.

In the following sections we will focus on the identification and localization of people inside the building by tracking and counting them.

2. System Description

The proposed system is based on a network of smart sensors that cooperating aim to understand *who is going where* inside a monitored building (i.e. the case study is represented by a university building). Here, smart sensors are represented by subsystems able to detect and to track people, to extract important features for event detection purposes and to send information for distributed data association. High level nodes fetch information generated by the sensors and associate different simple events, on the bases of rules from a knowledge base, to infer more complex and complete events. In our case, simple events are represented by states of an finite automata, see Figure 10.1, corresponding to simple actions as walking, keeping the stairs etc. Complex events, on the other hand, do not represent simply a sequence of events performed by a single person, but also semantic information. In particular from a sub-sequence, opportunely

selected, of simple events the system is able to recognize particular actions of interest for the customization of the environment.

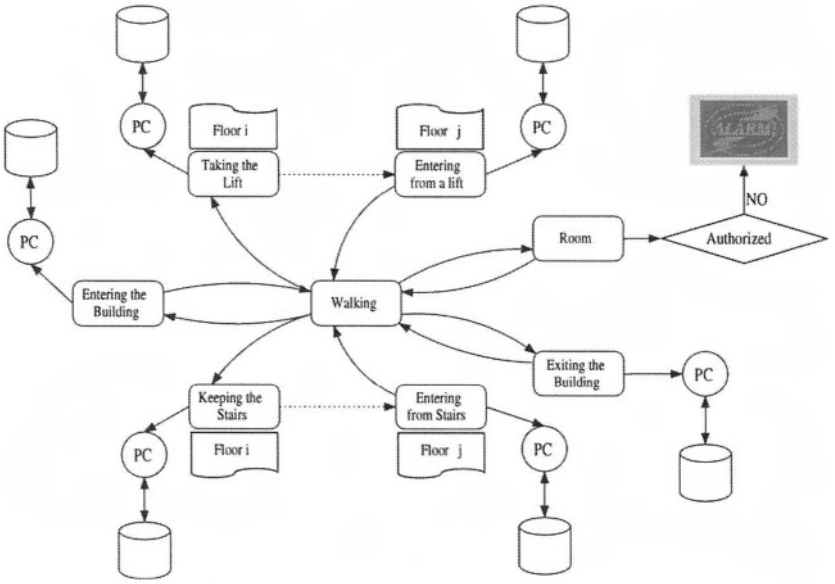


Figure 10.1. Event representation for the actions performed by a person in the building

Into the system architecture, smart sensors can be of two types, from sensors for a cooperative multisensor multitarget tracking purposes and sensors used to count people inside a building and on the floors. The former type is composed by cameras that by performing change detection techniques are able to segment moving objects inside the scene and to recognize people from other types of objects. Targets' position on a top-view map of the scene is also computed through a perspective transformation matrix obtained during the initial calibration of each sensor. Therefore, a tracker based on a Kalman Filter is applied on each sensor to compute the trajectories of each person.

The latter is represented by cameras mounted at each gate of the building and of the floors in order to acquire and to count people entering and exiting the floors. People blobs are first identified then tracked by a Kalman filter in order to have information about the direction of the persons moving inside the field of view. Hence, from information about the tracking phase the sensor is able to increment or decrement a shared variable representing the number of persons inside the floor to which the sensor belongs.

Information about movements are associate, in a centralized manner, to an ID related to the person performing the movements in consideration. Such information are used by a event detection and association module in order to

associate a state of a finite automata to the ID and finally to derive semantic information from a sequence of states associated to a single ID.

3. People tracking and counting

This module of the system aims to identify the trajectories of the people walking inside the building by maintaining also the number of persons for each floor. The problem is addressed by employing a distributed people tracker which allows to track people through an entire floor. In particular, each camera tracks people inside its field of view and exchange feature information with neighboring cameras once the target is going out of the field.

To maintain track of the number of persons on each floor we have disposed a set of people detectors installed at every entrance or exit. Such detectors are represented by systems composed by a camera placed in order to acquire a top view image which helps to segment people from the background.

3.1 People tracking

The system needs to maintain tracks of all objects simultaneously. Hence, this is a typical multi-sensor multi-target tracking problem: measurements should be correctly assigned to their associated target tracks and a target's associated measurements from different sensors should be fused to obtain better estimation of the target state.

A first tracking procedure occurs locally to each image plane. The system then executes an association algorithm to match the current detected blobs with those extracted in the previous frame. A number of techniques are available, spanning from template matching, to features matching [3], to more sophisticated approaches [4]. The approach used in our system was twofold, exploiting the Meanshift predictions [4] and matching blob features (Hu moments, base/height ratio, etc.).

To determine the target trajectories a 2D top view map of the monitored environment is taken as a common coordinates system where the correspondence between an image pixel and a planar surface is given by a planar homography [17, 7]. Measurement gating and assignment is then performed and the Mahalanobis distance can be used to determine the validation region as in [14]. This step reduces the probability of erroneous associations due to noise. The measurements coming from each sensor, for a given object, falling within the gating region are then fused together.

To deal with the multi-target data assignment problem, especially in the presence of persistent interference, there are many matching algorithms available in the literature: Nearest Neighbor (NN), Joint Probabilistic Data Association (JPDA), Multiple Hypothesis Tracking (MHT), and S-D assignment. More details on the employment of such techniques as consequence of the application can be found in [2–1, 15].

The trajectory on the top-view map of every object is modeled through a linear Kalman filter, where the state vector $\hat{x} = (x, v_x, y, v_y)$ is constituted by the position and velocity of the object on the map. At every frame (the

system processes 25 frames per second) a new measurement of the position is received. Finally, position estimates from different sensors are fused in a centralized fashion.

3.2 People counting

In this module, cameras are placed in order to have top view of the area around the gate. Hence, from the binary image $B^t(x, y)$, computed on the HSV color space, the system looks for objects in the scene. A search for connected components is then performed by discarding the ones with area below a given threshold. The great advantage of the top view is the nearly constant size of the objects, and this can be exploited for tracking and counting purposes. In fact, knowing the average area in pixels of a person from a given top view, it can be easily determined how many persons form a given connected component. The persons walking in the scene are tracked through the image to maintain a unique ID for each one [2, 15]. Despite the advantages given by the overhead placement of the camera, the tracking of the persons is non trivial due to the small field of view and possible crowded conditions. The following features are extracted for each blob and used by the tracking procedure:

- area
- density
- bounding box coordinates
- centroid coordinates
- centroid deviation from the center of the bounding box
- mean color values
- histograms of the H and S planes,

These features are used along with the Kalman filter [18] and Meanshift [4] predictions in the assignment phase. The system model used for the centroid's coordinates is the following:

$$\mathbf{x}(k) = \mathbf{A}\mathbf{x}(k-1) + \mathbf{w}_{\mathbf{x}}(k) \tag{10.1}$$

$$\mathbf{y}(k) = \mathbf{A}\mathbf{y}(k-1) + \mathbf{w}_{\mathbf{y}}(k) \tag{10.2}$$

where each of the state vectors x and y are composed by the position and velocity components and the state transition matrix is given by:

$$\mathbf{A} = \begin{pmatrix} 1 & T \\ 0 & 1 \end{pmatrix}$$

The tracking algorithm explicitly takes into account merge and split cases generated for example by two or more persons walking side by side: at some point the single blobs can be detected *(merge)* as one big blob and then split again.

4. Event detection and association

As shown in Figure 10.1 simple events are identified by states of a finite automata. First of all, when a person enters in the building, it is counted by a people counter installed to monitor the entry. Therefore, a vector $v_{p_i} = (f_1, \ldots, f_m)$ of features characterizing the $i-th$ person is extracted and associated to a unique ID which will identify the person. These features, will be used and updated in the future to allow people recognition during their motion inside the building. It is worth nothing how the vector of features can change its dimension as consequence of new features extracted or as consequence of new trusted feature in antithesis with the old ones.

The event detection and association module has the objective to assign to each active identifier (i.e. identifiers associated to a person still inside the building) a state representing an event. Having to monitor a building composed by multiple floors, the system has to face to the situation in which the target cannot be tracked continuously by sharing information among camera with overlapped views. While the system can determine the most appropriate camera to switch to when a person walks on a floor, this is not possible when a person keep the stairs or an elevator. In this case, the event association has to address an higher level of uncertainty represented by the fact that people are not tracked on the stairs or inside the elevators. In Figure 10.1, this uncertainty is represented by dashed lines. To address this problem a camera pointing to each entrance of each floor is responsible to extract important information for people recognition purposes. In particular, a vector $v_c = (f_1, \ldots, f_n)$ of the current feature is extracted for each detected person and compared with the entire set of feature vectors v_{p_i} associated to active identifier.

Let ID_i be the identifier of a person still inside the building, a distance function can be studied for the computation of the probability $P(ID_I|v_c)$ that the person ID_i is the person represented by the feature vector v_c. Therefore, the association of the detected people given the feature vector is performed by maximizing the probability. The event *person ID_i enters from the elevator/stairs on floor j* is therefore associated to the event *person ID_I takes the elevator/stairs on floor i.* The scheme of this data association is shown in Figure 10.1.

In the proposed system, event association is performed to extract semantic information from a sequence of single events. A set of inference rules belonging to a knowledge base has been adopted to derive complex actions. As an example from the sequence S={"A exits from room k","A walks on floor i",....,'A enters on floor j from stairs",'A stops in front of room l",'A does not enter room l"} the inference engine triggers the consequence "A looks for B" by knowing that the room l is associated to the person B.

5. Experimental results

The system has been tested on a real environment represented by a university building having a special room for medical purposes. In particular, the tests conducted have been performed first to test separately the performance of the people tracker and people counting modules.

In Figure 10.2 some frames of a testing sequence for people tracking are reported. In this context, the behavior of the system is resulted very good by extracting with a good reliability the trajectories of the persons moving inside the floor. The mean and standard deviation of the distance (in pixels, 1 pixel $\approx$ 10

Figure 10.2. Some frames of a sequence acquired during the test phase. Three people are entering the 3^{rd} floor from the stairs and are going to a Lab.

cm) from measured ground truth positions on the map of the walking persons in Figure 10.2 are respectively 8.66 and 4.42. On the map of the 3^{rd} floor represented in Figure 10.3 the trajectory computed for the persons of the same sequence are plotted. It is worth nothing how the trajectory represented by the continuous line at a certain point splits in two different trajectories. This is due by an error in the people detection module which is not able to distinguish two persons while they are occluded by a third one. Once the acquisition becomes optimal the system identifies both persons and their relative trajectories. The errors computed take into account also the problem of the miss detection.
Some frames of a test sequence for the people counter module are shown in Figure 10.4. The images are acquired by cameras placed at 3 meters from the floor and wide-angle (2mm focal length) lenses are used. These sequence are constituted by 7000 frames of 384x288 pixels each and lasts 280 seconds. The number of persons, equally subdivided in both directions, was 130 in total. The experiment accounted for a maximum of 4 persons walking simultaneously in the scene in different directions. The system reported a performance falling between 90% and 98%.

