Sensor fusion in smart camera networks for ambient intelligence

Citation for published version (APA):
Maatta, T. T. (2013). *Sensor fusion in smart camera networks for ambient intelligence*. Eindhoven: Technische Universiteit Eindhoven. https://doi.org/10.6100/IR755363

DOI:
10.6100/IR755363

Document status and date:
Published: 01/01/2013

Document Version:
Publisher’s PDF, also known as Version of Record (includes final page, issue and volume numbers)

Sensor Fusion in Smart Camera Networks for Ambient Intelligence

by Tommi Määttä

The work described in this thesis was carried out at the Philips Research Laboratories Eindhoven, Eindhoven University of Technology, and Stanford University in California. This research was commissioned and funded by Philips Research.

A catalogue record is available from the Eindhoven University of Technology Library.
ISBN: 978-90-386-3404-3
NUR: 959

Cover Design: Tommi Määttä
Reproduction: Ipskamp Drukkers, Enschede

Sensor Fusion in Smart Camera Networks for Ambient Intelligence

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de
Technische Universiteit Eindhoven, op gezag van de
rector magnificus, prof.dr.ir. C.J. van Duijn, voor een
commissie aangewezen door het College voor
Promoties in het openbaar te verdedigen
op maandag 01 juli 2013 om 14.00 uur

door

Tommi Tapio Määttä

geboren te Sotkamo, Finland

Dit proefschrift is goedgekeurd door de promotor:

prof.dr. H. Corporaal

Copromotoren:
dr.ir. A. Härmä
en
prof.dr.ir. H. Aghajan

Contents

II Framework and Experiments 61

Part I

Background

Chapter 1

Introduction - Visual Analysis

There are immense opportunities in both existing and emerging applications enabled by the use of vision networks. Many of the emerging smart environments incorporate unobtrusive vision networks as one of the sensing technologies. Wearable and planted sensors can provide more robust performance, but are hindered by their obtrusive nature in not allowing the user enough freedom or naturalness for activities of daily-life.

The vision that has been followed in the research depicted in this thesis has been defined as Ambient Intelligence. Ambient Intelligence solutions use embedded networks of different sensors to understand what individuals are doing daily, and based on the patterns found, they adaptively provide personalized support to these activities.

Monitoring people by vision networks has shown great potential. These networks have multiple cameras providing their observations of a scene. The scene can consist of vast areas due to having a distributed set of cameras, which from distance can unobtrusively provide their observations. Based on these observations the vision network can infer the appearance, location, activity or behavior of a person.

In building technology that is capable of observing people, many particular challenges are faced. People are a difficult category of deformable and adapting objects to detect, follow and describe. There is a multitude of appearances due to personal characteristics and clothing options. People perform even the same activities with various gestures and changes in pace. All these differences between people are affected by context such as the time of day and the surrounding environment.

Having the observations from many viewpoints creates a richness of data, but this richness alone does not guarantee improved accuracy for a vision network. It is the intelligent data fusion mechanisms that make the difference by increasing the certainty in the decisions made, and by making properties visible that otherwise would have not been noticed. Other issues include providing comparable, associated, and aligned data in a synchronous manner. Therefore, the design of a vision network that fully exploits the complementary nature of multi-sensor data by applying suitable fusion is a multi-faceted problem, which requires a systematic approach for streamlining the design process and for achieving best possible results.

In this thesis a vision fusion framework is proposed for systematically approaching the use of vision networks for providing user-centric services. The goal of the framework is to provide a systematic way for designing, building and using vision networks for monitoring people with great emphasis on the exploitation of fusion potential. Many aspects of data fusion within vision networks are discussed, and experimental results on fusion at image, feature, score and decision level are shown to highlight the most crucial dependences in fusion. Based on these findings and the approaches proven to be most efficient, the proposed vision fusion framework is showcased.

Section 1.1 starts this chapter by discussing the evolution of the technologies that have enabled the use of vision networks. The benefits and challenges of vision networks together with solutions provided by multi-sensor data fusion are discussed in sections $1.2-1.3$. This is followed by expanding on the aim of the vision networks for ambient intelligence solutions, with special attention to aspects related to monitoring people in sections $1.4 - 1.5$. The introduction concludes in section 1.6 by providing an overview of the thesis and its contributions. Throughout the thesis, the related work is discussed with each related subtopic.

1.1 Enabling Technologies

Four fields especially have made it possible for vision networks to exist, and much depends on the advances made in these fields on how and where vision networks will be deployed in the future.

- computing
- wireless networking
- imaging
- artificial intelligence

The progress witnessed in computing and wireless networking provide vision networks with the necessary resources for performing the required handling and transmission of data. Imaging and artificial intelligence supply vision networks with the necessary specialized tools for capturing the scene and understanding what has been captured.

Computing Alan Turing [1912-1954] is considered as the first person to introduce and define many of the concepts related to computer science and artificial intelligence. With the Turing machine in 1936 he provided the theoretical background for computer algorithms and computation. The first computers built were huge in size and weight, some of them weighing 50 tons, using 25 kW to perform only hundreds of instructions per second. Even these elementary machines proved their worth, first big success during WWII in breaking encrypted military communications. Several inventions and new technologies such as vacuum tubes, and later transistors with very-large-scale integration (VLSI) helped in creating smaller electronic parts, smaller computers that had commercial credibility by the 1950s. 1970s introduced us the microprocessor computers we know today. In 1993 Intel announced the first Pentium processor of their x86 architecture. This processor accompanied computers weighing only few kilograms, using only few tens of Watts for operation, but was putting out 60 million instructions per second.

Defined by Moore's law, we have been able to predict reasonably well the available computing power since 1965. The predictions have relied on the increased count of transistors that manufactures have been able to place on the same circuit area. Lately, engineers have introduced the use of more parallel computing solutions. However, new technologies are needed for sustainable development. New approaches, such as quantum qubit computing discussed by Deutsch in [1] and DNA-based computing originally presented by Adleman in [2], are under development. One widely promoted approach has been cloud computing [3], which helps in pushing the computation requirements to a specific remote site (servers) while maintaining the light and slim design of end-user devices.

The above described evolution of computing has made it possible for vision networks to rely on both heavy-load central video servers and light-load smart cameras, a type of cameras that have a low-level image processing unit on-board. In this thesis the real-time application experiments and multi-camera recordings would not have been possible without the use of modern CPUs and GPUs.

Wireless Networking The origins of wireless data transmissions can be traced back to Alexander Graham Bell. With a photophone device Bell was able to transmit a normal human conversation on a beam of light on April 1880 [4]. During the 1900s with the invention of radio, and later television broadcasts, antenna to antenna transmissions were vigorously developed. During the 1980s, fiber-optic communications came into widespread use through the invention of Internet. As complementary technology for covering difficult and complex areas Ethernet-based local area networks (LANs) were developed and deployed. Other small distance radio-channel based technologies such as Bluetooth were developed to satisfy transmission needs for energy-aware devices. Earth-orbiting satellites which by connecting to stations/devices on ground have created world-wide networks capable of relaying data across the globe.

We have reached a point in which there are various wireless technologies each suitable for a particular job, depending on the type of data and transmission distances. Considering vision networks, the most recent development has been the deployment of IP-cameras, each camera having its own unique IP-address for remotely accessing the video data. This has given unprecedented amount of freedom for deploying networks of multiple cameras, and thus enabling a completely new set of applications possible by the use of vision networks. The freedom in placement of personal computers and (IP)-cameras has been exploited throughout this thesis.

Imaging The most important technology for vision networks has been the evolution of imaging sensors, cameras for both visible and non-visible light. Use of black and white cameras can be traced back to 1903, when a 12 minutes long movie called The Great Train Robbery was filmed. By 1930s, color was introduced to capturing scenes in such classic movies as Gone With The Wind in 1939. During decades people worked on capturing more and better colors through new camera standards in film-making. American cinema introduced 3D films to its audience already in the 1950s, technology relying on stereo-cameras and wearable glasses for creating an immersive 3D effect.

Cameras can be divided into passive and active based on the imaging process. In a *passive camera*, lens captures and focuses the light emitted by the scene onto the internal image sensor formed by a pixel-grid. Light of the infra-red wave-lengths can be used to detect heat levels in a scene; excellent for detecting warm bodies in low-light situations. Cameras operating on visible light spectrum are nevertheless the most widely spread devices. *Active cameras* introduce to the scene some additional markers, usually by additional light source working jointly with the camera, when capturing the scene and the range. Range imaging techniques such as time-of-flight [5] and structured-light [6] have received wide interest due to improved capture of scene geometry and depth in real-time.

This thesis has focused on studying passive RGB color-cameras functioning on the visible light spectrum. Numerous types of passive color-cameras from cheap and mid-price off-the-shelve web-cameras to more professional IP-cameras have operated as visual sensors in the studied vision networks.

Artificial Intelligence Artificial Intelligence (AI) aims to create intelligent machines that are capable of perceiving, reasoning by processing data, and taking action to achieve success. AI-related research was originally funded by governmental institutions, such as U.S. Department of Defense. By 1980s simulated knowledge and analytical skills based expert systems received commercial success. Starting from the 1990s artificial intelligence became a crucial part of the modern technology industry, contributing heavily e.g. in logistics and data mining. Two especially important fields for research within AI have been *machine learning* and *machine perception.*

The aim of machine learning is to train a computer in such a manner that it can classify an observation to the correct category or class. The observations are given usually by sensors. Two of the most common machine learning methods are supervised and unsupervised learning. In unsupervised learning computer finds patterns in a stream of observations. In supervised learning the computer is provided with the correct class labels. Many different models to machine learning have been introduced, such as decision trees (DT), artificial neural networks (NN), support vector machines (SVM), and probabilistic graphical models (PGM). Each approach finds the most suitable model of its kind with regards to the analyzed statistics of the data. DT, NN and SVM based classifiers infer the class based on the current observation. PGM classifiers study how the feature values change within a time interval and based on this feature behavior assign the class label. Machine perception is the ability of a machine to observe environments and objects by sensors, such as cameras and microphones, and based on the observations make deductions on certain aspects of the observed item, Once analyzing visual observations, this ability is referred to as computer vision.

The research conducted in this thesis has applied various types of existing tools for modeling, depending on the nature of the observation data. The computer vision topics of object recognition and scene reconstruction have been of special interest. Within object recognition especially object detection, pose estimation, and facial recognition have been extremely useful.

1.2 Vision Sensor Networks

A multitude of aspects can be observed by a single camera. However, there are inherent limitations when relying on a single sensor for capturing a scene. A webcam e.g. works well in monitoring a person sitting on a chair working in front of a computer, but as soon as the user leaves this hot-spot, he is out of the coverage area of this otherwise perfectly functional camera. By having this drawback on coverage, the user's freedom is severely limited assuming he wants to be under continuous monitoring. Another major drawback of using a single camera is the difficulty, and occasionally impossibility, in dealing with occlusions and uncertainties.

Advantages
By deploying multiple cameras to observe the scene many of the drawbacks of single-camera vision can be avoided or at least treated. The *working range* of the vision network can accommodate a much wider area, by increasing the number of cameras. A user can be given more freedom for his movements. And a greater understanding of situations, such as proximity to others and social interaction, can be achieved.

The *effect of noise* sources or other non-interesting dynamic objects of the scene can be mitigated, by having the possibility for intersecting observations from multiple cameras, and for checking correspondences between the views.

Sometimes the scene might have an object, such as a opening and closing door, that interfers with the visual observations of a view. With additional cameras the *robustness against the interference* is increased, as some of the views are completely unaffected by the interference.

An important appearance or behavior property might only be *visible* from a specific viewpoint. If the vision network would not have a camera providing observations from that viewpoint, the occurrence of that important property would be missed. In general, the *resolution of a property is improved* as multiple estimates are used to refine the estimate with increasing accuracy.

Possibly the most crucial advantage comes in solving *uncertainty* of observations. A person can, e.g., be occluded by an object such as a table in one of the views, whereas another view might provide completely unobstructed view to the person. Therefore, a single camera can not provide its observations with high confidence, but as a whole, the network can build up a higher confidence by offering a joint estimate of the observed property.

Challenges
Deployment of multiple cameras gives rise to a set of practical challenges. A robust and fast system architecture has to be implemented in order to accommodate multiple cameras.

Various issues related to *data alignment* such as networking, camera calibration and synchronization have to be dealt with. The data traffic between cameras and a possible central unit has to be reliable and tractable. Cameras have to be calibrated both internally (for itself) and externally (with respect to other cameras) for some applications to have a common frame of reference. Most often it is desired for cameras to operate in the same frame-rate and to have no significant time-shifts during operation, for the observation samples to remain comparable and thus combinable.

Another set of issues related to *data association* arises as multiple individuals enter a scene covered by multiple cameras. In a vision network it is not anymore enough to track multiple persons within a single camera view, which is already a complex problem when the paths of inviduals cross and again separate. In addition to keeping track of persons within a view, a vision network has to also match all the detected people in all the views to each other. This is a complex problem, as people do not look the same from all viewpoints (e.g. multi-colored shirt), and under different lighting and camera settings, such as white-balance and gain, the appearance differences become even greater.

1.3 Multi-Sensor Data Fusion

The richness of observation data should improve the performance of the vision processing and therefore elicit more accurate system performance, but this is not guaranteed by the shear amount of additional observations. Catastrophic fusion [7] is a phenomenon in which the average accuracy of a multi-sensor system actually is worse than that of a single-sensor system. This can easily occur, but can be avoided by the design of smart data fusion mechanisms. These fusion mechanisms are the main focus of this research.

Fusion of Data
Multi-sensor data fusion, also referred to as sensor fusion, is part of a larger research field called information fusion. Information fusion aims to provide more certainty for the data. Uncertainty in data is decreased by combining the original data into a new set of data. In sensor fusion the original data is provided by sensors, or has been processed from the sensory data. By sensor fusion the resulting information is expected to be better for the later stages of the system, than would be possible by using each sensor data separately. With better data e.g. the classification results of a human posture analysis vision network should provide more accuracy, possibly with increased performance.

Developments in Sensor Fusion
The use of sensor networks was greatly inspired by the requirements of military surveillance systems in the 1970s. These multi-sensor systems, based on radars or sonars, were the first to introduce the terms and benefits of multi-sensor data integration by data fusion. The increased interest in distributed sensor systems was quickly met by research in 1980s conducted on statistical hypothesis testing [8] and optimum fusion structure for multi-sensor detection [9].

Further research to optimal data fusion strategies was performed in the 1990s with Aziz et al. [10] discovering the importance of the probability distribution of observations in addition to desired false alarm probability, when applying a k-out-of-n fusion rule. Multi-modal biometric systems received much interest in the start of 2000s. Jain et al. studied the effects of normalization techniques used for biometric traits in combination with fusion methods [11]. Dependencies were discovered, and they concluded that person-specific weighting of different biometric measures provides better performance than uniform inter-person weighting. Veeramachaneni et al. [12] highlighted the importance of another new aspect, the correlation between classifiers, and if modeled, how classification accuracy can be improved.

Fusion of Visual Data
The sensors used in vision networks, cameras, are based on capturing sequences of

images. Image data, and the measures and features one can compute from this data have special limitations.

The information derived from observations is always affected to some extent by the view and location of the cameras. How much these changes affect the image processing steps depends on how well e.g. the feature extraction considers changes to object scale and rotation. Therefore, processing needs to apply the same algorithms across cameras and algorithms need to provide as much view-independence as possible.

Many types of occlusions exist with vision networks, and these will affect the observations made. Self-occlusions are cases in which, e.g., a person's body may hide one of the person's arms behind it. Person-occlusions appear when other people walk between a person and the observing camera. Scene-occlusions exist when, e.g., an object occludes parts of a human body as the person moves behind the object.

The scene being observed can itself hinder image analysis efforts. The environment creates changes to the scene, e.g., by having changes in lighting, dynamic outdoor illumination and moving objects such as doors. If falsely detected, they will be included as a legitimate part of an observation, and possibly have serious consequences for scene analysis.

The vision fusion framework that is proposed in this dissertation takes these specific characteristics of visual data into account, and discusses topics which in general fusion dilemmas would have been overlooked.

1.4 Targeting Individuals

The research conducted focuses on user-centric applications for providing a meaningful service. The problems of detecting, tracking, and modeling of humans, their actions, and their behavior have been at the core of this research.

Person as an Object
From the point-of-view of low-level computer vision algorithms humans are a difficult group of objects to follow. Seen by the camera, a person is a deformable object. Even the same person takes many shapes as his direction and posture relative to the camera changes. In addition, a person's appearance can change as he wears different clothes through the day. These properties naturally vary also from person to person.

Person as an Agent
From the point-of-view of a knowledge base trying to model persons actions, things are no easier. People perform various actions in their own order and pace. The same person can do a similar task in various ways with minor changes in order and duration taken by sub-activities, such as opening a kitchen drawer when preparing a meal. Similarly, people have goals/intentions and there are many ways to succeed in reaching them, or in getting stuck. With statistical modeling approaches the aim is to count for all these variations and despite the differences to be able to detect the real underlying activity or goal.

1.5 Towards Ambient Intelligence

Purely from the application side the driving force for research and development was defined by various security and surveillance related services. For example, covering vast areas by monitoring people in order to detect abnormal behavior and misplaced luggage has been especially important in public spaces such as airports and shopping-malls. Many smaller environments, such as gas-stations, cafeterias and shops, have installed security cameras, for both monitoring the personnel and the public. Many health-care projects, such as elderly care [13] and hospital monitoring, are slowly discovering the benefits of vision networks.

As mentioned before, the benefits of vision networks have in the past been directly connected to public spaces and security-related monitoring applications. More and more personal services are appearing, such as in-car cameras for aiding in lane changing [14] and driver attention estimation [15].

It is considered very probable that vision networks will be a significant part also in private spaces, such as homes, for providing pleasure and convenience to users. In this thesis some new applications for vision networks within both private and public venues are presented. A home-to-home visual communication application is presented in chapter 6, and an extension of it to home-activity recognition is presented in chapter 7. Chapter 8 discusses detection of and response to repetitive behavior within shops, and in chapter 9 an application concept for providing personalized office ergonomics is presented.

1.6 Thesis Overview and Contributions

A look into the technological, human and application side of this research was provided in the previous sections. Vision networks are at the intersection of these three aspects with multi-camera data fusion being the key factor.

> *Having the technology to build vision networks capable of human behavior interpretation, the question becomes how to systematically across various applications design vision systems in a modular manner while integrating intelligent fusion approaches to the system design? What are the common approaches and their consequences to fusion potential and exploitation?*

This thesis aims to provide the necessary theory and systematical tools for tackling the above problem. This thesis consists of two parts. The first part provides the necessary theoretical background for following the experimental work presented in the second part.

Outline of Part I

The first part of the thesis starts by providing the necessary background on and theory of vision networks and sensor fusion. With each discussed topic further definitions and refinements are proposed in order to tailor them specifically for the fusion-friendly vision systems.

Chapter 2
This chapter gives an overview of common vision networks by first motivating their use through the many applications only they can enable. The chapter explains the ambient intelligence vision behind most of the experimental work performed in this thesis, and provides a look into emerging applications of vision networks. It elaborates on the vision network configurations, followed by an application-driven look into the common three-tier architecture (VNA).

Chapter 3
In this chapter an overview is given on sensor fusion concepts, on the type of data to be fused within a vision network, and on the fusion methods to do so. Inspired by known structures, such as classical three-level and Dasarathy categorizations, a further categorization is proposed for fusion processes in the context of vision networks. The corresponding levels of fusion are discussed through examples. The chapter concludes by gathering fusion methods in a new category-structure influenced by the work previously presented by Elmenreich.

Chapter 4
In this chapter the discussion on fusion expands to sensor fusion architectures and architecture models. A three-category architecture structure is proposed based on two previously presented categorizations. After considering synchronization issues, several application-driven architecture models are presented and discussed for their suitability for vision networks.

Outline of Part II

The second part of the thesis presents the vision fusion framework by illustrating the structure, the connections within and the main aspects of each part through a set of design rules. The proposed fusion framework is followed by four chapters each covering a multitude of fusion experiments for different applications. The second part concludes the thesis with a general discussion on the presented material.

Chapter 5
The proposed vision fusion framework (VFF) for designing and building fusion-friendly vision networks is presented. The structure and the connections of VFF are illustrated, by expanding on how it connects to VNA and to the three major aspects of fusion: architecture, level and method. Fusion architecture describes the structure in which multi-sensor data is gathered in. Fusion level defines the type of data: such as image and feature, that is been gathered. Fusion method defines the manner in which the gathered data is combined into a single estimate. A set of design rules for the VFF is formulated.

Chapter 6
This chapter presents an application for visual communication and awareness between remote sites. The application is provided with a 3D shape reconstruction, on which two fusion experiments are conducted. The first experiment studies the effects of camera configuration to shape estimation. The second experiment examines the accuracy of five proposed occupancy fusion methods over imaging sensor and camera calibration noise.

Chapter 7
This chapter presents an application to recognizing six basic human activities. The application is studied with three different silhouette datasets by relying on a tree-structured classifier. Five different camera fusion scenarios are proposed and their accuracy is studied over all the datasets.

Chapter 8
This chapter presents a methodology called Action History Matrices (AHM) to classify and detect behavior repetitions on various scales and time spans. The properties of AHM for detecting repetitive behavior are demonstrated in analyzing customer behavior in a smart shop environment. Based on the results, implications to fusion approaches are discussed.

Chapter 9
This chapter presents an application for office ergonomics in increasing self-awareness of workers. The application is studied with a recorded eight-camera dataset from an actual office. A system for providing information for personal ergonomics and general mobility is defined, data analysis results for personal ergonomics are shown, and experiments with various fusion approaches are conducted for the study of general mobility within the office.

Chapter 10
The last chapter concludes the thesis by starting with a review on the major aspects of vision networks and the future applications only they can enable. Finally, the conclusions are given on the suitability of the proposed vision fusion framework, for fully exploiting the potential of sensor fusion within vision networks.

Chapter 2

Vision Networks

A vision network is a system that by capturing images, processing them and modeling properties of the objects of interest aims to understand the scene. In this context, understanding usually results in a decision given out by the system. Understanding can be based on modeling geometry, shape, movement, and appearance. In essence, we are trying to mimic the capabilities of human vision in perceiving the world; a skill that we humans develop continuously over time from the moment of birth given our unique senses. This is a very challenging and an incredibly large problem area to tackle, given the limitations of electronic sensors and limited time-spans and datasets for teaching vision systems.

Various different application fields have benefited from deploying vision networks. Application areas such as medical image processing, machine vision, military support and autonomous vehicles are known to benefit from the extended scene understanding a vision network can provide.

Section 2.1 starts the chapter by providing a look into existing and emerging applications enabled by fusion powered vision networks. The chapter continues in section 2.2 by categorizing vision networks based on four aspects, each of which can affect vision efforts significantly. A common vision network architecture containing three system layers is presented in section 2.3 together with common error sources. Section 2.4 provides remarks concluding the chapter.

2.1 Motivation

Various different application fields have benefited from deploying computer vision techniques. Vision networks have enabled the use of these techniques to problems covering wide and complex areas. The traditional application domains have been heavily shaped by security and surveillance related research. Whereas, the emerging field of ambient intelligence solutions aims for more private spaces with emphasis on health and well-being through user support. It is important to be aware of and compensate for the limitations of visual sensors, to which end some complementary sensor technologies have been applied across application domains.

2.1.1 Traditional Applications

Applications for providing security and awareness are some of the traditional uses for vision networks, see below a more comprehensive list:

- *military*: detect soldiers, vehicles, and targets for missiles
- *surveillance*: detect suspicious people and uncommon events
- *industry*: detect product quality and guide automated processes
- *video conferencing*: enable clear communication between multiple sites and persons
- *autonomous vehicles*: provide driver support through awareness of surroundings

Military

Originally, military applications for computer vision were limited to the detection of enemy soldiers and vehicles, shortly followed by missile guidance systems. Much advancement has occurred in targeted missile guidance, modern missiles selecting the most appropriate target only once entering the predefined target area. Current efforts are in providing awareness of the battlefield, by capturing the combat scene by a multitude of sensors, even in multiple modalities such as video, audio and pressure. The remotely acquired awareness can be exploited well in time before sending in the troops, decreasing the loss of lives.

Surveillance

Having a complex, wide area covered with multiple sensors makes the task of detecting suspicious behavior very difficult for a human operator. Automation of selecting persons or events of interest greatly reduces the workload and eventually increases the accuracy of the operator. Having less data to process, the operator can, e.g., focus his attention to the screens that provide a view to the suspicious person.

Industry

The term of machine vision is applied to cases in which computer vision is used in industry. Machine vision aims to help in manufacturing and delivery processes by providing tools for, e.g., quality control and robot arm navigation. Quality control is applied to maintain the quality of products, e.g., in terms of automatically removing faulty or undesired products from the manufacturing line.

Robot arm navigation is automatically supported by providing the arm control unit with the necessary measurements on position and orientation of the product. Machine vision can be considered to aim for reducing manufacturing costs, improving quality of products, and reducing human-related accidents in hazardous environments or when handling hazardous materials.

Video Conferencing

As business has become more global, requiring remote sites to stay in close contact, a affordable approach to information exchange has been given in multi-person, multi-site video conferencing applications. For providing functionality that provides information without occlusions in a fast-paced environment, the solutions exploit camera networks in providing unobstructed views and multiple microphones in capturing the audio and speech. By having a clear virtual representation of the remote sites without unwanted artefacts, the communications can be carried out in a natural, efficient manner.

Autonomous vehicles

Major discoveries and new challenges for vision networks have been created by the increasing interest in autonomous vehicles and systems supporting the operator of the vehicle. A range of vehicles from in-the-water, on-the-land, and up-the-sky have all been subject to research and development. Various levels of automation have been deployed, such as giving the vehicle operator automatic support in avoiding collisions with other vehicles and pedestrians.

In fully autonomous vehicles, in addition to supporting functions, also the navigation and obstacle avoidance is left for the vision network. Fully autonomous missiles, UAVs (unmanned aerial vehicles), and space explorers, such as Mars Exploration Rover from NASA, have been successfully deployed. Many car manufactures have financed research for autonomous cars, but these vehicles still exist only as experimental versions. Nevertheless, great progress has been witnessed as the competing teams taking part in the DARPA Grand Challenges of 2004, 2005 and 2007 have successfully created driverless vehicles for the public roads.

2.1.2 Applications for Ambient Intelligence

Ambient Intelligence (AmI) serves as the vision for the future smart environments including consumer electronics, telecommunications and computing [16, 17]. AmI envisions electronic environments that are sensitive and responsive to the presence of people. Small, integrated and connected devices unobtrusively work together in supporting people in daily activities and emergency situations. Ambient Intelligence can be characterized by systems that have the following properties [18]:

- *Embedded*: Networked devices are integrated into the environment.
- *Context aware*: Can recognize the user and their situation.
- *Personalized*: Can tailor itself to meet the users needs.

- *Adaptive*: Can change in response to the user.
- *Anticipatory*: Anticipates your desires without conscious mediation.

As the vision states, the solutions AmI offers are applicable to many problems. An workstation allocation for an incoming laboratory user based on ongoing CPU, disk and network load was presented by Marchesotti et al. [19]. The deployed ambient intelligence system was connected to various sensor types: mouse/keyboard activity sensors, network adapter sensors and hard disk usage meters just to name a few. The system efficiently directed the incoming user to a workstation with a large amount of free resources. A similar guidance of users was demonstrated by Calbi et al. [20] with the users carrying a mobile-device that prompted messages guiding the user to his selected target by adapting to the users movements based on combined video-radio localization.

A smart home application that monitors an elderly person with a camera network and a wearable accelerometer is one of the most appealing uses of ambient intelligence solutions. One such system proposed by Tabar et al. [21] aims to detect troubling events and automatically alert the caretakers in case of a fall.

The applications studied in this thesis all follow the vision of AmI. The proposed vision networks provide AmI the tools to help this vision come true. It is good to notice that robustness against the variability that exists in real world scenarios can be best dealt with using complementary sources of information.

2.1.3 Complementary Sensor Technologies

It is worth noticing that most of the sensor networks applied to real world scenarios do not consist only of cameras, due to their inherent limitations in making observations and lack in algorithmic robustness. Therefore, the majority of the applications are provided with a multi-modal sensor network with additional contextual information. These systems are capable of robust reasoning with vision networks as one of the technologies for making observations.

Contextual Information

Zhang et al. proposed a multi-camera system for person tracking that was calibrated w.r.t. a real world map (outdoor) or a blueprint (indoor) [22]. The system described was applied to two different problems: critical infrastructure protection and hazardous lab safety verification. A military airfield was covered by 48 fixed and 16 PTZ (pan/tilt/zoom) cameras for making alerts on any intrusions to the protected space that was manually defined for all cameras by marking it on the global map. The alerts were localized based on the camera with the best view on the violator and the location. The lab safety problem was to ensure that the users adhere to the infectious disease laboratories two-person rule, which states that no person can alone work in the lab. The cameras were calibrated to the lab blueprint, which made it possible to fuse detections of human targets by single cameras into a joint detection when cameras so agreeing. The system kept count of people in the lab and alerted if it detected that a single person had been there alone for longer than 30 seconds.