6. AmI for training environments

In general, all the concepts put forward by the intelligent building can be easily *ported* to professional training environments. Professional training practice

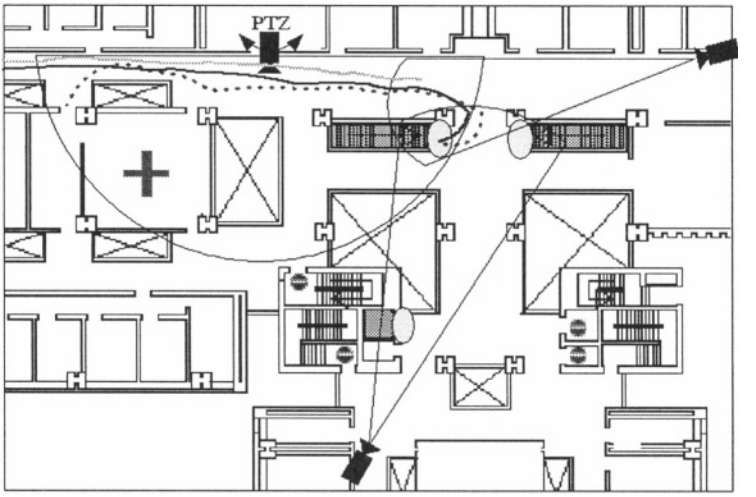


Figure 10.3. Floor map of the third floor of the University Building where the medical room is placed. Ellipses on the gates to and from stairs or elevator represent the people counter sensors that count people on each floor of the building. Rooms with "Do not enter " sign are rooms for which a granted access is needed.

max # persons	Performance 3m
1	100%
2	100%
3	95%
4	90%

Table 10.1. Performance of the system against the number of persons inside the view.

usually takes place in more than one environment where one or more instructors can directly or remotely guide or test the trainee. The training of astronauts and nurses are for instance two very different but somewhat also very similar domains for which AmI criteria can be employed. The former domain entails the training of a small number of professionals in tight spaces, the latter usually a large number of students in relatively large spaces. Both require very direct approach and a strict set of rules or protocol to follow. Professionals in both domains might have to work in many environments and under strict control of an instructor.

The Faculty of Health and Social Care Science at Kingston University runs cutting edge training methods for student nurses. Nurse training takes place in laboratories equipped as hospitals wards. The simulation commences with the delivery of a scripted handout to the students. The patient's history, diagnosis and current health status are discussed. The students are also given information on future admissions through out the day and any potential discharges. The team of both nursing and medical students is then given time to consider their

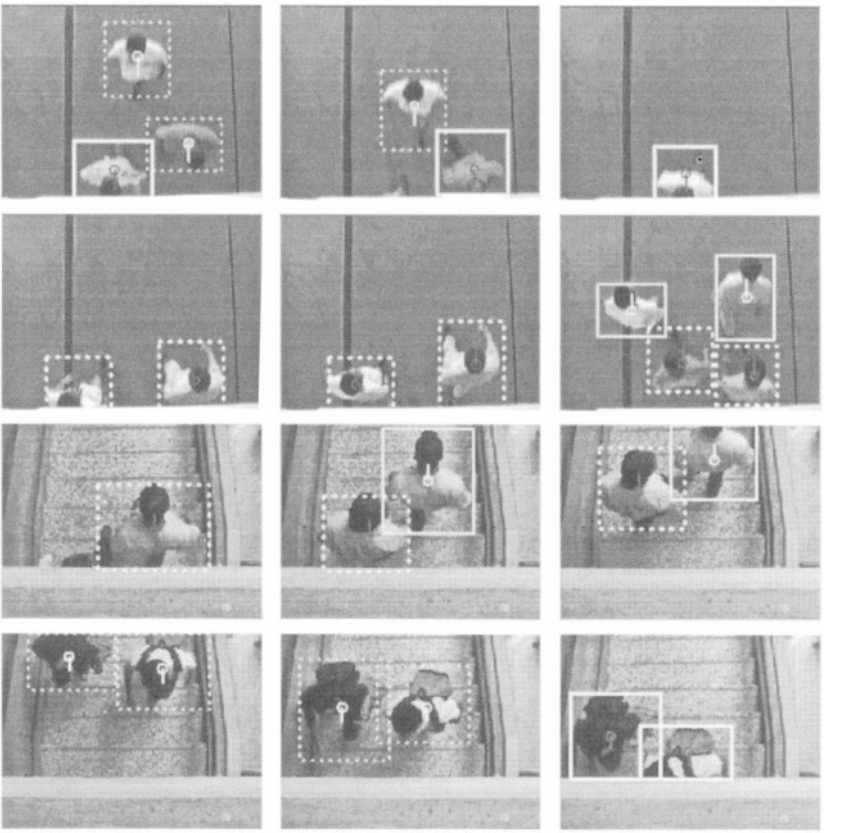


Figure 10.4. Some frames of sequences used to test the performance of the people counting module are here shown. The first two rows show images in which people respectively is entering and exiting the elevator of the third floor. The bottom two rows, instead, show images of people going respectively downstairs and upstairs. Continuous bounding box represent a person already counted by the system, while dashed bounding represent a person to be counted as entering or exiting the floor.

priorities and plan care delivery. Verbal feedback and video recording are currently used to identify student progression. However video recording is dependent on the careful placement of a limited number of stationary cameras and as such have proven problematic and limiting. Verbal feedback is conducted throughout the simulation exercises with individuals or in small sub-groups, whereas full feedback is given to the whole group at the end of an exercise. The current approach is limited because both methods of feedback have proven to be time and labour intensive and are heavily dependent on limited staff/lecturer

resources. Figure 10.5 shows two individual skills demonstrated by instructors.

Figure 10.5. Two views of one the Skills' laboratories at Kingston University.

The Ideal Objectives of the nurse training application include [1]

- To provide effective and objective feedback for individual students, during and following a simulated exercise, which identifies both best practice and highlights areas for improvement.
- Having a means to allow students to practice a task/skill repeatedly in a non-threatening environment, receiving effective feedback for the duration of practice.
- Providing students with a means of independent practice within the safe parameters of best practice.
- To equip students with effective yet time and resource efficient feedback.

AmI criteria here can be implemented in a system capable of monitoring the cluttered environment of simulation and practice sessions, to enhance the communication between instructor and trainee, including new methods to rehearse skills and replay performance. The system would make use of a sophisticated infrastructure using a network of cameras, a network of high specification computers, and sophisticated user interfaces for the real-time delivery of video contents, trainee performance evaluation and analysis.

The environment at the Faculty of Health and Social Care Sciences is illustrated in Figure 10.6. The environment is currently being equipped with a network of fixed cameras and a pan-tilt-zoom camera. The network will be used to capture visual data for individual practice skills and simulations and the data will be automatically stored in a annotated database to improve the current state of the art in professional practice training. All modules described

[1]This is outcome of a number of meetings held with Susan Rush, Kingston University Principal Lecturer leading the professional skills' laboratory.

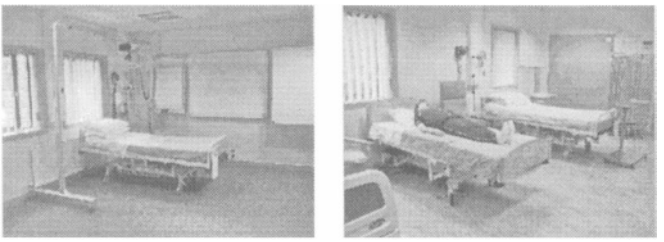


Figure 10.6. Two views of one the Skills' laboratories at Kingston University.

in the previous sections of the chapter can indeed be applied to identify trainees, track them throughout the environment and make not - for instance by storing information in a database - of their actions. In practice, in order to develop an intelligent system useful for the users - trainee and instructor - two types of handling information can be devised:

- On one hand a passive system, logging information about individual trainee over short and long period of times, during practice skills' session,
- On the other hand an active system, where information is build and on the fly communicated to the interest user, either real or quasi real-time.

The former is a standard monitoring system, very much in place in many visual surveillance installations. The latter is much more interesting and would have to implement a few of the criteria AmI advocates as necessary to create a real self-sustaining and truly interacting environment. A few problems still hold, including ethics and privacy of the trainee and, perhaps, the existence of a network of sensors, impinging on the performance of the student or professional. All these issues are currently being dealt with.

7. Conclusions

In this paper we have presented a cooperative network of smart sensors able to monitor the flow of people inside the building. The surveillance capacity of the proposed system is therefore used to customize the information supply for people inside the building. Two main aspects of current research in computer technology have been employed by using methods for environment security to customize the environment thus giving a sort of intelligence to the building. The domain of training of professionals can indeed benefit from the technologies presented in the chapter. The application domain presented here shows that there is the need for an interdisciplinary approach: technology alone cannot solve all problems, *above all* the human dimension must be taken into consideration.

Acknowledgments

This work was partially supported by the Italian Ministry of University and Scientific Research within the framework of the project "Distributed systems

for multisensor recognition with augmented perception for ambient security and customization " (2002-2004).The authors wish to thank Susan Rush, leader of the Professional Skills laboratory at Kingston University for her input on the training of student nurses.

References

[1] S.S. Balckman. *Multiple-target tracking with radar applications.* Artech House, 1986.

[2] Y. Bar-Shalom and X.R. Li. *Multitarget-multisensor tracking: principles and techniques.* YBS Publishing, 1995.

[3] R.T. Collins, A.J. Lipton, H. Fujiyoshi, and T. Kanade. A system for video surveillance and monitoring. In *Proceedings of the IEEE,* volume 89, pages 1456–1477, October 2001.

[4] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Trans. on Pattern Analysis Machine Intelligence,* 25(5):564–575, 2003.

[5] T. Derek and J. Clements-Croome. What do we mean by intelligent buildings? *Automation in Construction,* pages 395–400, 1997.

[6] S.L. Dockstader and T. Murat. Multiple camera tracking of interacting and occluded human motion. *Proceedings of the IEEE,* 89(10):1441–1455, October 2001.

[7] O.D. Faugeras, Q.T. Luong, and S.J. Maybank. Camera self-calibration: Theory and experiments. In *Proceedings of the European Conference on Computer Vision (ECCV),* pages 321–334, 1992.

[8] G.L. Foresti. Object recognition and tracking for remote video surveillance. *IEEE Trans. Circuits Syst. Video Technol.,* 9(7):1045–1062, October 1999.

[9] K. Huang and M. M. Trivedi. Video arrays for real-time tracking of person, head, and face in an intelligent room. *Machine Vision and Applications,* 14(2):103–111, June 2003.

[10] ISTAG. Scenarios for ambient intelligence in 2010. Technical Report 10, EC, February 2001.

[11] A. Kell. Intelligent buildings now. *Electrotechnology,* pages 26–27, October/November 1996.

[12] D. Koller, K. Daniilidis, and H. H. Nagel. Model-based object tracking in monocular sequences of road traffic scenes. *International Journal of Computer Vision,* 10:257–281, 1993.

[13] T. Matsuyama and N. Ukita. Real-time multitarget tracking by a cooperative distributed vision system. *Proceedings of the IEEE,* 90(7): 1136–1149, 2002.

[14] I. Mikić, S. Santini, and R. Jain. Tracking objects in 3d using multiple camera views. In *Proceedings of ACCV,* January 2000.

[15] A.B. Poore. Multi-dimensional assignement formulation of data association problems arising from multi-target and multi-sensor tracking. *Computational Optimization and Applications,* 3:27–57, 1994.

[16] B.J. Tordoff and D.W. Murray. Reactive control of zoom while tracking using perspective and affine cameras. *IEEE Trans. Pattern Anal. Machine Intell.*, 26(1):98–112, 2004.

[17] R. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation,* (4):323–344, 1987.

[18] G. Welch and G. Bishop. An introduction to the kalman filter. Technical report 95-041, University of North Carolina at Chapel Hill, Department of Computer Science, 1995.

[19] Zhigang Zhu, Guangyou Xu, Bo Yang, Dingji Shi, and Xueyin Lin. Visatram: a real-time vision system for automatic traffic monitoring. *Image and Vision Computing,* 18(10):781–794, October 2000.

Chapter 11

SUSTAINABLE CYBERNETICS SYSTEMS

Backbones of Ambient Intelligent Environments

A.H. Salden and MaKempen
Telematica Instituut, The Netherlands
{Alfons.Salden,Masja.Kempen}@telin.nl
Keywords: Ambient intelligence, field potentials and strengths, topological invariants, sustainability

Introduction

In artificial intelligence, cognitive science and cognitive engineering a main challenge for the next decades will be building sustainable cybernetic systems that can individually and/or collectively anticipate and attend to their own or environmental dynamics. The anticipation and attention measures foreseen and taken by cybernetic systems will determine decisively whether they can achieve their goals and accommodate their own and environmental changes. The current abundance and omnipresence of ICT architectures and infrastructures enable ubiquitous, pervasive, sentient, and ambient intelligent computing, communication, cooperation and competition of both artificial and societal organizations and structures. However, they appear to us as merely nice to haves in a rather unstructured and unorganized ICT infrastructure. In general they still lack cognitive engineering capabilities, namely those for anticipation and selection of attention. Instead of perpetually handcrafting standalone ICT infrastructures and integrating them, smart human-system network interaction paradigms are needed such that cybernetic systems can continuously select and embody (after reinforcement learning) suitable anticipatory and selection of attention schemes to bring those novel integrated ICT architectures and infrastructures to life. In short new paradigms for co-existence and co-evolution of humans, machines and their extensions are needed in order to simultaneously sustain both types of schemes. In this article we present a mathematical-physical framework that allows to model and deploy sustainable cybernetic systems.

A (sustainable) cybernetic system should realize its current states in terms of physical structures and organizations by taking into account, besides its past and present states, also its foreseen potential future states that can lead to the highest chance of fulfilling its current and future goals. Such states are embed-

ded and predicted by the system itself and/or enforced and communicated by its environment. Thus a cybernetic system should instruct itself to restructure and to reorganize itself in order to maximally achieve its own goals whatever that maybe. The goals in turn may be in line with constraints and opportunities put forward by such a system itself or by its environment. In this way a self-organization of the cybernetic system comes about that may guarantee the system's sustainability despite its possibly predestined own and environmental evolutionary dynamics [4]. The explorative and goal-directed behavior of a cybernetic system then displays itself not only as (reinforcement) learning, understanding and assessment of the system itself and its environment, but also as a functional re-organization and physical restructuring of a large number of its (imaginary) current and future states and/or organizations. Summarising continuous self-constrained functional reorganization and physical restructuring is a necessity - given its objectives/goals/tasks, internal and external states, and constrained and/or engendered by its co-evolving environment. Thus the question rises what causes self-structuring/assembly and self-organization of a cybernetic system and how can such a system embody that itself.

As stated, cybernetic systems should be endowed with anticipatory and attentive capabilities and capacities in order to tackle and cope with their own internal and environmental evolutionary pressures. The viability and sustainability of a cybernetic system in those aspects can be assessed on the basis of so-called fitness or intelligence measures for the anticipatory and selection of attention schemes, respectively, in dealing with dynamical changes of the system itself and its environment [4]. Up to today determining and making explicit proper fitness and intelligence measures are outstanding cybernetics problems with respect to self-organization and natural selection that are hardly ever satisfactory addressed. Fitness measures here we define as measures for the intertwining, entanglement and entrainment of (non-) local structures and/or organizations of the cybernetic system and those of its environment. Intelligence measures refer to the anticipatory and attentive (non-) local potentials (in solving existing, hidden and rising problems), embodied in the structure and the organization of the cybernetic system, relative to those potentials in its environment. Thus all this asks for smart anticipatory and attentive cybernetic systems that can simultaneously counterbalance or fuse both the phenotypic (evolutionary) and genotypic (stationary) dynamics of both the system itself and its environment. Therewith our initial problem of self-structuring and self-organization by the cybernetic system itself can be rephrased in terms of what are and how to embody/embed proper fitness and intelligence measures for anticipatory and attentive purposes.

Many scientists have proposed to define the above fitness and intelligence measures in terms of indicators or utility measures for self-organized critical states of social, biological, physical and ICT networks. Such indicators or utility measures relate to scaling laws (self-similarity) and symmetry breaking mechanisms of punctuated and far-off equilibrium network dynamics, respectively. These physical laws and mechanisms spell out which strategies are the most valuable ones that co-evolving systems could follow and apply during phases of self-organization and natural selection of anticipation and selection of attention schemes.

Having mastered such physical laws and mechanisms a cybernetic system should, besides anticipate, also know why, when and how to capture, to direct and to change attention towards relevant dynamical phenotypic and genotypic issues while enacting on itself, its collective and/or its environment. In this respect a cybernetic system should allow for the emergence of smart anticipatory and selection of attention schemes at appropriate spatio-temporal and dynamic scales. Of course, the latter schemes should couple to inference and association schemes that in turn are indispensable during anticipatory cycles, which certainly also include the common (explorative and intentional) perception-decision-action cycles occurring at lower dynamic evolutionary scales. Summarizing anticipatory or selection of attention (pre-) schemes of cybernetic systems together with their fitness and intelligence measures should upon enaction allow for the emergence of hierarchies of relevant niches for their own and environmental dynamics. Nevertheless, the question remains how to effectively embed and embody those schemes in networks of artificial systems, humans or their extensions.

How to realize (pre-) schemes for natural anticipation and selection of attention (NASA) in cybernetic systems is a problem that is hardly ever satisfactory tackled in computer and cognitive science. However, Ilya Prigogine in his Nobel Lecture [2] touches upon the fundamental perception-decision-action problem in science. Furthermore,Roger Penrose and Stuart Hameroff in their seminal work [8] dare to explain consciousness as a nonlocal quantum phenomenon in the brain. They emphasize the importance of first unraveling the relevant physical laws and mechanisms involved in cybernetic systems before one even can think of reaching any sensible levels of consciousness (awareness), understanding or intelligence.

Obviously, the fact that the evolution of a cybernetic system and its environment are difficult to access or to predict, makes a faithful structural embedding and functional embodiment of this natural system into a cybernetic system a real tour de force [3, 10]. Actually the perception-decision-action problem for cybernetic systems cannot be disentangled from the following more fundamental physical problems:

- Problem of measurement, calibration and gauging of an evolving cybernetic system: it relates to the problems of natural anticipation and selection of attention, causality, learning.

- Problem of complexity versus resolution of an evolving cybernetic systems: it relates to the problems of disorder versus order,predictability, renormalisation and sustainability (including scalability).

The above perception-decision-action problem has been adressed separately for one or a pair of modalities from a mono- or multi-disciplinary perspective. In [5, 10, 12] we proposed a mathematical physics framework that supports development and deployment of sustainable cybernetic systems. Our approach distinguishes itself from the mono- or multi-disciplinary approaches in the sense that the statistical physics geometry of the interacting environment, user and system are conceptually as well as data-driven physics-based. The other ap-

proaches advocate representing e.g. spatio- temporal ordering relations and derived geometric properties in terms of heuristic Euclidean invariant measures. Such measures are generally totally inadequate to capture in a robust and reliable way the statistical physics and geometry underlying interacting sustainable intelligent multimodal systems (SIMS) [13]. Furthermore, such approaches do not address possible coupling and associative (pre-) schemes between multimodalities. Our framework does not only allow robust and reliable modeling of complex systems, but also sustains acquisition of natural anticipation and selection of attention (pre-) schemes needed during co-evolution of humans and systems.

Intelligent agent systems are indispensable to create and sustain NASA (pre-) schemes. Specialization and leverage of these (pre-) schemes can be realized by collective intelligent human and software agent systems (CIA) [9]. At higher levels of network complexity similar systems can be posited for such purposes. Organizations, groups, individuals, ICT and knowledge systems all have limited capacities and capabilities. They need to free time and resources for differentiating, diversifying and integrating information and knowledge.

This chapter is organized as follows. In section 1 we present our mathematical-physical framework for encoding interplay and evolution of natural and cybernetic systems. In section 2 we show how to sustain natural and cybernetic systems by means of collective intelligent agent system architectures. We define measures for sustainability - the true measures for ambient intelligence - as topological currents supporting co-evolving (symbiotic) natural and cybernetic systems.

1. Encoding Interplay and Co-Evolution

In the sequel we present our mathematical physical framework to encode natural and cybernetic systems. In Section 1.1 we consider the machines of a natural system relevant for encoding a cybernetic system, whereas in Section 1.2 we consider those for co-evolving natural and cybernetic systems.