Heterogeneous Cameras

The majority of the video-surveillance solutions exploit heterogeneous cameras through image fusion. The cameras capture the same scene in different light wavelengths such as visual, infrared and thermal regions. This versatility in cameras helps the system to cope with the changes in illumination, separation of living and inanimate moving objects and other adaptive scene conditions. The fusion of color and thermal images for achieving more accurate human silhouette extraction was studied by Han and Bhanu [23]. Both image types were used to compute preliminary silhouettes which were used in image registration of the two video streams. The registered candidate silhouettes are combined into an improved silhouette by probabilistic means. This approach showed clear improvements in silhouette extraction compared to approaches not exploiting fusion between the color and thermal images.

Wearable Sensors

A more robust approach to surveillance has been to introduce additional sensors. In addition to following individuals with cameras they can be made to carry RFID-tags which greatly reduce the possibility for severe false detections or misses. Zhai et al. considered the prevention of retail-loss [24], in which some items bought are not being scanned properly by the cashier in order to hand them to the buyer for free. Multiple cameras were used to observe motion in barcode scanner-region and the bagging/belt region. If the observations did not match with the actual barcode input, a fake scan alert was raised based on these primitive events. Zhai et al. also considered tailgating detection, in which multiple persons are entering a secured area with only one valid badge being scanned. If the number of people reported by the badge-reader and the two cameras (frontal and side) do not match, an alert for the operator or the security personnel can be automatically raised.

Other Signal Modalities

Considering human behavior, audio can provide additional clues that can help in disambiguating visually similar patterns of activities. Gatica-Perez et al. [25] and Talantzis [26] proposed a audio-video based video-conferencing system for tracking the positions and speaking activities of multiple persons within a meeting room. The system proposed by Gatica-Perez captured the scene by a microphone array and a set of uncalibrated cameras, whereas Talantzis had the cameras calibrated. Talantzis combines data by merging tracking information from both modalities based on smallest Euclidian distance between audio and video estimates. Whereas Gatica-Perez combine the multi-modal data through a novel Markov-based observation model.

Gandhi et al. [27] investigated configurations for binocular camera systems for automotive applications in order to increase safety in traffic. The camera systems were there to help the driver in detecting other vehicles and pedestrians. The stereo matching based approach to detecting objects in traffic showed promise. Automotive support systems have also included infrared cameras [28], laser scanners [29] and radars [30] to increase the robustness of scene understanding.

2.2 Network Categorization

Vision networks can be categorized based on four practical system aspects. Each aspect has a profound effect on the development of algorithms that can serve the application requirements.

- *coverage*: how much the views of cameras overlap
- *placement*: what is the role of a camera w.r.t. a person
- *calibration*: how much cameras know of themselves and other cameras
- *collaboration*: how camera observations relate to each other

2.2.1 Coverage

Vision networks can be categorized in terms of the coverage of the region-of-interest (ROI) between the cameras:

- no view-overlap
- partly view-overlap
- full view-overlap

Systems in which the views of the cameras do not intersect in any parts of the scene observed are called non-overlapping networks. Traffic cameras in highway areas are a good example of such a system. The working range is therefore large, but assigning more confidence to some individual observations will be impossible, as no additional scene information is available from other sensors.

Partly overlapping networks are the most common ones, existing in various public spaces such as shopping malls and airports. The working range remains relatively large and occasionally more confidence can be had through complementary observations.

Fully overlapping networks can be considered to exist only when a certain volume-of-interest (VOI) has been defined in the scene, and all the cameras have the VOI fully in their field-of-view (FOV). This situation easily arises in dedicated spaces such as communication environments. The potential for fusion is great in dealing with uncertainties and visibility issues, but the range of such a system is very limited.

The research depicted in this thesis starts with the exploration of VOI-centered fully-overlapping vision networks, as presented in the first application study in chapter 6. For achieving both maximum user freedom and closer to real life implementation, the fully overlapping networks were exchanged to networks that exhibited both partly overlapping and non-overlapping ROIs within the entire VOI. In practice, the later application studies are extended from a single room scenario to cover scenarios involving multiple rooms and complex-shaped areas.

2.2.2 Placement

Vision networks can be also studied based on the placement of cameras within the VOI. In this thesis three roles of a camera have been defined:

- *terminal*: covers a very limited region

- *dedicated*: covers a larger, semantically important region
- *ambient*: covers an entire space with no particular focus

A camera might be fixed to observe only a certain area or employee. These cameras are usually called terminal specific cameras which provide high quality coverage, but do this within a very limited area. By increasing the distance from the observed area/person to include bigger regions in the FOV, such as the cubicle of an office-worker, we are exploiting so called dedicated cameras. Dedicated cameras will usually provide less observation detail, but will be able to capture broader movements and possibly longer term behavior.

A third role of a camera is that of an ambient view. An ambient camera has no specific object, area, or person it focuses on. Its job is to provide a general overlook of the observed scene. The quality of detail is thus expected to be low when compared to terminal and dedicated cameras, mostly due to the varying distances and orientations the person is captured in. A major benefit of the ambient cameras is the opportunity to observe uninterruptedly the actions of a person within the scene, as only by leaving the environment the person will be out of reach of the ambient cameras.

Research depicted in this thesis starts with the exploitation of dedicated cameras. As the observed environments grew bigger, ambient cameras were included to provide the applications uninterrupted observations. The last application study in chapter 9 was designed to explicitly exploit the hierarchical nature of all three camera categories.

2.2.3 Calibration

A third significant characteristic of a vision network is the manner in which the cameras are calibrated. Calibration defines the sense-of-self and sense-of-others a camera has. To this end, there are commonly considered to be two calibration processes:

- *intrinsic calibration*: how image gets distorted by a camera
- *extrinsic calibration*: how cameras are situated w.r.t. each other

Sense-of-self is achieved by intrinsic calibration, in which a camera is analyzed for internal imperfections in the imaging process. Imperfections include geometric distortions which are visible when straight lines in scene become bended lines on the image. By knowing the imperfections they can be corrected by image manipulation. Sense-of-others is achieved by extrinsic calibration, in which the camera gains understanding of its placement and direction with regards to other cameras and the scene.

Intrinsic calibration is independent of external factors. Therefore, once performed for a camera, the same corrections can be applied regardless of the application and environment later involved. Whereas extrinsic calibration is performed every time a change in any of the camera properties in the network has occurred. If the position or orientation of a camera changes, the vision network has to be extrinsically calibrated once again.

There exist many tools for performing extrinsic calibration successfully such as the automatic one proposed by Svoboda et al. [31]. As the workload in capturing the required calibration images and computing the view-correspondences is significant, these tools try to make the process less laborious. From experience it can be noted that the automatic process saves much time, but the robustness of the calibration results is not guaranteed by the toolboxes themselves.

The calibration operator, such as the man moving around with a bright led light-source, has to make sure he provides close to optimal conditions, such as speed, location and visibility of the bright calibration point, for achieving an accurate calibration of multiple cameras. The fragility of the extrinsic calibration is therefore considered in this thesis as one of the major reasons for not opting for point-to-point calibration accuracy, but rather relaxing calibration to grids and areas-of-interest, or in some cases omitting extrinsic calibration all together.

Research depicted in this thesis started by relying on fully calibrated vision networks. Calibration was done both manually and automatically. With manual labor the results of the calibration were adequate and guaranteed to be usable. With automatic toolboxes the actual workload was less, but part of the process of calibration became less transparent. Therefore, in cases of failure the automatic calibration had to be fully performed again, till usable accuracy was achieved.

In overall, the extrinsic calibration of multiple cameras is fragile. If camera location or orientation changes, the extrinsic calibration needs to be performed again. With manual approach one might be able to compensate for these changes per individual camera, but with the automatic tools the consequence is usually having to acquire a new set of calibration data for all cameras. Therefore, the amount of time needed for performing calibration, validating it, and maintaining it can easily increase to unmanageable levels. Because of the fragile nature and easily increasing workload, an extrinsically calibrated vision network was only exploited in the first application study in chapter 6.

2.2.4 Collaboration

Considering the collaboration between the cameras of a vision network, three different categories have been presented [7, 32] based on sensor configuration:

- *complementary*: for more complete understanding of the scene
- *competitive*: for more certainty in the same property
- *cooperative*: for attaining property otherwise not accessible

A complementary network has sensors that can be used to combine information for achieving a more complete understanding of the observed phenomenon. Sensors do not directly depend on each other, thus they can operate individually. In a competitive network the sensors independently provide their observation of the same property. By competitive fusion the uncertainties can be dealt with, and the effect of erroneous observations can be cancelled. In a cooperative network as a result of combining the observations from multiple independent sensors, information can be derived that otherwise would not be accessible.

The majority of the application studies conducted for this thesis focused on cameras that provided their observation of the same property. Therefore, competitive vision networks were widely used. The last application study for personal ergonomics in chapter 9 exploited also observations regarding different aspects of a working style given by cameras at a different level of camera hierarchy. Thus by aiming also for a more complete understanding, the system exploited both cooperative and complementary fusion.

2.3 Three-Level Architecture

All the previously mentioned application fields and categorizations of vision networks have been crucial for the development of the architecture and algorithms of the modern vision systems. The next sections will introduce the common vision network modules, illustrate the structure, and highlight the challenging areas in which fusion of the multi-camera data can assist the vision system. See Figure 2.1 for an illustration of the proposed three-level structure of a common vision network architecture (VNA).

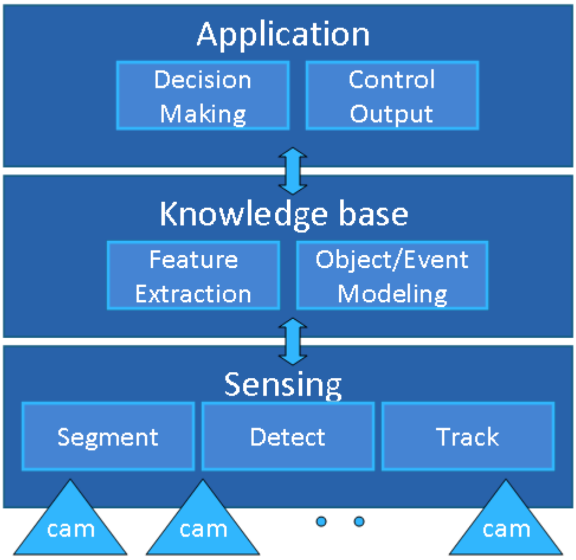


Figure 2.1: *The common three-level structure of a Vision Network. Multiple cameras provide observations, which are processed from raw data into attributes that are used to infer a decision that optionally affects the scene through control.*

2.3.1 Visual Sensing and Segmentation

The most rudimentary way to categorize a vision system is to consider it at three levels. The lowest level is considered as sensing and segmentation. This level includes the sensors and the low-level image processing tasks.

The sensors in a vision network are cameras that produce a 2-dimensional image based on light intensity (gray and heat images), or several spectrum bands of light (color images). The process of capturing an image on a light-sensitive sensor is called *image acquisition*, which is the first requirement for making observations of a scene.

Depending on the tools available and the application, some preprocessing steps might be necessary. Dealing with algorithms with low flexibility the resolution and frame-rate might have been fixed during the implementation stage. These implementation details need to be matched between cameras. Depending on the image sensor sensitivity for noise and bad contrast between objects and background, noise reduction and contrast enhancements can be applied.

The above mentioned preprocessing steps handle entire images. Eventually, at some part of the processing chain, the focus of the vision algorithms needs to be directed to specific areas and objects within images. For this purpose *image segmentation* can provide much help, by excluding areas of the image with nothing to give for higher level understanding of the scene. Excluded areas and objects are referred to as background, whereas areas and objects of interest are referred to as foreground.

Instead of segmenting foreground from the background, vision algorithms can directly approach objects by *detection*, like detection based on predefined types of movement, shape and appearance. Detection is usually a computationally heavy process, thus it is often preceded by simple image segmentation. Segmentation removes areas that otherwise would consume computation power due to running the detector across the entire image.

2.3.2 Knowledge base

The middle level we call the knowledge base, because at this level the data from the sensing level becomes information. Information involves the attributes of the area or object of interest. The attributes can be used in describing properties and further in modeling the behavior of the properties. The transformation of data into knowledge is achieved by *two processes*: feature extraction and object modeling.

The level of detail and the amount of data in a 2D image can be great. In an image sequence the detail is multiplied by the number of images the system is capturing every second, resulting in a large amount of data. In addition images themselves are a rather poor representation of objects. For providing more accurate and generic representations it is a common practice to reduce image data into feature data, sometimes referred to as measurements.

Reduction of data is achieved by *feature extraction*. Features can be very simple, such as lines and edges on an image. More localized features may focus e.g. on corners and blobs within a region-of-interest (ROI) of an image. Complex features are commonly used in describing object related properties, such as texture and shape for object appearance, and motion for object state. Whatever the features, the choice for them is driven by the requirements from object modeling.

Object modeling can be considered as creating a model of certain fixed structure, for describing the characteristics for an object class. The characteristics are usually either related to the appearance or the activities of the object. The model is defined in such a way that it can be used to test if an observation of a random object belongs to an object of that class.

Model training is a data-driven approach for defining the model structure and the expected values of the characteristics for the specified class of objects. When considering vehicles in traffic, examples on object classes may include pedestrians, bicycles, cars, buses and trucks. If modeling vehicles, both appearance, such as size and shape, and motion, such as average speed and movement direction, would provide discriminative features for separating between object classes.

2.3.3 Applications and Services

The third level in a computer vision system involves applications and services. Generally speaking, system modules that are used to build each of the discussed levels, are highly dependent on the application, and vice versa. Given limitations by the low-level vision processing tools, only those services can be included in the application that can be provided with sufficient level of data and knowledge. If no limitations exist within low-level tools, the application can be determined to provide both basic and advanced services. However, services that appear feasible in the planning stage of a system, can turn out too ambitious once faced with realities of the uncertainties involved in computer vision tasks.

Services are adaptively provided based on the computed decisions. Naturally decisions can be of various types, such as pass-or-fail decision for automatic inspection in quality control, or a recognition decision of a persons identity, or an alert decision to other systems involved. Decision, as a result of the scene analysis efforts, can be used as input for control modules. Control modules adapt environment properties based on the decision, and possibly external input from a user or a domain expert.

2.3.4 Common Error Sources

Each of the vision and modeling related challenges is capable of causing serious loss in the accuracy of decision making of the vision network. Naturally, any error will decrease the accuracy of the module it was created in. However, the effect of any error is not limited to the source-module. It will propagate to other modules of the same level, and further propagate the effect to following system levels. Some algorithms can correct some of the errors made by the preceding processing steps, but the most powerful tools for dealing with error mitigation are provided by sensor fusion.

Sensing and Segmentation
Low-level image processing tasks of sensing and segmentation system-level will have to include person detection and tracking tools. Detection is done for finding the object on the image plane that resembles a person. Detection is usually performed either by foreground segmentation or template matching. Tracking helps in sustaining the detection of a person as time passes and situations change.

Foreground segmentation separates the regions of the image that have changed from the regions that have not. Change is most commonly detected as a changing color of a pixel. Given large enough deviation of the color from the previously learned color of the pixel, the pixel can be marked as a pixel belonging to the foreground. The foreground is thus usually introduced by objects that were not in the scene when training the model for the background took place. The objects detected as foreground might not actually be new objects to the scene but already existing objects having dynamic behavior, such as leaves on the tree moving because of the wind, or a door pushed open by a person entering a room. Thus segmentation alone is not enough for robust detection of individuals.

Template matching provides a more direct way of finding persons [33]. A template, such as a smaller 2D image of the contours of an object, is overlapped by the observed image in various positions and scales. Matching computes the similarity of an image region to the template e.g. by counting the occurrences of similar trends or by computing a cross-correlation. Based on a hit threshold, the objects that provide a higher match with the template than the threshold are marked as detected objects. Therefore finding a person is not anymore influenced by the scene dynamics, unlike with image segmentation. However, having eluded this problem there will be various shapes that partly will match the template, giving rise to false hits. The usual approach with template matching is to find a suitable trade-off with false hits and missed hits, hits that were not detected for an object of the right type. The number of positions and scales for the template are the two most important parameters in defining this trade-off.

As has been discussed, both of the detection tools have their drawbacks, but these drawbacks are of different nature. Because of this fact, it is possible to build a hybrid method that will effectively cancel the drawbacks and exploit the benefits of both approaches. With image segmentation we can detect all the areas where a new person can exist. For all these candidate areas we can run the template matching with full parameter range, and select the areas/objects that are most like a person. For sure, in some situations there will exist false hits and misses, but the rate of these erroneous events will be highly decreased.

Tracking a person has two major challenges: crossing paths and occlusions. Based on a detection from a previous frame it is relatively straightforward to use that detection as origin for the search of a person in the current frame. This approach is called tracking-by-detection [34], in which previous hits are mapped to new hits based on their corresponding distances on the image plane. However, given rise to the two challenges mentioned, more intelligent tracking approaches are needed for solving ambiguities. These approaches could include the modeling of a person's appearance, such as color, texture and shape properties, which in case of crossing paths of two persons would after object merger help in identifying which person is which. For occlusions some restrictions for the change in the size of the detected body could work to alert the vision system. The system could apply anthropometric measures in forcing correct detection, e.g., in case of person walking behind a table thus their legs getting hidden behind the static object.

Feature Extraction

After a track on a person has been established, it is possible to describe the important properties of the individual. Feature extraction computes measures that are considered important related to properties-of-interest. For example, features such as object aspect ratio, and principal angle of silhouette pixels are good indicators for human posture.

The accuracy of feature extraction can decrease due to following factors: badly centered detection, person occlusion, people overlap, and bad camera viewpoint. If sensing has not been able to keep the track centered on the person, this might cause missed or additional foreground areas to be included in the feature computation. Similarly, an area of a person might be occluded by a static object or another person, causing similar deviation to feature computation. Most common difficulty in vision networks rises from situations in which the person is poorly situated with regards to an observing camera, such as facing away from a camera. Given poor viewpoint, the feature computation can be affected negatively due to loosing detail otherwise visible, such as not been able to observe forward extended arms from a frontal view.

Modeling of Property
Three challenges in making a decision on the type of a property should be pointed out: limited training data, dissimilar training data, and power of discrimination by the features. If not having enough feature-data of each of the classes, the model training will result in a limited model that is unable to capture the subtleties of feature distributions. If again training data is very dissimilar to the data gathered from the current environment, testing data, the feature subspaces for each of the classes will not match between the training and the testing data. Given a poor choice on features, the class subspaces might be overlapping to a large extent, causing difficulties for discriminating between the different classes.

2.4 Remarks

A categorization of vision networks based on various factors was presented in hope to illuminate the main aspects involved in utilizing vision networks. The common vision network architecture was illustrated, and the purpose and error sources of each of the three architecture levels were discussed. Due to the chain-like structure of vision networks, the errors not only affect the module in which they occur, but the effects also propagate to the final decision and the following control output. The effect of an error therefore spreads through the network, decreasing the achievable accuracy.

In the experimental chapters of this thesis we quantitatively study how data fusion can help in removing or at least in mitigating the errors evident in creating estimates of features, class probabilities and class labels. With intelligent fusion of multi-sensory data the estimates given by a specific network level can be controlled for outliers, general data trends, and most suitable or important aspects of data behavior. Consider a process that aims to detect the maximum of movement, while not being affected by outliers. Outliers include such phenomenon as other people moving in the background within the bounding box that contains the person-of-interest (POI). If such an outlier of movement is not ignored, the estimate of maximum movement exhibited by the POI will be inaccurate. The most effective procedures for providing sound estimates for vision networks, given multi-camera observations, are mechanisms of sensor fusion, which will be discussed in the next two chapters.

Chapter 3

Fusion Levels and Methods for Vision Networks

The richness of data available in a vision network offers an opportunity for increased decision accuracy. Richness in having multiple estimates for the same property does not by itself guarantee success. The improvements in system accuracy are made possible by intelligent data fusion mechanisms. These mechanisms provide an estimate of the multi-sensory data that makes vision network processes capable of providing good, stable accuracy. In this chapter an overview is given on principles behind data fusion, on the type of data to be fused within a vision network as the fusion level, and on the fusion methods to do so.

Section 3.1 starts the chapter by presenting some known definitions for data fusion, based on which a vision network specific definition is proposed. In section 3.2 the classical three-level and Dasarathy categorizations [35] for the fusion level are described, and a further categorization is proposed in the context of vision networks. In accordance to the data-type based categorization all levels of fusion are discussed through examples. Section 3.3 proposes a new categorization for fusion methods, inspired by the method categories presented by Elmenreich [36]. Section 3.4 provides remarks concluding the chapter.

3.1 Process of Fusion

Data fusion, or information fusion, has been given many different definitions by a variety of research fields. Some definitions related to fusion are given below.

> ***Data fusion*** *is a process dealing with the association, correlation, and combination of data and information from single and multiple sources to achieve refined position and identity estimates, and complete and timely assessments of situations and threats, and their refinements of its estimates and assessments, and the evaluation of the need for additional sources, or modification of the process itself, to achieve improved results. (JDL, 1987) [37]*
>
> *The* ***basic problem in multi-sensor systems*** *is to integrate a sequence of observations from a number of different sensors into a single best-estimate of the state of the environment. (1988) [38]*
>
> ***Data fusion*** *is a formal framework in which are expressed means and tools for the alliance of data of the same scene originating from different sources. It aims at obtaining information of greater quality; the exact definition of greater quality will depend upon the application. (1998) [39]*
>
> ***Information fusion*** *is the study of efficient methods for automatically or semi-automatically transforming information from different sources and different points in time into a representation that provides effective support for human or automated decision making. (2007) [40]*

Within the scope of vision networks, given the type of data it handles and the application challenges it commonly faces, the following description is considered to highlight the most crucial parts and purpose of the fusion process. Respectively, a simple illustration of a sensor fusion process is given in Figure 3.1.

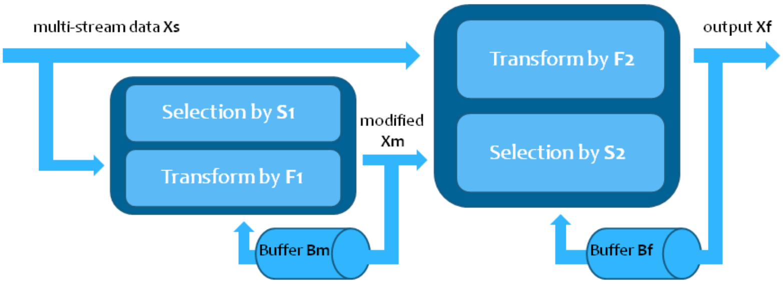


Figure 3.1: *Given a stream of data from multiple sources X_s sensor fusion combines the data into a single estimate X_f based on iterative processes of selection S_i and transform F_i on original data, already modified data X_m and previous fusion results.*

Data fusion *is a possibly iterative process of selection and transformation, in which initial multiple observations of the same property are combined into a single estimate. The estimate is expected to have captured the essential value and behavior of that property, creating increased certainty in the observed data and hence increased accuracy in decision making.*

3.2 Levels of Fusion

Data fusion can be categorized in multiple ways depending on the selected aspect. A three-level model has been proposed by Elmenreich [36] when considering the basic types of data involved in a fusion process. A refined five-level categorization was later introduced by Dasarathy [35], focusing more on the fusion process input and output characteristics. In this section, a further categorization is proposed for fusion levels in the context of vision networks. The section concludes by discussing the proposed levels and by giving illustrated examples.

3.2.1 Categorization

To introduce the basics for different types of data and the fusion processes transforming them, two categorizations presented in literature are shortly discussed.

Three-Level Model
A common practice has been to distinguish between three different fusion levels [36].

- *low-level*: images
- *intermediate-level*: features, scores
- *high-level*: decisions

Low-level fusion, or *raw data fusion*, creates new data by combining raw data given by multiple sensors. In the common three-tier Vision Network Architecture (VNA), which was previously illustrated in chapter 2 ,this applies to the Sensing layer having input as images.

Fusion at the *intermediate-level* can be performed by combining data describing a certain feature, computed from the raw source data. This process is sometimes referred to as *feature-level fusion*, corresponding to Knowledge Base in the VNA.

High-level Fusion combines the decisions or scores for decisions inferred from features into a single decision or score. In the VNA, the decision making process and thus decision fusion are part of the Application-layer.

Dasarathy Categorization
Since the three-level model does not explicitly present the fusion processes in which the input data type differs from the output data type, a refined categorization into a 5-level model was proposed by Dasarathy [35].

Offering categories based on input-output characteristics gives a more complete picture of the fusion process. The original categorization is presented in Figure 3.2. For example, raw data from multiple sensors is processed in feature extraction into a feature, and in pattern recognition a decision is inferred based on a set of features from multiple sensors. These cases could be categorized either based on input or on output data type, but this would not be an accurate description of the process that has taken place.

Proposed Four-Level-Model
Within the context of visual networks, four different types of data exist for fusion purposes:

- *image*: one or multiple stacked matrices

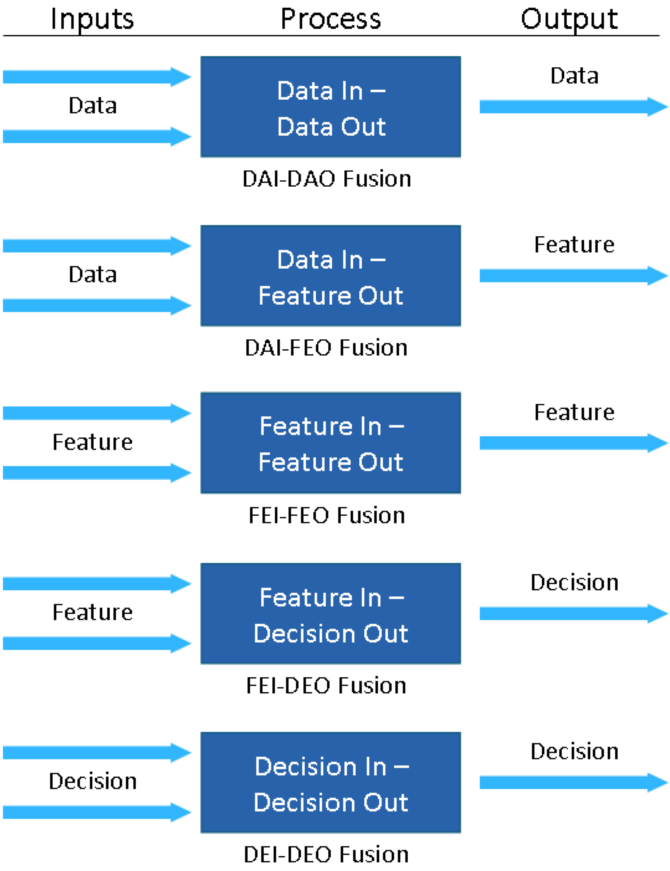


Figure 3.2: *Fusion categorization proposed by Dasarathy based on I/O characteristics of the fusion process [35].*

- *feature*: numerical value, optionally normalized to a certain range
- *score*: scores and probabilities for a class
- *decision*: discrete symbol as a class label

By not considering only the type of data but also the possible processes of fusion handling it, fusion within vision networks can be categorized as presented in Figure 3.3. A similar notation to one used by Dasarathy, presented in Figure 3.2, was used for presenting the different types of date.

Images can entail one or multiple stacked matrices filled with numerical values. Numerical values can be binary (segmented image), integer values within a certain range, such as between 0 and 255 for an intensity image, or continuous values commonly between zero and one.

Both *features* and *class-scores* are almost exclusively numerical values, commonly normalized to have continuous values between zero and one. *Class-decisions* are defined by discrete symbols. Symbols are given by a finite predefined set, in which

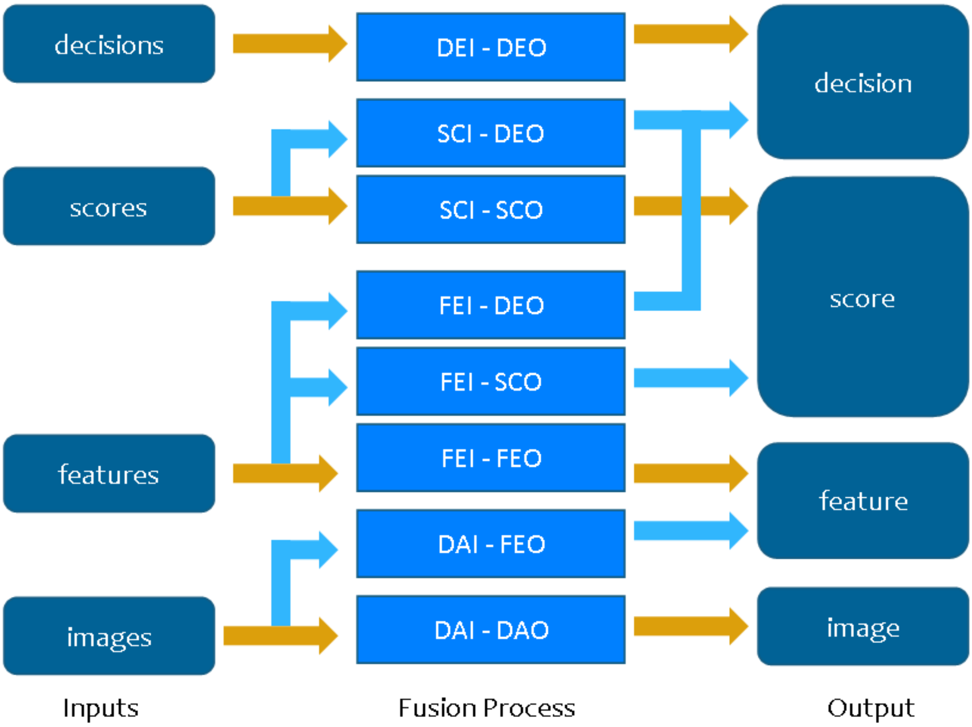


Figure 3.3: *A proposed categorization for data fusion within VNA, with respect to the data and the fusion process. The flow of the same property has been colored in yellow, light blue indicates transformation of the property.*

both the original and fusion resulted values reside. Therefore, data values within a vision network can be presented in of the three formats:

- *binary*: as in either false or true, or zero or one
- *numerical*: as in integer values or decimal values
- *symbolic*: as in a letters

Given the type of data, the possible approaches for combining data are naturally limited to the techniques that can analyze the data and be optimized for the type in question.