1.1 Encoding Interplay between Natural and Cybernetic Systems

Let us model an encoding I of an observable as a transduction (with losses) of a vector-valued current m representing a natural system M onto a vector-valued current n representing an (induced) cybernetic system N, and vice versa:

DEFINITION 11.1 *An encoding I of natural system M onto cybernetic system N is defined by a transduction I:*

$$I : M \rightarrow N; \; I(m) = n.$$

Each current is a density field for a physical system either being a natural or cybernetic system. The cybernetic system current is a superposition of several currents representing or consistent with a natural system and a dissipation current. Note that, the number of components of a cybernetic system will normally be less than that of a natural system. Under encoding there always occurs a

high level of structural and functional abstraction that is controlled by natural and cybernetic system requirements. Furthermore, the problem of optimal transduction coincides with the perception-decision-action problem.

Now an encoding of other observables derived from observable m follows upon finding a set of so-called equivalences of encoding (Definition 11.1) that is invariant under a gauge group:

DEFINITION 11.2 *A gauge group G on encoding is a set of transformations leaving invariant the encoding (Definition 11.1) of the natural system M, the transduction I and the cybernetic system N.*

The gauge group related to the encoding is normally given by a system of partial integro-differential equations with initial-boundary value conditions on *I*, *M* and *N*. In Section 1.2 we make the gauge group explicit as the evolution of the natural system. We observe that this group and the evolution of a natural system cannot be disentangled. Thus observables do not have any meaning before having that evolution empirically derived and verified. Furthermore, that a gauge group does not have to satisfy at all the standard group properties as the existence of an inverse. For example, if the gauge group is covered by a renormalisation group that in turn is defined by a particular diffusion equation, then the flow is uniquely directed with no possibility of return or defining an inverse. Therefore, it might be more suitable to rename in the future a gauge group into a gauge renormalisation functor category. Again in Section 1.2 and in Section 2 we elaborate on the implications of these issues; in the sequel we just assume that a gauge group possesses all the group properties.

Now the question arises how to derive a set of equivalences of encoding (Definition 11.1) invariant under a gauge group. There exist in the mathematical literature several rich methods to systematically retrieve such a set of equivalences based on (for references see [10]:

- Differential and integral geometry,
- Lie group theory applied to system of partial integro-differential equations with initial-boundary value conditions,
- Algebraic and differential topology.

In the following we briefly summarise the differential and integral geometric method for obtaining such a set of equivalences. For applications of this method to computer vision problems the reader is referred to [5, 6, 12]: the mathematical-physical objects referred to in the sequel are therein made explicit Analogously the same methods can be applied to multimodal interaction and other problems [13]. However, in Section 1.2 we will see that only a set of equivalences consistent with an encoding of the evolution of a natural system can come about upon application of the other methods as well.

A set of equivalences of the encoding (Definition 11.1) invariant under a gauge group (Definition 11.2) come about after setting up a (co)frame field, metric and/or connection invariant under the gauge group.

Let us make explicit the structure and function of these physical objects in a cybernetic system.

DEFINITION 11.3 *A frame field* (ϕ_p) *is a section of the tangent bundle T of encoding (Definition 11.1).*

By exponentiating the frame vector field ϕ_p one obtains a parametrisation or labelling of the natural system *M*, the transduction *I* and the cybernetic system *N*.

DEFINITION 11.4 *A coframe field* (dv^p) *is a section of the cotangent bundle T* of encoding (definition 11.1).*

The coframe field in combination with the frame field allow the measurement of actions of the gauge group (Definition 11.3) on the encoding (Definition 11.1).

Frame field (Definition 11.3) and coframe field (Definition 11.4) then satisfy not necessarily a duality constraint:

DEFINITION 11.5 *A frame field (Definition 11.3) and a coframe field (Definition 11.4) of the encoding (Definition 11.1) are their duals, if and only if:*

$$d\phi^p(\phi_q) = \delta^p_q,$$

where δ *is the Kronecker delta-function.*

DEFINITION 11.6 *A metric tensor* γ *is a (non-degenerate) bilinear form on* $T \times T$ *such that*

$$\gamma : T \times T \to \mathrm{K};\ \gamma(\phi_\mathrm{p}, \phi_\mathrm{q}) = \gamma_\mathrm{pq},$$

where γ_{pq} *are the components of the metric tensor, and K is the field of numbers measuring distances and angles of actions of the gauge group (Definition 11.2).*

A metric not only allows to measure classical notions of distance and angle such as Euclidean distance between points. Differences in other more complex gauge group actions related to e.g. energy or vector potentials can also be read out.

DEFINITION 11.7 *A connection* Γ *on encoding (Definition 11.1) is defined by one-forms* o^q_p *on its tangent bundle T:*

$$\nabla^\Gamma : T \to T \times T^*; \nabla^\Gamma \phi_p = \omega^q_p \otimes \phi_q,\ \omega^q_p(v_r) \in \mathrm{K},$$

where $\otimes$ *denotes the tensor product, and* ∇^Γ *is the covariant derivative on encoding (Definition 11.1), and* K *a field of scalar numbers representing physical observations related to gauge group actions in (Definition 11.2).*

The metric (Definition 11.6) and the connection (Definition 11.7) are in general assumed to be compatible with each other.

DEFINITION 11.8 *The metric (Definition 11.6) and the connection (Definition 11.5) are compatible if and only if:*

$$\nabla^\Gamma g = 0.$$

This means that, e.g., the angles between and lengths of vectors measured by the metric tensor under the parallel transport associated with the connection are being preserved. Note that, as will become clear shortly, it is not necessary to require the existence of a so-called metric connection (connection is fully determined by a metric; connection coefficients expressible in terms of derivatives of the metric components with respect to the frame fields) to derive a suitable set of equivalences of the encoding (Definition 11.1). The stipulation of a connection or a Lie derivative is in general necessary and sufficient.

In order to categorise encoding (Definition 11.1) so-called curvatures of frame fields in (Definition 11.3) are read out.

DEFINITION 11.9 *The curvature Φ_i of a frame vector field ϕ_i in (Definition 11.3) at point p on a two-dimensional surface S parametrised by frame field (Definition 11.3) is defined by:*

$$\Phi_i(p, S) = \oint_C \nabla^\Gamma \phi_i,$$

where the sense of traversing circuit $p \in C$ is chosen such that the interior of the circuit C on surface S is to its left.

Using Stokes' theorem curvature (Definition 11.9) can be expressed as:

$$\Phi_i(p, S) = \int_{C^\circ \subset S} \nabla^\Gamma \wedge \nabla^\Gamma \phi_i = \int_{C^\circ \subset S} \Omega_i^j \phi_j,$$

where C° is the interior of the circuit C in S, $\nabla^\Gamma \wedge$ the covariant exterior derivative in which $\wedge$ is the wedge product consistent with metric (Definition 11.6) and/or connection (Definition 11.7), and Ω_i^j represent the so-called curvature two-forms. These curvature two-forms measure the inhomogeneity of the gauge group actions (Definition 11.2). They are quite common in differential geometry, defect theory, gauge field theory and general relativity. Only lately theur were introduced in computer vision and cybernetics in general [5, 13]. Note that they should not be confused with the curvatures that appear as invariant functions (zero-forms) under a gauge group in equivalence problems concerning, e.g., planar curves under the group of Euclidean movements. Actually they form the desired topological charges, potentials and curvatures for pinpointing down the structure and function of natural system, transduction and cybernetic system.

From these curvatures we can in turn derive higher order curvatures $\Phi_{i;j_1 \ldots j_k}$ by taking successively covariant derivatives $\nabla_{v_{j_l}}$ with respect to frame vector fields ϕ_{j_l}. Together they form the first set of locally and directionally equivalences that quantify encoding (Definition 11.1).

EQUIVALENCE 1 *The set of directional equivalences of encoding (Definition 11.1) is given by:*

$$\Phi_{i;j_1 \ldots j_k} = \nabla^\Gamma_{\phi_{j_k}} \cdot \ldots \cdot \nabla^\Gamma_{\phi_{j_1}} \Phi_i,$$

where ; indicates taking a covariant derivative.

If there are symmetries underlying the encoding, then it is worthwhile to try to find the irreducible equivalences [5]. (Equivalence 1) allows the quantification of the inhomogeneity or coherence of the encoding (Definition 11.1): it allows even to bridge the heterogeneity in encodings.

If we consider a set of circuits, *{C}*, on a set of related surfaces, *{S}*, through point p, then (Equivalence 1) at p satisfies obviously a local conservation law (superposition principle) such that the directional information will become obsolete.

EQUIVALENCE 2 *A set of local equivalences of encoding (Definition 11.1) is given by:*

$$\bar{\Phi}_{i;j_1\ldots j_k}(p,\{S\}) = \sum_{\{S\}} \Phi_{i;j_1\ldots j_k}(p,S),$$

being total curvatures of the vector fields ϕ_i in frame field (Definition 11.3) over the set of all surfaces, {S}, each of which contains one corresponding circuit C, through point p.

These local equivalences explain the problem of describing, for example, encodings solely on the basis of a local analysis. A local analysis, namely, maps a sequence of directionally equivalences onto one number. Recall, that point p is an initial point on C and that this point and the diametrically located end-point of C in curvature (Definition 11.10) determine actually a direction.

(Equivalence 2) can be complemented by a set of global equivalences for a region U of the encoding (Definition 11.1).

EQUIVALENCE 3 *A set of global equivalences of encoding (Definition 11.1) is given by:*

$$\tilde{\Phi}_{i;j_1\ldots j_k}(\{S\},U) = \int_U \bar{\Phi}_{i;j_1\ldots j_k}(p,\{S\})dU,$$

in which U is a region on N not necessarily of constant dimension nor simply connected to point p.

Up to now we unravelled only sets of local and directionally equivalences over either simply or multiply connected regions. The question arises what are possible interactions between such currents?

First of all there would be a gauge group transformation needed to bring the (co)frame fields and connections at two (more than two) positions in line. The latter gauging will be noticeable for an external observer. An internal observer, the point of view chosen in our framework, will not be aware of this action as he or she is falling freely along some kind of geodesic. Now we are in the position to carry out the comparison by establishing a set of joint equivalences, comparable to semi-differential, multi-local, simultaneous or joint (differential) invariants [5, 12].

Assuming the considered gauge group to be living on the whole encoding (if different gauge groups would be applicable over subencodings the analysis does not deviate) the most immediate construction follows upon computing the structure functions for a multiple point set of morphisms. The latter set should not be confused with the gauge group. The elements of this set of transformations are generated by a set of gauge invariant propagators:

$$\Psi_{pj_1\ldots j_k} = \Phi^{p}_{i;j_1\ldots j_k} \nabla^{\Gamma(p)}_{\phi_i(p)},$$

where p labels a selected point. The Lie algebra of this set of propagators determines the sought set of joint equivalences.