3.2.2 Fusion on Images

Fusion on images can be performed in multiple ways, depending on the type of images captured. Images may have been captured by different cameras, different camera settings, and with additional hardware.

Space Discovery
With the same type of images, e.g., a 3D shape reconstruction can be cross-sectioned by relying on a set of calibrated silhouette images [41]. The result of the shape

reconstruction is called Visual Hull. The Visual Hull is usually computed based on an occupancy testing for volumetric pixels called voxels. The 3D shape can be additionally colored by color consistency analysis between the views [42], see Figure 3.4 for illustration.

Multi-Spectral Capture

Sometimes it is beneficial to use cameras operating within different ranges of the light spectrum. For example, by combining an image captured on visible light spectrum with an image taken in the infrared(IR) spectrum, an improved segmentation of objects can be achieved [43]. This is due to the complementary nature of the two image modalities. Visible light image is good for capturing light intensities, colors, and textures. Whereas an IR image, a thermal image, detects heat sources, such as warm bodies of people, see Figure 3.5.

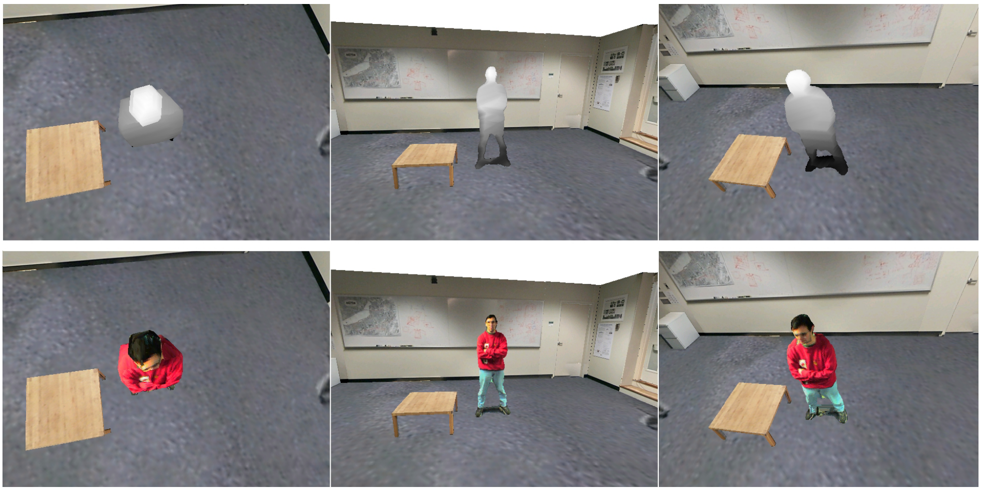


Figure 3.4: *Example of image-based visual hull (IBVH) reconstruction. The images of the top row show depth maps of the computed visual hulls. The lower images show colored renderings from the same viewpoint. [42]*

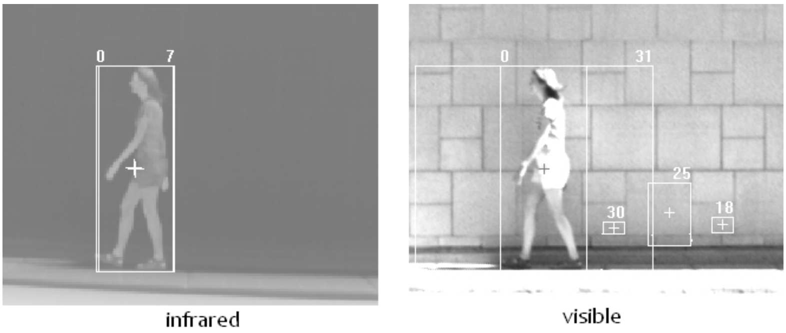


Figure 3.5: *Pedestrian detection in an outdoor scene, starting from left, original IR-image and visible-light image. With permission from [43].*

Internal-Settings Capture

The same type of images can also be used for creating filtered images and images with new information. For example, by taking photos of the same object with the same camera, but with different focus settings, depth of the object can be captured based on depth-from-defocus (DFD) methods. DFD was first introduced in [44]. An extension of a similar approach, depth-from-focus (DFF), combined with multi-baseline stereo matching to a multi-view situation for capturing 3D surface of an object is presented in [45].

An improvement to traditional DFD, that uses circular apertures of different sizes, was given in [46] in the form of coded aperture pairs. Due to their complementary nature, the coded pairs were found to provide both depth estimation with greater fidelity and a high-quality all-focused image, from the two photos captured with different apertures, see Figure 3.6.

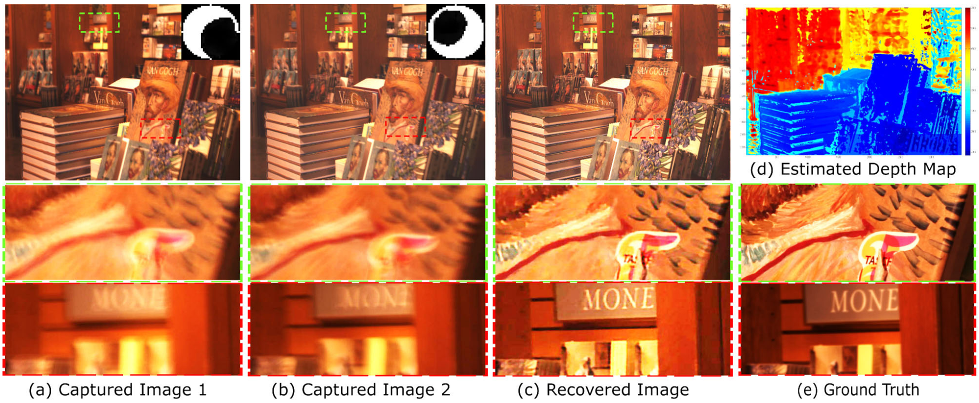


Figure 3.6: *A bookstore captured with two different apertures, a-b) captured images using the coded aperture pair with two close-ups below. The focus is set at the middle of depth of field. c) The recovered image with corresponding close-ups. d) The estimated depth map without post-processing. e) Close-ups in the ground truth image captured by a small aperture f/16 and a long exposure time. [46]*

External-Hardware Capture

A camera can be hooked to external units for more advanced imaging. Considering a camera connected to a flash light-source, two images, one with flash and one without, can be captured [47]. These two images can be combined for creating a high-detail image that preserves the object appearance, such as warmth of the colors. See the example in Figure 3.7.

Optionally, a regular color camera can be coupled with a IR emitter and IR depth sensor for acquiring RGB-D images, which are RGB images combined with per-pixel depth information, see example in Figure 3.8. Such devices have been introduced by the gaming industry first in 2010 in the form of a motion sensing device called Kinect to the Microsoft XBOX game console, soon followed by other game console manufactures. RGB-D Mapping has been introduced for modeling of environments by dense 3-dimensional maps. It is directly applicable to various fields, such as robot navigation and telepresence [48].

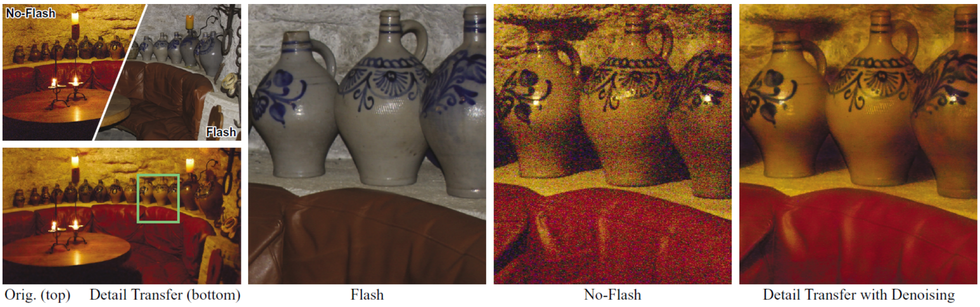

No-Flash
Flash
Orig. (top) Detail Transfer (bottom)
Flash
No-Flash
Detail Transfer with Denoising

Figure 3.7: *A flash image captures the high-frequency texture and detail, but changes the overall scene appearance to cold and gray. The no-flash image captures the overall appearance of the warm candlelight, but is very noisy. The detail information from the flash image is used to both reduce noise in the no-flash image and sharpen its details. [47]*

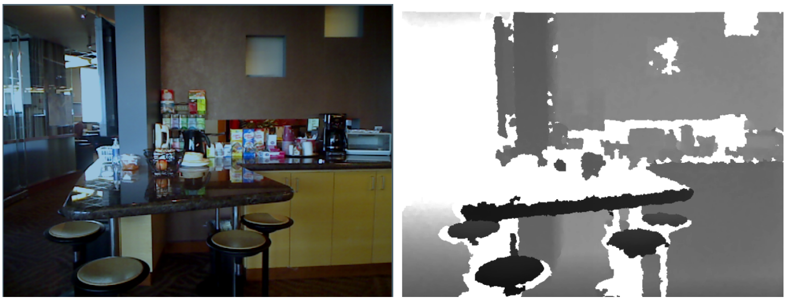


Figure 3.8: *Left) RGB image, and right) depth information captured by an RGB-D camera. Systems of 2011 can capture images up to 640x480 pixels resolution at 30 frames per second. White pixels on the right image are without depth value, mostly due to occlusions, maximum distance, and surface angle and material. With permission from [48].*

3.2.3 Fusion on Features

Commonly there exist three categories of human-related features that are derived from captured images; all three categories are suitable for data fusion. Geometric fusion covers all simple geometric descriptors for edges, lines and corners. Appearance features are more descriptive as they include posture, shape, texture and color properties. Global movement, created by the person as one entire object, and local movement, created by some body-part, are examples of motion features. Different feature types are in short:

- *geometric*: edges, lines, etc.
- *appearance*: posture, texture, etc.
- *motion*: by entire body or by a body part

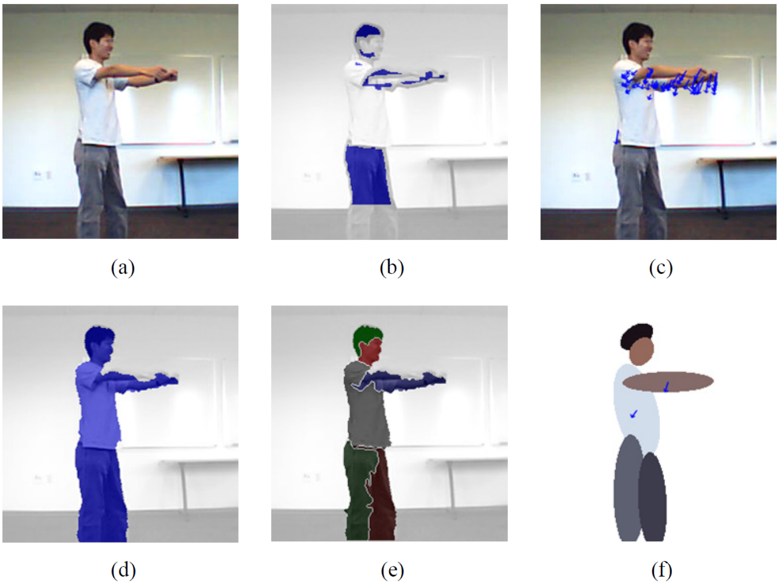


Figure 3.9: *An example of a body part segmentation based on features of motion and appearance. a) the original image, b) segmented image by background subtraction, c) optical flow as measure of motion, d) watershed segmentation based on results in b and c, e) K-means clustering of colors detected in the foreground, f) body parts represented by ellipsoids with corresponding average color and motion. With permission from [49].*

For human gesture analysis, Wu et al. [49] used a two-feature combination of optical flow and color information for initiating markers to be used in watershed segmentation for finding foreground regions. The corresponding feature of motion and feature of appearance are illustrated in Figure 3.9. Estimates for both features were used to refine each other, resulting in better features due to the complementary nature of the two features. Collaboration between cameras was performed in later stages of the vision network first for creating gesture elements, followed by inferring the gestures themselves.

3.2.4 Fusion on Scores and Probabilities

Consider a detector based on matching, that gives out a similarity/dissimilarity score as a result. If the score is not already within a fixed range, it will be normalized to a common range, thus facilitating the direct comparison and fusion of scores received from multiple sensors. A similar process can be performed given class probabilities for each class from each of the sensors. Scores are mostly used in binary-classification tasks, such as authenticating a user either as a genuine user or as an impostor. Whereas class probabilities are most beneficial with multi-class classification problems.

Fusion of alert confidences was studied by Yu et al. based on a exponentially weighted Dempster-Shafer Theory of Evidence for implementing a intrusion detection system [50]. Each sensor in the system outputs an alert with a level of confidence attached to the alert. These alerts with corresponding confidences are combined by weighting the confidences based on the trust in a specific observer/sensor for a particular observation type. The trust could, e.g., be defined according to sensor location, such as local versus remote, or by using data gathered in the past observations to analyze sensor credibility.

Fusion of audio-visual detector scores was studied by Garcia-Salicetti et al. for person authentication within multi-modal biometric analysis [51]. The two detectors processed the voice and the signature of a person, thus producing a similarity score for both of these uncorrelated biometric traits, see Figure 3.10 for illustration of the clean score distributions. The relative spread of speech scores is low and a great deal of speech scores seem to overlap between the client and impostor classes. This would suggest that speech scores do not bring much additional information to be exploited in multi-modal fusion. No comparison to single modality results was provided. The two scores were combined by two different methods for fusion, Arithmetic Mean Rule (AMR) with three different normalization techniques and Support Vector Machine (SVM). They conclude that if scores are affected by noise, the fusion method can deliver best accuracy only by taking into account the distributions of the scores.

3.2.5 Fusion on Decisions

Any decisions, such as class symbols from a predefined set, can be combined into a single decision. Decisions can include labels for persons identity, activity, posture, intention, and any other property describable by a finite set of symbols.

An exploitation of modality complementaries was proposed by Qian et al. for analysis of responses to visual target detection [52]. Both EEG and pupil responses were separately used in classifying responses in certain cognitive tasks. The classification results from both modalities where then combined by a likelihood ratio (LRT) based fusion, see Figure 3.11. The accuracy of the resulting fused decisions showed significant improvement over single modality results.

3.2.6 Fusion Across Levels and Over Time

Fusion of data can also be performed in a hybrid fashion across multiple levels of a vision network. With a hybrid approach it is possible to achieve more consistent system accuracy than when relying solely on fusion at a single system level, but the complexity and analysis will be significantly more difficult.

Given a multiresolution approach, such as detecting movements on different scales and computing the same features from each of the scales, we have multiple estimates for the same features. Similarily, a feature extractor suite can be used to provide estimates for the same feature, such as estimating the posture angle of a person with various means.

For making decisions, one can apply a classifier suite, instead of a single classifier. This results in multiple estimates of the decision, given by classifiers trained with their particular training procedures and labeling considerations of e.g. temporal behavior of feature data.

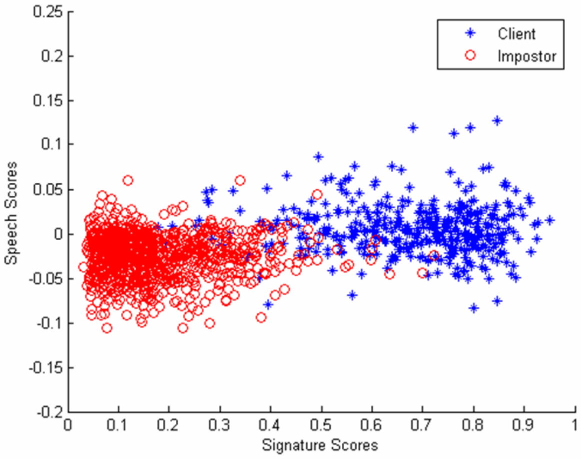


Figure 3.10: *Distributions of signature and speech detector scores [51].*

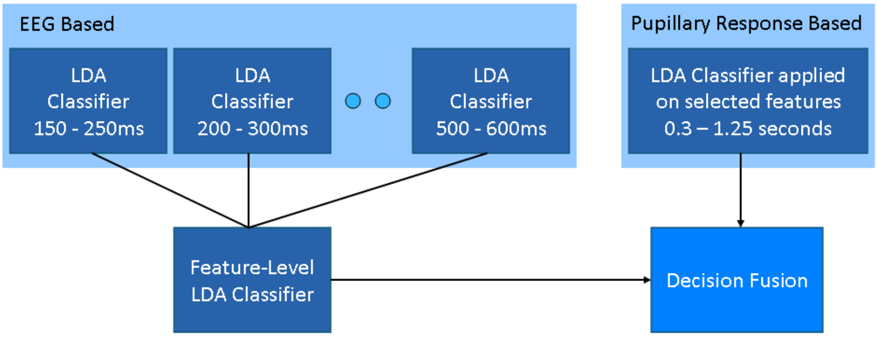

EEG Based
LDA Classifier 150 - 250ms
LDA Classifier 200 - 300ms
LDA Classifier 500 - 600ms
Pupillary Response Based
LDA Classifier applied on selected features 0.3 – 1.25 seconds
Feature-Level LDA Classifier
Decision Fusion

Figure 3.11: *An example of fusion for decisions inferred from EEG and pupillary response by linear discriminant analysis (LDA) classifiers. [52]*

Fusion can also process data from a certain period of time, looking into all the observations and deductions made within this time window. Such a process is called temporal fusion [35]. Various processes, such as temporally averaging data acquired from different sensors and keeping a track on a person, are examples of temporal fusion. Furthermore, temporal fusion of decisions given by the same or different sensors has e.g. been exploited in sequential decision problems. Therefore, temporal fusion is applicable at any of the levels of data hierarchy, thus giving another dimensionality to be exploited in fusion processing.

For recognizing human emotions, Mansoorizadeh et al. applied a two-modality classification approach with a hybrid fusion method [53]. The two signal modalities, face expressions and speech prosody (rhythm, stress, and intonation), where fused both at feature and decision level. Face and speech information were combined only when both information sources were concurrently available. Their fusion approach is illustrated in Figure 3.12. The proposed architecture achieved better average accuracy than any of the corresponding single-modality or single-level fusion approaches.

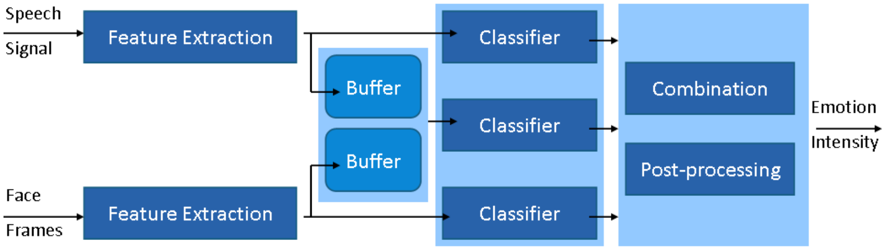


Figure 3.12: *An example of hybrid fusion on both features and decisions for recognizing six different emotional states: anger, disgust, fear, sadness, happiness, and surprise. The decision based on combined features is used as a third decision input for decision-level fusion, forming a cascade of fusion processes. [53]*

According to the authors the superior average accuracy was due to both exploiting cross correlations of features and increasing robustness by decision fusion.

3.3 Methods of Fusion

Depending on the system level, and more specifically the type of data such as images and class-decisions, different methods for fusion are applied. All the proposed fusion methods can be considered as some sort of weighted mixture of two basic processes: selection and transformation. Selection based techniques pick a single or multiple data elements out of all the available elements. Transform based techniques create one or multiple elements based on a set of given elements. Naturally, both processes can be combined by applying them in a cascade or iterative manner, enabling various fusion methods.

Categories by Elmenreich
A number of fusion methods, based on forms of selection and transform, have been proposed in literature. Depending on the field of interest, such as data mining or biometrics, different techniques have been applied. Four different categories of *fusion techniques* were presented by Elmenreich [36]:

- *Filter Algorithms*: (temporal) smoothing and prediction
- *Sensor Agreement*: sensor voting and selection
- *Decision Methods*: Bayesian and Dempster-Shafer inference
- *World Modeling*: occupancy maps and certainty grids

The above categorization by Elmenreich should be considered as a mixture of types of algorithms and types of applications for these algorithms. For example, the occupancy testing in world modeling can be performed by sensor agreement or by decision methods. Therefore, this categorization would not present a reasonable separation between the different types of fusion method algorithms.

Categories by Sanderson and Paliwal
Another categorization of fusion methods has been proposed by Sanderson and Paliwal in [54,55]. They introduced three different broad categories: pre-mapping, midst-

mapping, and post-mapping fusion. The mapping refers to either 1) a classifier providing a hard decision or 2) an expert providing an opinion, e.g., a score within the [0, 1] interval for each possible class. Pre-mapping fusion is performed before any classifier is asked to make a decision or give an opinion. Midst-mapping fusion combines the data during the mapping process that turns sensor or feature data into decisions or opinions. Post-mapping fusion is performed after the mapping process.

Sanderson and Paliwal presented the specific fusion method categories according to the well established levels of data: sensor, feature, opinion, and decision; see Figure 3.13 for illustration of their categorization.

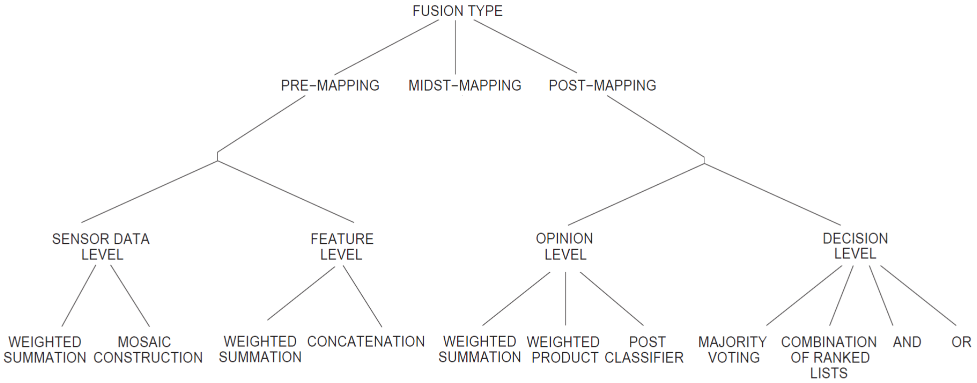


Figure 3.13: *The categorization of fusion levels and methods by Sanderson and Paliwal given in a non-exhaustive tree. Image modified from [55].*

This categorization provides a similar separation between fusion levels than the one proposed in this thesis, but it also highlights the classification step as a great divider of fusion methods. One could argue if this division is really justified as some fusion methods, such as weighted summation, are exploited on both sides of this division. Jain et al. provided a more detailed listing of fusion methods based on this categorization [11]. They also made a further separation of opinion level into two levels of 1) (matching) score and 2) rank (based on decreasing order of confidence).

Proposed Categories

In this thesis, another categorization for fusion methods is proposed based on the manner in which the data is processed:

- Sensor Agreement
- Linear Logical Methods
- Filter Algorithms
- Propabilistic Methods
- Binary Decision Methods

Sensor agreement methods, such as sensor voting and sensor selection, are methods with most accessible analysis of the system performance. That is, the contribution of individual sensors to the fusion result is not difficult to trace back. Linear Logical Methods provide the most direct and simplest approach to combining features or decisions. Filter algorithms, such as Kalman-filtering and particle-filtering, have received much attention and have been successfully employed in real-time systems for temporal fusion. Fusion on decisions has received much interest and convincing results on stability have been shown.

By connecting the previously presented fusion levels of data and the fusion methods applicable for each level, an instructional diagram of possible fusion approaches within common VNA can be defined. The proposed diagram is illustrated in Figure 3.14.

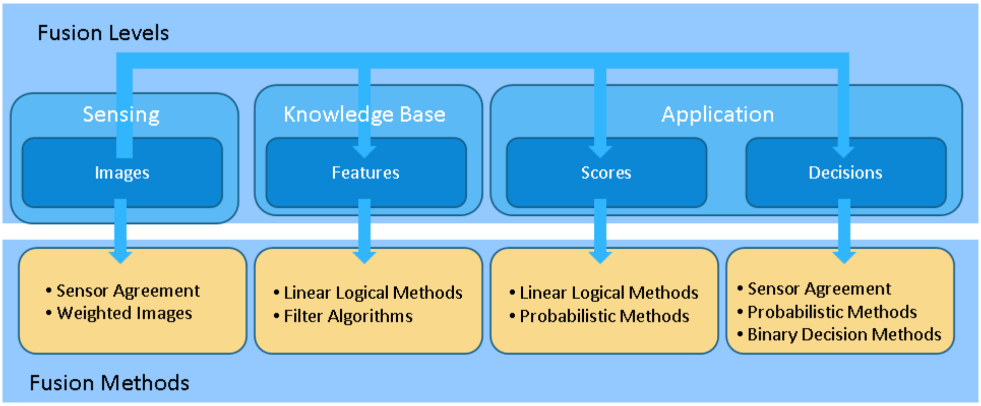


Figure 3.14: *The common fusion approaches for vision networks presented by the data levels and the applicable methods for each level.*

3.3.1 Sensor Agreement

Both sensor voting and selection methods provide an easy to implement, transparent way to combine data.

Sensor Voting

Sensor voting can be regarded as a fusion technique for solving sensor conflicts. Voting gives the solution by emphasizing the majority view of all the observing views. Voting has been regarded as computationally inexpensive and to some extent fault tolerant.

Arrow [56] pointed out that having a greater number of choices does make determining a fair voting strategy difficult, and that there exist a drawback of consistency in voting techniques. A problem of ordering of votes can emerge in a cyclical process [57], in which the point at which the vote is entered can influence the outcome to the sensors favor. As it was discussed in [58] based on the work by [59], voting strategies can be considered as the best starting approach to fusion problems, if no reliable covariance estimates for the sensors exist. Therefore, other fusion techniques can regard sensor voting as a performance benchmark for improving performance by incorporating a required level of learning to the fusion process.

A commonly used voting method is the so called *majority voting*. As its name implies, the result from fusion is defined as the decision having the majority of the classifiers inferring it as the decision candidate. A paper dealing with speaker identification [60] used two approaches, dynamic time warping (DTW) and cepstral coefficient distance measure, to classify the speaker and combined the classifier outputs over a period of time. When employing the combination of classifiers, majority voting provided the best accuracy by correctly recognizing 98 out of 100 speakers.

In [61] a technique for multi-sensor data fusion based on voting implemented through boolean algebra was presented. Three different sensors were used as input sources for illustrating the boolean approach, the sensors being millimeter-wave (MMW) radar, laser radar and imaging infrared radiometer. Given multiple confidence levels to each sensor, the proposed sensor fusion, implementable by AND and OR logic operators, increased the performance of the sensor system and enabled specification of a certain false alarm probability. The approach proposed in the paper is illustrated in Figure 3.15.

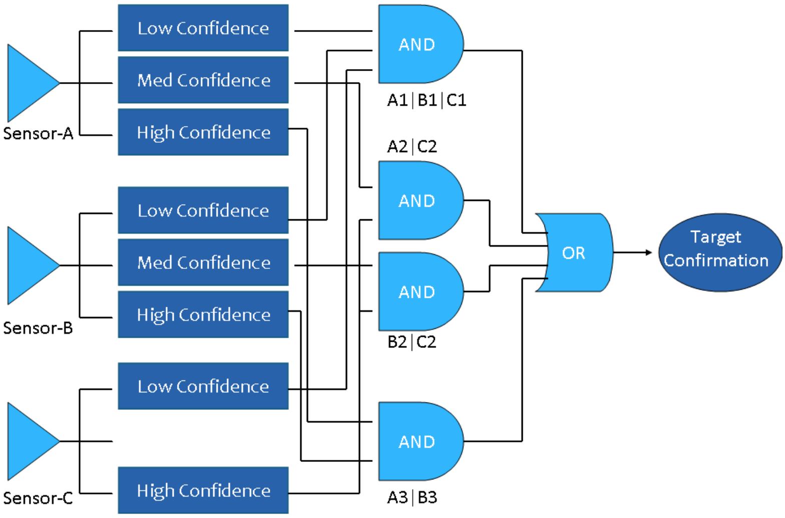


Figure 3.15: *An example of a Sensor Voting based data fusion implementable by logical operators. [61]*

Another voting fusion scheme was presented in [62] in the context of antitank landmine detection by a remotely controlled vehicle equipped with multiple sensors. Three detection sensors were mounted in front of the vehicle, which was expected to increase the probability of detecting landmines. The three sensors used were Forward Looking Infrared (FLIR) camera, a Minimum Metal Detector (MMD), and Ground Penetrating Radar (GPR). For correctly combining the sensor inputs, the intervals for three confidence levels: low, medium and high, were found by processing trial data. Through voting based data fusion they were able to limit the number of false alarms to a practically manageable level.