EQUIVALENCE 4 *A set of joint equivalences of encoding (Definition 11.1) is given by the structure functions* Φ^{r}_{pq}*:*

$$[\Psi_p, \Psi_q] = \Phi^{r}_{pq} \Psi_r,$$

where Ψ*'s are gauge invariant propagators.*

Note that ratios or differences of the components of the aforementioned equivalences are also joint equivalences but can be retained by simple algebra on the set of local and directional equivalences.

Up to now we have only considered circuits C. Applying homotopy and homology theory it is clear that among the set of local and directional equivalences there are also topological invariants such as winding numbers for higher order groups of homotopy, and Betti-numbers for higher order groups of cohomology. These invariants require the analysis of a normalised hypervolume-form on a hypersphere rather than a circle C and higher order differential forms both invariant under the gauge group (Definition 11.2) and consistent with encoding (Definition 11.1). For recent applications of homotopy and homology theory to computer vision and cybernetics the reader is referred to [5, 7, 12, 13].

More importantly, in establishing sets of local, global and joint equivalences we restricted ourselves to a mere integration of comparable structural equivalences over an object or to some multi-local interaction between them, respectively. Being aware of (self)-interactions in a natural system one would like in addition to retrieve a set of functional equivalences of encoding (Definition 11.1). Among these equivalences are (generalised) Vassiliev invariants, linking number and Möbius energies for knots, links, braids and more general CW-complexes but as will see also co-evolving systems.

EQUIVALENCE 5 *A set of functional equivalences* $\tilde{V}$ *of encoding (Definition 11.1) are (generalised) Vassiliev invariants, linking numbers and Möbius energies of CW-complexes.*

In Section 1.2 these and the other type of equivalences appear to be just (topological) invariants of specific homotopy and cohomology groups.

An important property of a physical object or process F contained in encoding (Definition 11.1) is its invariance under the gauge group (Definition 11.2).

DEFINITION 11.10 *A physical object or process F, consistent with encoding (Definition 11.1), is invariant under gauge group (Definition 11.2) if and only if:*

$$GF = F.$$

All above mentioned physical objects and (either reversibel or irreversible) processes are by definition unaffected by the corresponding gauge group.

THEOREM 11.11 *The frame field (Definition 11.3), coframe field (Definition 11.4), metric (Definition 11.5), connection (Definition 11.6), and equivalences (Equivalence 1), (Equivalence 2), (Equivalence 3), (Equivalence 4) and (Equivalence 5) are invariant under gauge group (Definition 11.2).*

PROOF 1 *Proofs follow simply by applying gauge group to the induced encoding (Definition 11.1), computing the geometric objects and comparing them with the initial ones.*

Note that in gauge field theories one normally defines a gauge transformation as an inhomogeneous invertible linear transformation of the frame field. This transformations leads subsequently to a transformation of the gauge fields, i.e. the connection one-forms. The field strengths of the gauge fields, i.e. the curvature two-form valued vector fields, behave then covariantly. The only equivalences of the encoding numerically invariant under this gauge group are topological invariants that correspond to the Jordan normal forms of the matrix representations of the curvature two-forms. These topological invariants concisely represent groups. The operationalisation, i.e. quantification and qualification, of the functor category of such groups plays an important role in the actual encoding of the evolution of a natural system (see Section 1.2). Also we come back therein why we don't make a distinction between reversible and irreversible process.

Summarising, we have presented a method for encoding the interplay among cybernetic and natural systems in terms of equivalences corresponding to a frame field, coframe field, metric and connection invariant under a gauge group. This method allows us to retrieve local, multi-local, nonlocal, global and topological information about the encoded observable and evolution of the encoding. The latter information can be of help in order to steer policies and ensure the sustainability of the natural system as well as the cybernetic system. The observables and their equivalences invariant under the gauge group is fully determined by the evolution of the natural system. This implies also that the gauge group to be considered is intimately related to that evolution. Moreover, that the analysis of the evolution of the natural system in terms of measures not invariant under a gauge group consistent with this evolution cannot be of any use: they can never form the proper reference variables or indicators for sustainable cybernetic systems. This principle of gauge invariance or equivalence is by the way at the very heart of any relativity or renormalisation theory and forms an integral part in the encoding of the co-evolution of a natural system and a cybernetic system in the next section.

1.2 Encoding Co-Evolution of Natural and Cybernetic Systems

In cybernetic system theory it is common use during modeling phase of the natural or artificial system to postulate a posteori some hypotheses concerning its dimensionality and the laws underlying its evolution. One fixes the number of independent and dependent variables or observables, and the evolutionary laws expressed in these observables. We like to coin latter systems as closed. Next to these closed systems are open systems that are characterised over time by varying dimensionality and changing evolution laws. Closed systems are like their simulations reactive and very rigid in their formulation, whereas open systems are characterised by fuzzy anticipatory and predictive controls of their states and laws. Nevertheless, an open system can be covered by a closed system during a certain period.

Let us give some examples to illustrate the above classification of natural and cybernetic systems.

EXAMPLE 11.12 *Meteorological and climatological systems are in general modelled as closed. The systems are represented by a fixed set of observables governed by a fixed set of conservation laws normally represented by a deterministic system of PDES with initial-boundary conditions.*

EXAMPLE 11.13 *Anthropological systems such as those for politics, economy and technology are very open. The economical system over time shows emergence and annihilation of observables such as currencies, and of evolutionary laws for them. E.g., one has left the gold standard for the dollar as reference currency. Furthermore, the trading rules with respect to derivates have become ever more intricate due to changes in market rules and in legislation.*

EXAMPLE 11.14 *Biological systems are from the very onset designed to be open and in particular to be adaptive to changing environmental conditions and to become anticipatory and thus predictive both by learning to ensure persistence of the systems themselves.*

In closed systems the notion of the observables is clear, whereas for open ones they seem to become more and more obscure due to fact that the evolution laws also change over time. However, over time also open systems display changes in the encoding. These changes are caused by a disaggregation of the observables and the deterministic system as a whole, and by a true evolution of the deterministic system from one to another.

In Section 1.2.0 we explain how to arrive at a functor category of evolutions that can serve as framework for encoding the evolution of natural and cybernetic systems. In Section 1.2.0 we propose to use this framework to quantify a path of natural transformations between different evolutionary phases of the system, analogously a renormalisation theory in theoretical physics does with the difference that we are not forced to choose a group-like relation between the evolutionary phases. Structural and functional transitions in the dynamics are very well possible and are in sustainable systems certainly to be welcomed.

For the proofs of the theorems and alike in the remainder of this section the reader is kindly referred to the listed references in [10].

Preliminaries. In the sequel we merely briefly point out how to arrive at a functor categorification of co-evolving natural and cybernetic systems. In this context the decision problem in topology plays a central role. This problem concerns whether or not two topological spaces are homeomorphic, i.e. whether there exists an invertible one-to-one and onto mapping between them that is continuous. This problem can be solved by associating topological invariants (groups) to such spaces and comparing them. This solution to the decision problem is based on a so-called decategorification, i.e. forgetting about morphisms or in this case homeomorphisms of the topological spaces. We will observe that there are different categorifications possible, namely as homotopy groups and cohomology groups. But we will also realise that these categorifications, functors, can constitute objects in a functor category with natural transformations between them as morphisms. Finally, we carry this process of categorification over onto that of co-evolving natural and cybernetic systems.

Let us first start dwelling on the notion of a category.

DEFINITION 11.15 *A category consists of objects and morphisms between the objects.*

A category of all topological spaces consists of objects, i.e. all topological spaces M belonging to a particular type of mathematical universe U, and of morphisms, i.e. all continuous mappings f, of pairs $M, N \in U$ from M to N. Besides this category in the sequel also a category of all groups G in U is essential in the process of encoding evolution. It consists of individual groups as objects and group homomorphisms f^{***} between the groups as morphisms.

Categories can be studied by means of structure preserving mappings between them called functors.

DEFINITION 11.16 *Let $\mathcal{A}$ and $\mathcal{B}$ be categories. $F : \mathcal{A} \rightarrow \mathcal{B}$ is called a covariant functor, if $F(a, f^a) = (b, f^b)$ with $f^b = F(f^a) : F(A) \rightarrow F(B)$ and F respects the rule of composition of mappings: $f^{cb} f^{ba} : A \rightarrow C$ implies $F(f^{cb} f^{ba}) = F(f^{cb}) F(f^{ba}) : F(A) \rightarrow F(C)$ and identity mapping $F : A \rightarrow A$ implies identity mapping $F(f) : F(A) \rightarrow F(A)$.*

Analogously one defines a contravariant functor $\tilde{F} : B \rightarrow A$.

In order to relate and to compare functors F_1 and F_2 one uses the notion of natural equivalence and natural transformation.

DEFINITION 11.17 *Let $\mathcal{A}$ and $\mathcal{B}$ be categories. Functors $F_1, F_2 : \mathcal{A} \rightarrow \mathcal{B}$ are naturally equivalent if for all pair of objects (A, B) in category pair $(\mathcal{A}, \mathcal{B})$ there exists isomorphisms $\psi(A, B)$ and $\phi(A, B)$ in category $\mathcal{B}$, such that $\phi(A, B)(F_1(f)) = F_2(f)(\psi(A, B))$.*

DEFINITION 11.18 *Let $\mathcal{A}$ and $\mathcal{B}$ be categories. Functors $F_1, F_2 : \mathcal{A} \rightarrow \mathcal{B}$ are naturally transformations of one another if for all pair of objects (A, B) in category pair $(\mathcal{A}, \mathcal{B})$ there exists transformations, i.e. no isomorphisms, $\psi(A, B)$ and $\phi(A, B)$ in category $\mathcal{B}$, such that $\phi(A, B)(F_1(f)) = F_2(f)(\psi(A, B))$.*

Thus by means of the natural equivalence or transformation defined by the set $\{(\psi, \phi)\}$ it is possible to map the encoding of properties of the functor F_1 into those of functor F_2.

Now we are ready to define the functor category on the basis of the set of functors F_1 and F_2 from category $\mathcal{A}$ to category $\mathcal{B}$ and the set of all natural transformations from functor F_1 to functor F_2.

DEFINITION 11.19 *The functor category consists of functors as objects and natural transformations as morphisms.*

Categorification follows upon adding to the topological spaces or groups the morphisms between them. It is based on the analogy between sets and categories. There is the analogy of equations between elements and isomorphisms between objects, sets and categories, functions and functors, and equations between functions and natural transformations between functors.

In the next paragraphs we illustrate the process of functor categorification for higher order homotopy groups and higher order cohomology groups of topological spaces. We will see that these groups can be decategorified by a system of winding numbers and a set of Betti numbers, respectively.

Higher order homotopy groups. In order to define equivalences of a topological space Y the homotopy of mappings between a topological space X and Y is essential.

DEFINITION 11.20 *Two continuous mappings*

$$f, g : X \rightarrow Y$$

from topological space X onto Y are homotopic,

$$f \sim g,$$

if there exists a mapping

$$F : X \times [0,1] \rightarrow Y,$$

such that

$$F(x,0) = f(x), \;\; F(x,1) = g(x).$$

The space of the equivalence classes of mappings is denoted by $[X, Y]$. Obviously, here, X serves as some reference space in order to assess structures of Y.

The most important homotopy groups are the fundamental group and higher order homotopy groups of a n-dimensional manifold $Y = M$. In order to define these groups we need first a notion of loops X and next that of a product of loops.

DEFINITION 11.21 *A* ***k-loop*** *is a mapping l of cube I^k in M that maps the boundary ∂I^k on a fixed point $x_0 \in M$:*

$$\begin{aligned} l(t) &\in M, \;\; t_i \in [0,1] \\ l(t) &= x_0, \;\; t_i = 0, 1. \end{aligned}$$

Note that, homotopy of k-loops is a conjugacy of k-loops, i,e. there exists a continuous mapping ψ of the k-cube I^k on the space of k-loops l such that two k-loops l and l^* are joined by ψ with $\psi(t_i = 0) = l$ and $\psi(t_i = 1) = l^*$.

DEFINITION 11.22 *The product of two k-loops l and m is defined by:*

$$(l \cdot m)(t) = \begin{cases} l(2t_1, \ldots, t_k) & t_1 \in [0, \frac{1}{2}] \\ m(2t_1 - 1, \ldots, t_k) & t_1 \in [\frac{1}{2}, 1] \end{cases}$$

A unit element, associativity and an inverse under the above multiplication rule for k-loops are now readily defined. Furthermore, the set of homotopy classes of k-loops with this multiplication $[l] \cdot [m] = [l \cdot m]$ can easily be proven to form a group. Thus we are in the position to define the higher order homotopy groups.

DEFINITION 11.23 *The k-th order homotopy group $\pi_k(l(x_0), M)$ is the set of homotopy classes of k-loops with multiplication $[l] \cdot [m] = [l \cdot m]$.*

If topological space M is simply connected, then it can be shown that $\pi_k(l(x_0), M) = \pi_k(l(x_1), M), \forall x_0, x_1 \in M$. Furthermore, that $\pi_k(l(x_0), M_1 \times M_2) = \pi_k(l_1(x_0), M_1) \times \pi_k(l_2(x_0), M_2)$.

It's straightforward to define whenever two spaces are homotopic and which conditions have to be satisfied.

DEFINITION 11.24 *Two spaces M and N are of the same homotopy type, $M \sim N$, if there exist mappings $f : M \rightarrow N$ and $g : N \rightarrow M$ such that the compositions $f \odot g \sim id_N$ and $g \odot f \sim id_M$.*

If this homotopy holds it's readily proven that $\pi_k(l_m, M) = \pi_k(l_n, N)$.

The higher order homotopy groups $\pi_k(l(x_0), M)$ can be decategorified in terms of winding numbers.

DEFINITION 11.25 *With the k-th order homotopy groups $\pi_k(l(x_0), M)$ are associated winding numbers ν_k of mappings $f : S^k \rightarrow M$ defined by:*

$$\nu(f) = \int_{S^k} f^* \omega,$$

where S^k a k-sphere, ω is a normalised volume-form on S^k and f^* is the pull-back of the mapping f.

Note that these winding numbers for the higher order homotopy groups are just a small subset of the class of functional equivalences (Equivalence 5) defined in Section 1.1. For applications of such a decategorification within the realm of computer vision the interested reader should turn to [5, 7, 12].

Higher order cohomology groups. In order to retrieve topological invariants of a manifold M one may study function spaces on M and in particular the (de Rham) cohomology groups on M. In order to make these cohomology groups explicit we need first to define the space of closed and exact k-forms:

DEFINITION 11.26 *The space* $Z^k(M)$ *of closed k-forms* $\alpha \in \Omega^k(M)$ *on manifold* M *is defined by:*

$$Z^k(M) = \left\{ \alpha \in \Omega^k \,\|d\alpha = 0 \right\},,$$

where d *denotes the ordinary exterior derivative operator.*

In applications we cannot restrict ourselves to these ordinary derivatives; adaptation of the derivatives and the function spaces considered on the manifold M to the considered gauge group (Definition 11.2) becomes really indispensable.

DEFINITION 11.27 *The space* $B^k(M)$ *of exact k-forms* $\alpha \in \Omega^k(M)$ *on manifold* M *is defined by:*

$$B^k(M) = \left\{ \alpha \in \Omega^k(M) \,\middle\|\, \exists \beta \Omega^{k-1}(M) \alpha = d\beta \right\}.$$

Obviously, here, exactness implies closedness.

Although every closed form can be locally written as an exact form there exist global topological obstructions determined by the cohomology groups.

DEFINITION 11.28 *The cohomology groups* $H^k(M); k = 0, \ldots, n$ *on manifold* M *is defined by:*

$$H^k(M) = Z^k(M)/B^k(M).$$

Thus with each $\alpha \in Z^k(M)$ is associated an equivalence class $[\alpha] \in H^k(M)$ with $[\alpha] = [\alpha + d\beta], \beta\Omega^{k-1}(M)$.

Now the cohomology groups $H^k(M)$ are topological invariants of manifold M:

THEOREM 11.29 *If* $\phi : M \rightarrow N$ *a diffeomorphism, then* $H^k(M) \sim H^k(N)$.

Note that every closed form can locally be written as an exact form on the basis of the cohomology ring $H^*(M) = \bigoplus_{k=0}^{n} H^k(M)$.

The topological invariants of a compact manifold M can subsequently be represented by Betti-numbers.

DEFINITION 11.30 *The Betti-numbers* b_k *of a manifold* M *are defined by:*

$$b_k = \dim H^k(M).$$

For example, b_0 measures the number of path-connected components of M. If $b_0 = b_n = 1$, then the manifold M is simply connected as well as orientable. In that case the other Betti-numbers are related to each other through the Poincaé duality: $b_k = b_{n-k}$. Moreover, we can define the so called Euler characteristic.

DEFINITION 11.31 *The Euler characteristic* χ *of manifold* M *is defined by:*

$$\chi(M) = \sum_{k=0}^{n} (-1)^k b_k.$$

The latter topological information can be cast into the Poincaré polynomial $P_t(M) = \sum_{k=0}^{n} b_k t^k$,i.e. $\chi(M) = P_{-1}(M)$. For this polynomial the so-called Künneth formula holds: $P_t(M \times N) = P_t(M)P_t(N)$ for a product-space of manifolds.

For applications of this kind of decategorification in computer vision or cybernetics in general the interested reader may turn to [5].

Functor Category. In the sequel we show that the homotopy and cohomology groups are generated by a pair of corresponding functors on the category of topological spaces. Moreover, we will observe that this pair of functors are related through a natural transformation.

Homeomorphic topological spaces can be associated isomorphic higher order homotopy groups.

THEOREM 11.32 *If the k-th order homotopy group $\pi_k(M)$ on topological space M is isomorphic with the k-th order homotopy group $\pi_k(N)$ on topological space N, then M is homeomorphic with N (Necessary condition).*

If f a homeomorphims between M and N, then $f^* = \pi_k(f)$ is a group homomorphism between $\pi_k(M)$ and $\pi_k(N)$. Here π_k are functors associating with any topological space its corresponding higher order homotopy group $\pi_k(M)$. Summarising, one has a conjugacy relation f^* between higher order homotopy groups that are associated to topological spaces that are related to each other through a homeomorphism f. Note that, this conjugacy differs from the conjugacy of k-loops mentioned above.

Similarly, homeomorphic topological spaces can be associated isomorphic cohomology groups.

THEOREM 11.33 *If the k-th order cohomology group $H^k(M)$ on topological space M is isomorphic with the k-th order cohomology group $H^k(N)$ on topological space N, then M is homeomorphic with N (Necessary condition).*

If f a homeomorphims between M and N, then $f^{**} = H^k(f)$ is a group homomorphism between $H^k(M)$ and $H^k(N)$. Here H^k are functors associating with any topological space their corresponding higher order cohomology groups. Summarising, one has a conjugacy relation f^{**} between higher order cohomology groups that are associated to topological spaces that are related to each other through a homeomorphism f.

Thus rules π_k and H^k are both covariant functors of the category of topological spaces into the category of groups. More precisely, topological spaces are associated with groups and homeomorphisms with group homomorphisms by both functors. The above pair of functors and the natural transformation f^{***} between them constitutes an object and a morphism in a functor category. This categorification permits us in turn to quantify the extent in which the modeling of a topological space M by π_k deviates from that by H^k.

Categorification of Encoding Interplaying Systems. Up to now we have applied category theory over topological spaces without further specifying these

spaces. The question arises how such a category looks like over natural and cybernetic systems. For a fixed space-time region the natural and cybernetic system can be considered closed and its categorification can be based on several classification methods.

The developmental dynamics of the natural and cybernetic systems can then be represented by a system of partial integro-differential equations with initial and boundary value conditions.

LAW 1 *The developmental dynamics of the closed cybernetic system is governed by the following law:*

$$\frac{d\psi}{dt} = e(\psi; \alpha),$$

with suitable initial and boundary value conditions and where t is time, ψ is a set of encodings e denotes the driving force on the encodings and a specifying a classification of the developmental dynamics.

The classification a of the closed natural and cybernetic systems can be brought about by considering (see [10] and references therein):

- Symmetries,
- Curvatures,
- Conservation laws.

Consequently the classification α can be stated as follows:

CLASSIFICATION 1 *The classification α of a closed cybernetic system is given in terms of symmetries, curvatures and conservation laws underlying the developmental dynamics in (Law 1).*

THEOREM 11.34 *(Classification 1) is gauge invariant under the symmetries of the developmental dynamics in (Law 1).*

One retains the symmetries, curvatures and conservation laws for those systems, which are not necessarily in divergence form, through the use of symbolic packages. These classifications yield a set of equivalences as in Section 1.1 but now for each closed system another set. Obviously the system category consists of natural and cybernetic systems and of each their symmetries. Actually the functors used in the previous paragraphs can be shown to generate the topological invariants of (Classification 1) of the encoding of the natural and cybernetic systems. The classification comprises besides aspects of the developmental dynamics of the natural and cybernetic systems also initial and boundary value conditions that can form constraints within the system.