Sensor Selection

Data fusion methods based on *sensor selection* have proven to be efficient for many applications, especially given many sensors S_i aiming for achieving optimal system performance, often in an energy-aware manner. Regardless of selecting either a set S_{set} of one or many of the available sensors S_{all}, a criterion or criteria is always defined as a measure of the suitability of a certain sensor set S_{cand}. Based on the suitability scores $Suit(S_i)$ sensors can be ranked and only the data from the best sensors is used. Additionally, a cost function can be included to balance/restrict the selections due to context, coverage, energy-consumption and other similar concerns. A compact linear representation is presented in the following general formulation of sensor selection:

$$S_{set} = \underset{S_{cand}}{\operatorname{argmax}} \left[\sum_{S_i \in S_{cand}} Suit(S_i) - Cost(S_i) \right], \text{ with } S_{cand} \subseteq S_{all} \tag{3.1}$$

A geometrical approach to sensor selection was proposed in [63]. The approach was demonstrated by two cases: initial sensor placement, and sensor relevancy during operation. Measurement noise was modeled by a Gaussian approximation. The selection was based on the distance between the desired and the available information. In [64] an energy-aware sensor selection method was proposed for the detection of social context such as activity and location. As the resources for sensors between and within a mobile phone were considered limited, the aim was to maximize the 'Quality of Information' (QoI) of context detection. The QoI provided by a sensor was directly related to its accuracy, based on which the overall QoI was computed. Two methods, brute force and heuristic, were provided for computing the overall QoI. Results presented showed that the device life could be prolonged without sacrificing the diversity of complementary information.

For achieving a good trade-off between energy consumption and quality of tracking, a dynamic sensor selection method was proposed in [65]. Particle filtering was deployed for object tracking, and tracking was performed by a varying number of sensors at any given time, depending on the sensor selection scheme. The scheme was named Niche Overlap-based Sensor Selection (NOS) and it adapts to the changes in tracking conditions, such as sensor network density and the geometry of the target and sensors. As the object moves through the scene, crossing areas having different niche overlap measurements, the conditions change. The proposed NOS method outperformed some rivaling methods by offering both higher tracking accuracy and longer lifetime.

In the context of event detection, Bajovic et al. proposed a sensor selection method [66] using the Kullback-Leibler and Chernoff distances between Gaussian distributions as the two selection criteria. The distances were computed between the distributions of the selected measurements under the two hypothesis of event occurring and not occurring. They proposed an algorithm with affordable complexity for computing the distances. The proposed solution for selecting a subset of sensors performed similar with the optimal selection in many cases, staying 5% below the optimal accuracy at all times, providing significant computational savings at the same time.

3.3.2 Linear Logical Methods

Two commonly used ways for combining measurements or features from multiple sensors are *concatenation* and *transform.*

Concatenation

In feature-concatenation the feature-vectors $\boldsymbol{f}_i$ from all sensors S_i are combined by simply stacking the feature-vectors one after another. This results in a feature-vector containing all the feature information from all r sensors:

$$\boldsymbol{f}_{concat} = [\boldsymbol{f}_1, \boldsymbol{f}_2, .., \boldsymbol{f}_r] \tag{3.2}$$

A similar concatenation operation can be performed on images taken of the same subject with each camera providing observation of a different part. In this manner one image is created out of many, a process that is referred to as *mosaic construction.* An example of mosaic construction is given in Figure 3.16 provided by a paper dealing with wide area aerial surveillance exploiting piecewise affine modeling for image distortions.

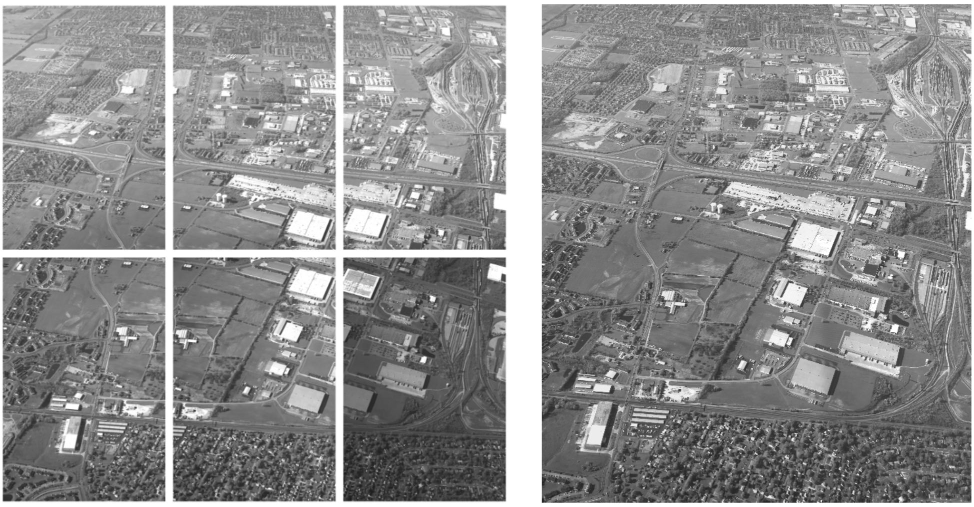


Figure 3.16: *Image Mosaic: left) an array of cameras captures part of a scenery, right) a single virtual image covering entire area is created for scene analysis. With permission from [67].*

Three drawbacks can be identified for concatenation as a fusion method. First, there is no explicit control mechanism to define the relative influence of a sensor or reversily to understand the constribution of a single sensor on the final result or decision. Second, the observations need to be provided in a synchronous manner with the same frame rates for the fusion to perform well. Any deviations in time (features) or shifts in place (images) can not be tolerated. The third drawback is more specific to feature concatenation, as the dimensions of the combination feature-vector can become large, and thus the feature-space involved will become huge. This can easily lead to the so called *curse of dimensionality* problem [68], in which the data becomes sparse, which again weakens the chances for obtaining sound and reliable results based on statistical analysis.

A method called combined-View, which combines sensor measurements by concatenating them into a single observation-vector, was studied in [69] in comparison two decision-level fusion methods: a best-view selection and a mixed-view using data from all cameras. Combined-view was found to have low complexity, longer training times, and mid-level classification accuracy, around 85% in the studied multi-view activity recognition. However, it does suffer from low transferability, as it can only be applied to the same setup and order of multiple cameras.

Transform
The second common approach aims to find the most representative form of a single feature-vector, based on all the given feature-vectors. This is achieved by a transformation function, optionally driven by a defined criterion *crit*, such as the minimum-mean-square-error (MMSE). Other options for the transformation function exist such as minimum, maximum, median and other similar operations aiming to capture a particular aspect of the data.

$$\boldsymbol{f}_{transform} = F\{[\boldsymbol{f}_1, \boldsymbol{f}_2, .., \boldsymbol{f}_r] \, | crit\} \tag{3.3}$$

Posterior Rules Some of the simplest approaches to decision making are simple rules forced on the posterior probabilities of class decisions. Consider r sensors S_i producing the posterior probability of a class c_j as $P(c_j|\boldsymbol{f}_i)$. Let $c \in \{1, 2, .., m\}$ be the class that is finally assigned. The following fusion rules can be applied to determine the class.

Product Rule
Assuming statistical independence of all feature vectors $\boldsymbol{f}_i$, the class c can be given by Product Rule as follows:

$$c = \underset{j}{\operatorname{argmax}} \, \Pi_{i=1}^{r} P(c_j|\boldsymbol{f}_i) \tag{3.4}$$

Sum Rule
Assuming also that the posterior probabilities computed do not deviate much from the prior probabilities, the Sum Rule can be applied:

$$c = \underset{j}{\operatorname{argmax}} \sum_{i=1}^{r} P(c_j|\boldsymbol{f}_i) \tag{3.5}$$

Max and Min Rules
Similarly Max Rule and Min Rule can be deployed to find the best estimate for the class, as either maximum or minimum value respectively:

$$c = \underset{j}{\operatorname{argmax}} \left[\max_{i} P(c_j|\boldsymbol{f}_i) \right] \tag{3.6}$$

$$c = \underset{j}{\operatorname{argmax}} \left[\min_{i} P(c_j|\boldsymbol{f}_i) \right] \tag{3.7}$$

All the above fusion rules are very simple to implement and intuitive to understand. Kittler et al. studied the above rules in their theoretical framework on compound classification by starting from the Bayesian decision rule and by deriving the rules based on statistical assumptions [70]. For the product rule they pointed out its severity as, e.g., a single close to zero probability can effectively inhibit a particular class to be chosen almost regardless of the opinions of the other sources. They found the sum rule to provide better accuracy than other rules such as the product rule, min rule, max rule, median rule, and majority voting. They concluded based on sensitivity analysis that the superiority of the sum rule is most likely due to its robustness against estimation errors.

Weighted Linear Summation

The so called *Weighted Linear Sum* (WLS) method can be used to combine scores or probabilities from multiple sensors, assuming they can be considered as independent. The accumulated evidence for each class c is the weighted sum of the individual evidence, in this example class score, given by each of the sensors S_i:

$$score(c_j) = \sum_{i=1}^{r} [w_i \times score_i(c_j)] \tag{3.8}$$

As presented in [71] for detection of focus of participants in a meeting, both observations of who is speaking and who is facing who can be combined in a following manner. The term α was used to define the importance of observations with regards to each other. A suitable value for α for weighting decisions from the two modalities was found through empirical studies. The basic combination rule was written as follows:

$$p\,(Focus) = (1-\alpha)P(Focus|Gaze) + \alpha P(Focus|Sound) \tag{3.9}$$

The WLS is also a simple and intuitive fusion method, that provides an explicit mechanism for prioritizing between multiple sensors and modalities. Because of the weighting mechanism, sensors with less occlusions or modalities with better robustness can be emphasized accordingly. The WLS requires no training and performs often good compared to other logical fusion methods previously proposed in literature [71]. The WLS method has achieved the same levels of accuracy in many scenarios as the more complicated methods based on Dempster-Shafer (DS) theory and the weighted D-S approach [72].

3.3.3 Filter Algorithms

Filtering has been widely used for smoothing previous observations and for predicting new observations based on temporal data. Two of the most applied filtering methods for data fusion have been *Kalman Filtering* and *Particle Filtering*. The most common usage for filters has been in the domain of temporal data, or sequential data, for estimating a value that best represents the gathered data, and thus provides the optimal system accuracy. Similar estimation can be performed for data simultaneously gathered from multiple sensors.

Kalman Filtering

Kalman Filtering provides an improved estimate of a state based on series of measurements, either by multiple sensors or over time. Kalman filters have been applied to numerous applications, such as navigation and guidance, some of these discussed below.

The Kalman-filter functions in two-steps. The filter produces estimates of the current state variables and their uncertainties in the prediction step. In the second step, called update step, the estimates are refined by a weighted average based on the new measurement, latest observation. More weight is given to estimates with higher certainty. Therefore, the filter requires only the present observation and the previously calculated state for its recursive nature to provide real-time performance. Kalman assumes the underlying system to be a linear dynamical system with all the error terms and measurements having a (multivariate) Gaussian distribution.

Extensions of the Kalman filter for nonlinear systems have been developed, such as Extended - and Unscented Kalman Filter.

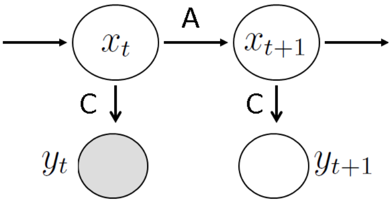


Figure 3.17: *The relations of states and measurements of a Kalman filter presented as a graphical model. [73]*

A graphical illustration of a Kalman filter is presented in Figure 3.17, in which x_t refers to a state consisting of $m \times 1$ mean vector $\hat{x}$, and a $m \times m$ covariance matrix P with m being the number of parameters. Considering a standard linear Kalman filter, the state transition from time instant t to $t+1$ is defined as follows:

$$x_{t+1} = Ax_t + w_t \tag{3.10}$$

$$p(w) \sim N(0, Q) \tag{3.11}$$

Matrix A defines the *state transition matrix* and w_t is used for presenting the noise as a Gaussian random variable with zero mean and a *process noise covariance matrix* Q. Q is assumed independent of the state and its purpose is to model the possible changes between consecutive states that are not dealt with the state transition matrix.

In addition, the relations between the states and the measurements can be expressed as follows. The state is related to a measurement by matrix C, which is an $m \times n$ matrix. Measurement noise is presented by v_t with a zero mean and R as the *measurement noise covariance matrix.*

$$y_t = Cx_t + v_t \tag{3.12}$$

$$p(v) \sim N(0, R) \tag{3.13}$$

Due to common difficulty in measuring uncertainty in the measurements, the behavior of a filter can be discussed in terms of gain, Kalman filter gain especially. With a high gain, the filter places more weight on following the trend set by the measurements. With a low filter gain, the model predictions are followed more closely. This smooths out noise, but decreases the responsiveness of the filter.

Two different ways of measurement fusion by Kalman Filtering were studied in [74]. First Kalman-based method performs the measurement fusion as a concatenation of the feature data. The lengthened feature vector is given to a single Kalman-filter for getting a state estimate. The second method merges the measurements into same as original length feature vector based on minimum-mean-square-error criterion. The merged feature vector is given to a single Kalman filter for obtaining the final state estimate. Both methods were found functionally equivalent if the sensors had identical measurement matrices. The second method being more efficient due to shorter feature vector length. However, if sensor identicality did not hold, the first method outperformed the second one.

For advanced tactical systems, Puranik et al. investigated different fusion architectures based on Kalman filters in the context of data fusion and track fusion [75]. Three different fusion architectures were studied: centralized fusion, decentralized synchronous fusion, and asynchronous decentralized fusion. Each of the fusion architectures were able to provide more certainty in location and speed estimations than a single track estimate, asynchronous fusion providing less certainty than other architectures.