For one equivalence given (Classification 1), for example, a conservation law, it is possible to derive a hierarchy of infinitely many equivalences. The methods for finding related conserved densities and fluxes should then not be confused with those for finding the equivalences in a standard setting as in

Section 1.1. The gauge group and thus the related equivalences are namely bound to be consistent with the symmetries of the developmental dynamics of the cybernetic system. The use of recursion operators that can generate a hierarchy of infinitely many symmetries then come into play to derive novel conservation laws [12].

Now the natural and cybernetic systems categorification consists of defining those systems as systems of partial integro-differential equations with initial-boundary value conditions, isomorphisms between them, functors over them and natural transformations between those functors. The decategorification occurs upon forgetting about the isomorphisms and retaining the equivalences of the classification. In reality our encoded natural and cybernetic systems trace out some region of the system categorification: there is an evolution of the natural and cybernetic systems.

The cybernetic system, however, can only partly follow over time a natural system, or vice versa, due to limited resources. This optimal abstraction, as reckoned in the introduction, implies a certain level of resolution of the encoded natural and cybernetic system dynamics, i.e. the resolution with which equivalences supported by those systema are observable or foreseeable. Thus in order to stay as close as possible to the natural system it's essential to couple the cybernetic system in an optimal way to the natural one, an vice versa. In doing so a higher resolution cybernetic system should be projectable onto a lower resolution one while satisfying superposition principles for the equivalences. The reader rightfully can object that it's sheer impossible to predict such a path or sweeping out system cateorification space given the counterintuitive behaviour of subsystems such as that of the anthroposphere. However, as we will demonstrate in the next section behaviour of a cybernetic system can and should be modelled simultaneously in order to be consistent with the natural system itself. From the standpoint of scenario analysis the latter behaviour manifesting itself in terms of bifurcations in the evolution of the cybernetic system can be of great importance in their own right too. But a suitable average of these bifurcating evolutions may be equally well valuable (see also Section 2). Note that emergence and annihilation of cybernetic system dynamics is no real problem as they can be readily be covered by natural transformations as will be demonstrated in the next section.

Categorification of Encoding Co-Evolving Systems. The decategorification of the natural and cybernetic systems, (Classification 1), obviously, is a function of time-parameter t. The evolution of this decategorification depends on the developments perceivable on a larger time-span. Thus different scenarios, for instance, will cause in general different evolutions of (Classification 1). In the context of sustainable development of above systems this evolutionary dynamics asks for a suitable selection mechanism. This mechanism in turn requires measures of sustainability β weighting the different evolutionary dynamics. As the natural and cybernetic systems can be viewed as essentially open because of the variability of (Classification 1), these systems can again be

represented by systems of partial integro-differential equations with initial and boundary value conditions.

LAW 2 *The developmental and evolutionary dynamics of the open cybernetic system is governed by the following law:*

$$\begin{aligned} \frac{d\psi}{dt} &= e(\psi;\alpha), \\ \frac{d\alpha}{dt} &= E(e,\alpha;\beta), \end{aligned}$$

with suitable initial and boundary value conditions and where t *is time,* ψ *is a set of encodings,* e *denotes the driving force on* ψ*,* α *specifying (Classification 1),* E *the driving force on (Classification 1) and* β *denotes the sustainability measures.*

A major challenge for future research is to make such a developmental and evolutionary dynamics explicit through an effective scheme to obtain objective classifications, sustainability measures β and driving forces E on these classifications given e and β.

Analogous to the developmental dynamics of a closed cybernetic system the developmental and evolutionary dynamics of open natural and cybernetic systems can be classified. From analogy with the developmental dynamics it's clear that in this case the measures of sustainability β in this case characterise both the open natural and cybernetic systems.

CLASSIFICATION 2 *The classification* β *of open natural and cybernetic system is given in terms of symmetries, curvatures and conservation laws underlying the developmental and evolutionary dynamics in (Law 2).*

THEOREM 11.35 *(Classification 2) is gauge invariant under the symmetries of the developmental and evolutionary dynamics in (Law 2).*

The above description of a cybernetic system parallels that for biological systems [3], in which (Classification 1) represents the genomic and environmental characteristics and (Classification 2) represents measures of adaptation of the biological systems as a whole to a particular class of phenotypical and genomical dynamics. Moreover, many other natural systems, e.g. learning systems generating predictive models, are subjected to similar modeling relations. Above it's silently assumed that all these systems form also an integral part of sustainable systems.

Contrary to Rosen's exposition [3] in our framework the natural and cybernetic systems as a whole cannot be disentangled from their subsystems. They form in general each an integral part of one another. For example, an organism and a cell nor an individual and a society can be dissolved from one another. This inseparability of a system and its subsystems is manifest in the coupling between developmental and evolutionary dynamics in (Law 2). This inseparability also manifests itself as the fact that natural and cybernetic systems in essence are nonintegrable and are part of and subjected to irreversible processes.

A cybernetic or natural system with a parasitic developmental and evolutionary dynamics is bound to become extinct because of e.g. environmental deprivation of resources. In order to prevent such catastrophes from happening it is necessary and sufficient to incorporate constraints or control mechanisms in the cybernetic system. The latter build in restrictions on the developmental and evolutionary dynamics permit then the survival of the subsystems and the systems as a whole. But the latter controlled natural or cybernetic system obviously asks for a anticipatory, learning and attentive system, such that an evaluation and validation of (Classification 2) is feasible, and such that in the end their outcomes can be used to direct their actions and evolutions. The essence of a predictive system, however, is that it's also capable to foresee the nonintegrable changes in the developmental and evolutionary dynamics in (Law 2) of natural systems. Deploying cybernetic system related to a natural system asks thus for the application of renormalisation theory [12].

From our above exposition it's more than reasonable to link the various measures of sustainability, (Classification 2), to measures of structural and functional stability of natural and cybernetic systems under genomic or evolutionary changes, and to link (Classification 1) to measures of Lyapunov stability under perturbations of the observables. In Section 2 we introduce a method to filter natural and cybernetic systems in developmental as well as evolutionary sense, such that a stable classification of the cybernetic system can be retained despite possible perturbations within the developmental and evolutionary dynamics.

2. Sustaining Ambient Intelligence

In the analysis of co-evolving natural and cybernetic systems it makes no sense to stick to the highest resolution information about their behaviours, because their developmental and evolutionary dynamics are depending on the specific intricacies of their encoding procedures. However, larger scale aspects of these dynamics will be partially equivalent or even the same despite the fact that cybernetic and natural systems will display at highest resolution levels very different characteristics. A cybernetic system consistent filtration of the developmental as well as the evolutionary dynamics, according to a dynamic scale-space paradigm [5, 10, 12] just ensures the (partial) equivalence of both the cybernetic and the natural system dynamics. In Section 2.1 we introduce an exchange principle that allows us to diffuse and stabilise the encoded evolution of the natural and cybernetic systems in order to achieve simultaneously Lyapunov and structural stability for the developmental and evolutionary dynamics, respectively. In Section 2.2 we recapitulate the methods of the previous section for retrieving reliable indicators for sustainability of natural and cybernetic systems.

2.1 Propagating Structure and Function

Considering an ensemble of natural and cybernetic systems one notices that they are in a modern geometric, topological and dynamical sense perturbed

versions of each other. It's clear that we can consider the following perturbations of natural and cybernetic systems:

- Perturbations of their encoded observables,
- Perturbations of their encoded developmental and evolutionary dynamics.

The first type of perturbation consists of non-integrable and integrable deformations of the frame fields, coframe fields, metrics and/or connections resulting in a change of the curvatures and even of the induced physical field equations. The non-integrable deformations due to noise (irreversible processes) and relative resolution differences in the encodings cause changes in (Equivalence 1), (Equivalence 2), (Equivalence 3), (Equivalence 4) and (Equivalence 5). The integrable deformations consistent with the postulated gauge group on the contrary do not cause curvature changes. However, in practice the encoding of natural and cybernetic systems are the result of a density field requiring the integrable deformations to belong to a certain class in order not after measurement to give rise to non-integrable deformations.

The second type of perturbations directly affects the natural and cybernetic systems as characterised by (Classification 1) and (Classification 2). Here also the perturbation can be such that the natural and cybernetic systems stay within the realm of a particular class of dynamics and that only a deformation of the constitutive parameters, α and β, is necessary to ensure that the systems are conjugate. However, in general a change in the cybernetic system dynamics cannot be captured in terms of an integrable deformation. In this case nonintegrable deformations or natural transformations (see Section 1.2.0) are necessary to relate the cybernetic systems.

In order to extract from natural and cybernetic systems a stable and reproducible set of equivalences and classifications despite these perturbations we proposed in the realm of computer vision [5, 12] a dynamic scale-space paradigm controlled by the equivalences and classifications themselves. Essential in the context of this paradigm is the derivation of a proper induced exchange principle for equivalences and classifications between among natural and cybernetic systems.

From an encoding of a natural system, (Definition 11.1), governed by developmental and evolutionary dynamics, (Law 1) and (Law 2), respectively, a stable and reproducible set of equivalences and classifications despite above types of perturbations can be retained by coupling the exchange principle intrinsically to the encoded natural system dynamics. Essential in this context is the assessment of the topological interactions activated among natural and cybernetic systems. These interactions can be quantified by these co-evolving systems system as topological currents.

For a complete and irreducible set of equivalences and classifications a committed ordering of the activated cybernetic system can be succinctly formulated through the use of a statistical partition function Z related to a free energy F for (Equivalence 1), (Equivalence 2), (Equivalence 3), (Equivalence 4) and (Equivalence 5), and (Classification 1) and (Classification 2).

EQUIVALENCE 6 *The statistical partition function Z related to free energy F for irreducible (Equivalence 1), (Equivalence 2), (Equivalence 3), (Equivalence 4) and (Equivalence 5), and (Classification 1) and (Classification 2) of the cybernetic system is defined by:*

$$Z = \sum_{V} \prod_{x,i} \exp\left[-F\left[V_i(x)\right]\right],$$

with

$$\begin{aligned} F\left[V_i(x)\right] &= -\log Z \\ &= \sum_{i,k,p} dv^p \left(\tilde{V}_{i;\pi_k(g_1 \ldots g_k)}(x, \tau_{i;\pi_k(g_1 \ldots g_k)})\right), \end{aligned}$$

where x labels any state, i.e. a space-time region with some dynamics, giving rise to equivalence or classification V, π_k a permutation of a sequence of $k \geq 0$ integers $(g_1 \ldots g_k)$ with $k = 0$ for labeling frame vector fields v_{g_k} and $\tau_{\alpha;\pi_k(g_1 \ldots g_k)}$ (inner) dynamic scales consistent with gauge group (Definition 11.2) and the equivalences and classification $\tilde{V}_{i;\pi_k(g_1 \ldots g_k)}$.

Again gauge invariance of the statistical partition function for the developmental and evolutionary dynamics holds.

THEOREM 11.36 *(Equivalence 6) is invariant under gauge group (Definition 11.2).*

PROOF 2 *Follows immediately from the gauge invariance of (Equivalence 1), (Equivalence 2), (Equivalence 3), (Equivalence 4) and (Equivalence 5), and (Classification 1 and (Classification 2).*

The partition function can be conceived as a measure of the topological, geometric and dynamical complexity of the developmental and evolutionary dynamics. The advantage of our measure is that it readily substantiates and extends information theoretical measures as proposed in computer vision [1]. Furthermore, the new measures of complexity are to be preferred for their conciseness. Moreover, the dynamic scale-space paradigm, therewith, falls nicely in the realm of modern theory of dynamical systems.

Besides the induced gauge invariant canonical parametrisation of space-time and dynamics also a topological interaction is needed to ensure a filtration yielding a hierarchy of partially equivalent ensembles of cybernetic systems that are slight perturbations of one another. This topological current is in our dynamic scale-space paradigm brought about by the statistical partition function (Equivalence 6). Studying two 'local' factors $Z(p_1)$ and $Z(p_2)$ in the statistical partition function going from state p_i to state p_j involves a factor $k(i,j)$ to generate $Z(p_j)$ from $Z(p_i)$, whereas going from p_j to p_i requires a factor $K(i,j)$ (assume $k \leq K$) to generate $Z(p_i)$ from $Z(p_j)$, such that $kK = 1$, i.e., the notable Artin-Whaples formula in disguise. Realising that the interaction

can only be defined through the interactions between two adjacent states, it is more than reasonable to let a topological current between states p_i and p_j to be controlled by the partition function Z^r for two-state interactions capturing all the possible couplings between the states of all pairs of states:

$$\begin{aligned} Z^r &= \prod_{i \neq j} Z^r_{ij} = \prod_{i \neq j} \left(\frac{K(i,j) + K^{-1}(i,j)}{2} \right) \\ &= \prod_{i \neq j} \cosh \left(F(p_i) - F(p_j) \right). \end{aligned}$$

Note that these interactions between two states do not exclude long range forcings: $Z(p_i)$ can instantaneously incorporate natural and cybernetic systems properties such as (Equivalence 5) and (Classification 1) and (Classification 2). The finiteness of the transport velocity need not to be in contradiction with the fact that the action is instantaneous and nonlocal!

With this two-state coupling partition function, Z^r, there's associated also a two-state coupling free energy F^r:

$$F^r = -\log Z^r.$$

Assuming the co-evolving natural and cybernetic systems to be a closed system for a particular region of space-time, then the change in the state of the co-evolving system can be realised by a change in the two-state coupling free energy F^r. Keeping in mind that the free energy, see (Equivalence 6), should be preserved, i.e., $dF(p_i) = -dF(p_j)$, this change in the two-state coupling free energy, $\mathrm{d}F^r$, is given by:

$$dF^r = -\sum_{i \neq j} \tanh \left(F(p_i) - F(p_j) \right) dF(p_i).$$

Thus the geometric or topological charges of the developmental and evolutionary dynamics have become the generators of an induced filtration of the dynamics of the co-evolving system. Now let us consider again the interaction mechanisms between pairs of states p_i and p_j, and define the topological current to be the curl of the induced connection on the two-state coupling free energy:

DEFINITION 11.37 *The topological current for the free energy on the activated co-evolving system is defined by:*

$$\begin{aligned} j^F &= \nabla^\Gamma \wedge dF^r \\ &= -\frac{\nabla^\Gamma_{v_s} F}{\cosh^2 \left(\sqrt{g(\nabla^\Gamma_{v_s} F, \nabla^\Gamma_{v_s} F)} \right)} dv^s \wedge dF, \end{aligned}$$

where v_s is connecting free energy states $F(p_i)$ and $F(p_j)$ of the co-evolving system.

Note that the topological current is steered by (Equivalence 1), (Equivalence 2), (Equivalence 3), (Equivalence 4) and (Equivalence 5), (Classification 1) and (Classification 2).

As the free energy (Equivalence 6) should be preserved the dynamic exchange principle for free energy is made manifest through a physical law involving the topological current (Definition 11.37):

LAW 3 *The dynamic exchange principle for free energy says that the change per unit scale τ in the free energy (Equivalence 6) in a region Ω of the cybernetic system is equal to the exchange of free energy F between this region and its surrounding across their (common) boundary $S = \partial\Omega$ quantised by topological current (Definition 11.37):*

$$\delta_\tau F = -j^F,$$

with suitable initial and boundary conditions.

THEOREM 11.38 *The dynamic exchange principle residing in (Law 3) is gauge invariant.*

PROOF 3 *Proof follows those of (Theorem 11.11) and (Theorem 11.36).*

This law is in perfect harmony with the second law of thermodynamics that states that the entropy of a closed co-evolving system with time is only increasing if elementary subsystems not all in their ground states are permitted to interact. Here the equivalences and classifications living on the cellular regions are recombined causing a substantial simplification of the natural and cybernetic systems.

2.2 Indicators of Sustainability

In the previous section we have presented the exchange principle co-evolving natural and cybernetic systems. The question arises how do we quantify and qualify the multi-scale characteristics of such co-evolving systems.

It is clear that for the multi-scale co-evolving natural and cybernetic systems again similar analysis methods are available as for a closed and open systems (see Section 1.2).

CLASSIFICATION 3 *The classification γ of a filtered co-evolving natural and cybernetic systems is given in terms of symmetries, curvatures and conservation laws underlying the filtered developmental and evolutionary dynamics in (Law 3).*

THEOREM 11.39 *(Classification 3) is gauge invariant under the symmetries of the filtered developmental and evolutionary dynamics in (Law 3).*

This classification may form the basis for a set ofindicators for spatio-temporally as well dynamically Lyapunov and structural stability. A subset may serve as sustainability measures for co-evolving natural and cybernetic systems.

2.3 Collective Intelligent Agents

Architectures for collective intelligent agents (CIA) can automate and sustain NASA (pre-) schemes within SIMS [13]. The agents in a CIA environment are best breed agents selected from possibly distributed CIA development and deployment platforms: they essentially behave like co-evolving natural and cybernetic systems. Both platforms and environments may communicate with physical actors being either individual humans or groups with their own NASA-SIMS capacities and capabilities. The (off-spring) actor software agents support the actors to interact, communicate and collaborate with each other in an ever-complex multimodal way. The development and deployment environment looks after embedding and embodiment of the diversification of the intelligence of NASA (pre-) schemes in SIMS and environment following the laws for propagating structure and function of section 2.1. Thereto, the actor agents use (not necessarily language determined) agent communication languages (ACLs) and negotiation strategies to set up interactions among humans and systems: they could talk physics. Furthermore, the CIA development and deployment platforms generate genetic algorithms (GAs) with sustainability measures for NASA and other self-organization (pre-) schemes as proposed in section 2.3. By registering and assessing fitness and utility of those (pre-) schemes during operations within the CIA environment reinforcement learning within SIMS can be effectuated. Here again reinforcemnt learning may follow similar dynamic exchange principles as in section 2.1. Note that NASA-ing in SIMS by means of ACLs, negotiation strategies, GAs with sustainability measures are most efficiently developed and deployed by means of computer algebra systems.

3. Conclusion

We have laid down a framework for modelling and analysing the sustainable development of co-evolving natural and cybernetic systems. The modelling consists of the encoding of observables and their evolutions. The observables are gauge invariant equivalences that are consistent with the developmental and evolutionary dynamics with respect to these observables. The classification of latter dynamics is of eminent importance in the derivation of those equivalences as well as their classification. In order ot retain stable measures of sustainability for the dynamics as a whole an adapted dynamic scale-space paradigm has been proposed.

It is obvious that our presented framework asks for a very sophisticated problem solving environment. First of all the developmental and evolutionary dynamics of the co-evolving natural and cybernetic systems together with a predictive and (reinforcement) learning model have to be put in place. In this context it is essential to let the problem solving environment to be too integral parts of the co-evolving and natural cybernetic systems.

References

[1] C. R. Rao, "Information and accuracy attainable in the estimation of statistical parameters," *Bull. Calcutta Math. Soc.,* Vol. 37, pp. 81–91, 1945.

[2] I. Prigogine,*Time, structure and fluctuations*,Nobel Lecture, 8 December, 1977.

[3] R. Rosen,*Anticipatory Systems,* Pergamon, New York, 1985.

[4] S. A. Kauffman,*The Origins of Order: Self-Organization and Selection in Evolution,* Oxford University Press, Oxford, UK, 1993.

[5] A. H. Salden, *Dynamic Scale-Space Paradigms,* Ph.D. thesis, Utrecht University, The Netherlands, 1996.

[6] A. H. Salden, B. M. ter Haar Romeny and M. A. Viergever,"Differential and integral geometry of linear scale spaces,"*Journal of Mathematical Imaging and Vision,* Vol. 9, pp. 5–27, 1998.

[7] S. N. Kalitzin, B. M. ter Haar Romeny, A. H. Salden, P. F. M. Nacken and M. A. Viergever, "Topological numbers and singularities in scalar images; scale-space evolution properties," *Journal of Mathematical Imaging and Vision,* Vol. 9, pp. 253-269, 1998.

[8] R. Penrose and S. Hameroff, "Quantum computation in brain microtubules? The Penrose-Hameroff Orch OR model of consciousness," *Philosophical Transactions Royal Society London (A),* Vol. 356, pp. 1869–1896, 1998.

[9] D. Wolpert and K. Tumer, "An Introduction to Collective Intelligence," In *Handbook of Agent Technology,* AAAI Press/MIT Press, 1999.

[10] A. H. Salden,"Modeling and Analysis of Sustainable Systems," European Research Consortium in Informatics and Mathematics (ERCIM), Paris, France, November 1999.

[11] A. Riegler, "The role of anticipation in cognition," In *Proceedings of the American Institute of Physics on Computing Anticipatory Systems,* Vol. 573, 2001, pp. 534–541, 2001.

[12] A. H. Salden, B. M. ter Haar Romeny and M. A. Viergever, "A Dynamic Scale-Space Paradigm," *Journal of Mathematical Imaging and Vision,* Vol. 15, pp. 127–168, 2001.

[13] A. H. Salden and J. de Heer, "Natural Anticipation and Selection of Attention within Sustainable Intelligent Multimodal Systems by Collective Intelligent Agents," In *Proceedings of the 8-th World Multi-Conference on Systemics, Cybernetics and Informatics (SCI 2004),* Orlando, Florida, USA, July 2004.

Index