In [76] a study into sensor-data fusion was performed in the context of an autonomous vehicle equipped with a few different sensors, such as magnet position-measurement, camera-aided GPS-like positioning and detector for wheel-rotations. The redundant signals measured were combined by a nonlinear Kalman-filter that produced a good signal hardly affected by disturbed or missing observations.

3.3.4 Propabilistic Methods

Two of the most widely used decision methods in data fusion are Bayesian networks and the Dempster-Shafer Theory of Evidence. These decision methods are based on inference.

Inference is a process in which a decision is made between multiple propositions based on the knowledge and observations available. In *classical inference* the decision is made between multiple hypothesis H_i, from which one is chosen to be more probable than the others, given the observation O at hand. No a priori information about the likelihoods of different hypotheses are included in classical inference [77].

Bayesian Inference
Bayes introduced a priori information for hypothesis H as part of his theorem [78], defined as follows:

$$P(H|O) = \frac{P(O|H)\,P(H)}{P(O)} \tag{3.14}$$

With *Bayesian inference* the probability of numerous hypotheses can be tested against each other with a prior information part of the decision making. Bayesian inference computes the probability of each of the hypothesis given the observation $P(H_i|O)$, and declares the hypothesis with the highest probability as the true hypothesis of all the n number of hypothesis being tested.

$$P(H_i|O) = \frac{P(O|H_i)\,P(H_i)}{\sum_{i=1}^{n}\left[P(O|H_i)\,P(H_i)\right]} \tag{3.15}$$

For sensor fusion, Bayesian inference has one major drawback, the required initial knowledge. The Bayesian process requires the probabilities for an observation $P(O)$, as defined in Equation 3.14, and observations conditional probability $P(O|H_i)$, as defined in Equation 3.15, to be known for the system in question. Some of these probabilities may not always be empirically available, but as they need to be provided, personal judgement can be used to provide subjective probabilities [77]. However, subjective probabilities are not beneficial for inference accuracy.

Bayesian inference has been applied to various applications, such as autonomous sensor networks [79]. For multi-vehicle localizations, in [80] Bayesian networks were used in both combining the data from the measurement sensors of each vehicle and in modeling the vehicle interconnections globally. A single vehicle was localized based on fused information from three sources, Differential Global Positioning System (DGPS), ABS-wheel sensor based odometry, and Geographical Information System based road extraction. This information was further fused by a Bayesian approach with previous vehicle pose estimates. For localizing multiple vehicles, the leader of the train was used as a starting reference, following vehicles only compared by a rangefinder to the vehicle in front.

In [81] a Bayesian sensor fusion formulation was provided to infer all three-dimensional static occlusion objects based on silhouette clues of dynamic targets captured from various camera viewpoints. The approach explicitly aimed to detect the occluding objects present in the scene, by applying Bayesian estimation to the decision of occupancy of a certain part of space, based on the Shape-From-Silhouette method [41]. Due to the Bayesian approach for detection and accumulation of occluder information, the results showed robustness to noise and potential for many applications by offering only soft decisions about scene state.

Dempster-Shafer Theory

Another mathematical approach to decision making was introduced by Dempster in 1967 [82]. Whereas Bayes dealt with probabilities, Dempster's rule of combination operated on beliefs and mass functions. Dempster's work was further developed by his student Shafer by 1976 into what is nowadays referred to as *Dempster-Shafer Mathematical Theory of Evidence* (D-S Theory) [83]. This theory has been applied since for representing incomplete knowledge, updating beliefs, and for combining evidence [36].

In D-S theory Ω is used to refer to *the frame of discernment.* Ω contains all the mutually exclusive propositions individually and as a union of propositions. 'The frame of discernment' Ω in authentication of two persons could e.g. be defined by following propositions:

$$\Omega = \{\text{personA, personB,}\{\text{personA, personB}\}, \text{someone else}\} \tag{3.16}$$

Each sensor S_i will assign its beliefs over Ω, a process commonly referred to as Basic Probability Assignment (BPA). This assignment is done by a belief function m_i by sensor S_i. Function m has the following constraints [83]:

$$m(\emptyset) = 0 \qquad\qquad \sum_{A \subseteq \Omega} m(A) = 1$$

To aggregate the total belief in a subset A, all the BPAs of all the subsets B of A need to be summed. Resulting in a *belief* measurement that presents how much all the available evidence support the proposition A:

$$Belief(\emptyset) = 0 \qquad\qquad Belief(\Omega) = 1$$

$$Belief(A) = \sum_{B \subseteq A} m(B), \forall A \subseteq \Omega \tag{3.17}$$

Some of the remaining evidence may be assigned to propositions that are joint with A, therefore are not necessarily supporting the negation of A. All the observations that do not exclude the given proposition define the *plausibility* of a proposition:

$$Plausability(A) = \sum_{B\cap A\neq\emptyset} m(B) = 1 - \sum_{B\cap A=\emptyset} m(B), \forall A \subseteq \Omega \tag{3.18}$$

The combination rule, marked by operator $\oplus$, of D-S theory for computing the probability of a proposition A given observations from two sensors i and j can therefore be written as follows, for all $A \subseteq \Omega \neq \emptyset$:

$$\begin{aligned}(m_i \oplus m_j)(A) &= \frac{\sum_{X\cap Y=A} m_i(X)m_j(Y)}{1-\sum_{X\cap Y=\emptyset} m_i(X)m_j(Y)} && (3.19)\\ &= \frac{1}{1-K(m_i,m_j)}\sum_{X\cap Y=A} m_i(X)m_j(Y), && (3.20)\\ &\text{with } K(m_i,m_j) = \sum_{X\cap Y=\emptyset} m_i(X)m_j(Y) && (3.21)\end{aligned}$$

K is used to highlight the *conflict* of the two sources of evidence m_i and m_j by aggregating the total contradiction, or reversely the *agreement* as $1 - K$ [84].

Examples of the usage of D-S theory can be found for many application purposes. In [72] the D-S theory is reviewed, some weakpoints are addressed, and a solution to these weakpoints is given in the form of a weighted Dempster-Shafer Evidence combination rule. The weighted approach is introduced to mitigate the effects of the 'equally trusting' approach to sensors of the original D-S theory. Meeting participants focus of attention was used as data for the comparison of three fusion methods, weighted linear sum method, standard D-S method, and the weighted D-S method. The data was gathered by a omni-directional camera observing all the participants, each of which had additionally a microphone standing in front for capturing audio activity. Head-pose estimates were extracted as features from video, from audio a simple flag for declaring engagement by voice was used as feature. All three compared fusion methods achieved higher accuracy than relying on single signal modality, but no significant differences between the three were discovered.

A similar, but empirical, study into sensor uncertainty is performed in [84] using D-S theory as the fusion method. They propose an uncertainty based technique for quantifying the evolution of D-S theory based fusion. The technique is demonstrated in the context of identifying different aircrafts by using Electonics Support Measure (ESM) sensors, by finding the optimal combination of sensors for achieving the least uncertainty.

D-S theory has been widely used in applications of biometrics. In [85] D-S theory was deployed for person authentication based on face and voice score levels. The scores are given by the verification systems of each modality, and are mapped into evidence values for enabling fusion of modalities. D-S method was compared between two additional fusion methods, the sum rule and the log-likelihood ratio based fusion. All fusion methods outperformed any of the single modalities, and D-S offered better performance compared to other fusion methods explored.

3.3.5 Binary Decision Methods

Given two hypotheses, H_0 for no-detection and H_1 for detection, and local decisions u_i given by each sensor in the network, a final global decision u_0 is defined based on the local decisions. This decentralized hypothesis testing problem is solved by hypothesis methods. Tenney and Sandell developed the theory for obtaining the distributed Bayesian detection rules for decision combining [8]. The theory helps in finding the optimum decision rule for each sensor in a sensor network.

Chair-Varshney Rule
For finding the optimal fusion structure for the given detectors an optimum data fusion structure was given by Chair and Varshney [9] in 1986.

Within the fusion structure the local decisions u_i are weighted with w_i according to the reliability of the detector. Reliability is defined through probabilities for false hits P_{F_i} and misses P_{M_i}. The global decision is defined by comparing the weighted sum to a threshold. Their findings have later been referred to as the Chair-Varshney fusion rule, definitions are given here below:

$$DFR(u_i, .., u_m) = \begin{cases} 1 & , \text{if } w_0 + \sum_{i=1}^{m} w_i u_i > 0 \\ 0 & , \text{otherwise} \end{cases} \tag{3.22}$$

$$w_0 = \log \frac{P_1}{P_0} = \log \frac{P(H_1)}{P(H_0)} \tag{3.23}$$

$$w_i = \begin{cases} \log \frac{1-P_{M_i}}{P_{F_i}} & , \text{if } u_i = 1 \\ \log \frac{1-P_{F_i}}{P_{M_i}} & , \text{if } u_i = -1 \end{cases} \tag{3.24}$$

Factors for Optimality
Stearns demonstrated in 1983 that optimal fusion accuracy on decisions by a sensor network can be only achieved if taking into account two factors: the global false-alarm probability, and the type of k-out-of-n fusion rule applied [86]. The most commonly used k-out-of-n fusion rules have been the AND and OR fusion rules, both of which operate as their name suggests. Stearns studied a two-sensor case and concluded that AND was superior to OR at low false-alarm probabilities, whereas OR was superior to AND with high false-alarm rates. A third contributing factor to optimal fusion rule was introduced by Aziz et al. in 1997:

- type of k-out-of-n fusion rule
- probability of global false-alarm w.r.t. probability of detection
- type of probability distribution of observations.

They discovered that the type of observation probability distribution, such as Gaussian or exponential, influenced the selection of optimal fusion rule [10]. Therefore, it can concluded that the optimum fusion rule for multi-sensor distributed detection consists of the above three factors.

3.4 Remarks

In this chapter the existing theory for sensor fusion was presented and a more suitable definition for vision networks was proposed. Based on general theory on fusion levels and fusion methods, more suitable categorizations for vision networks were provided and some illustrative examples were shown.

Fusion within vision networks can be considered to have two functions. On one hand the function of fusion is to build certainty and stability from multiple observations. On the other hand to create robust sensitivity for capturing low-visibility properties. Which of these two functions is more important will depend on the applications, and should be reflected in the fusion approach.

The availability of a fusion level and a fusion method depend on each other. Only certain methods can be applied at certain levels and vice versa. A similar approach to the fusion method, such as majority voting, can be implemented across fusion levels, but functionality of the method approach will not be exactly the same across levels. At feature level, the majority voting approach can be implemented e.g. by an averaging operation. Whereas at decision level, averaging operation on symbols is not possible, but counting the class label that has the majority is. In this case averaging and majority voting were considered to be comparable in functionality, but the decision results will not be exactly the same. Therefore, it is not easy to correctly compare the results each fusion levels gives, as the functionality of the fusion method approach has to be slightly different at each of the levels. In contrast, if one defines the level of fusion, the results given by various fusion methods at that level can be rightfully directly compared.

Sensor agreement fusion methods provide a good starting point for computing benchmark fusion accuracy in decision making. Linear logical methods can be used to tailor the fusion for capturing specific sides of the data, such as maximums, and enables the use of weights for setting relative influence between sensors. Filter algorithms can be successfully adopted for handling multiple simultaneous estimates, instead of sequential data, but algorithmic complexity is high. Probabilistic methods may require prior information which might be hard to come by, but would provide increased accuracy. Bayesian inference and D-S Theory have been widely used in many research fields giving good, robust results.

Within vision networks there are considered to be four fusion levels, each level with their own set of possible fusion methods. This creates a large space of possible combinations to test, if done exhaustively. The proposed framework, design rules and experiments later in this thesis aim to help in avoiding exhaustive experimentation by following a systematical approach.

Chapter 4

Fusion Architectures for Vision Networks

The choice on the way fusion will be applied in a vision network is driven by the requirements from the application and the service it provides. The manner in which data is gathered and possibly further distributed is defined by the fusion architecture. It is probable that no fusion architecture that would perform optimally in all settings exists. To meet the requirements, a suitable architecture for applying the different fusion methods at different fusion levels has to be defined.

Section 4.1 starts the chapter by providing two different categorizations for fusion architectures, based on which a three category structure is proposed. Regardless of the architecture chosen the data has to be captured, transmitted and processed. For providing all this data in a timely manner, aspects in synchronization of sensors and data delivery are highlighted in section 4.2.

Many models such as JDL, Waterfall and Boyd have been proposed over the years for defining architectures. Many of these models were originally tied to a certain application-domain. The main aspects of these architecture models and their suitability for vision network fusion architecture is shortly discussed in section 4.3. The chapter concludes in section 4.4 by providing a short summary of Part-I of this thesis and its connections to Part-II.

4.1 Architectures for Fusion

Having multiple sensors gives opportunities for performing the fusion in many locations of the system and physical site. Three multi-sensor management architecture categories of centralized, decentralized, and hierarchical were presented in [87] by Xiong and Svensson. Categorization to hierarchical and fully distributed was given in [88] by Sinha et al.

Xiong and Svensson
In the *centralized* approach, all the sensory data is gathered in a central unit. The central unit receives the data, combines it, and gives out the resulting fusion data. This approach is straightforward to implement, fusion results can be expected to be of high accuracy, but approach lacks in flexibility, such as including many additional sensors.

Decentralized approach pushes the fusion process to individual sensors, thus making sensors capable of receiving data from others with added intelligence on board of the sensor. Added intelligence makes the sensors themselves more expensive, local communication between a subset of sensors requires additional coordination efforts, and redundancy in the data can have negative effects on system accuracy. The decentralized approach does offer better scalability, requiring less from resources, and flexibility, in case of sensor failure. Also the design and implementation of the decentralized approach can be performed in a modular manner.

The third category, *hierarchical*, is a mixture of the two previous approaches. Therefore, the hierarchical approach relies on having some local fusion sensor networks together with a global central fusion unit. The major difficulty in this hybrid approach is how to map the global requirements to the local-level for achieving highest accuracy.

Sinha et al.
In contrast, an architecture categorization to two classes, *hierarchical* and *fully distributed*, was presented in [88]. Within hierarchical structure, the combined data from local fusion centers is transmitted to higher level fusion center(s), which combine the estimates and depending on the level of hierarchy send the new estimates further up. Whereas in fully distributed architecture the fusion centers broadcast their estimates to other fusion centers, which by incorporating the new arrived information update their estimates respectively.

The hierarchical architectures are considered to provide less robustness, because a failure of a higher order fusion center makes all subordinate fusion centers unusable for fusion. On the other hand, a fully distributed system requires overall more computational efforts and they do not provide a single, comprehensive, global picture of the system at hand.

Proposed Categorization
In reflection to the two alternatives presented for architecture categorization, another structure is proposed in this thesis. Three classes of architectures are expected to exist: fully centralized, fully distributed, and hierarchical. See Figure 4.1 for an illustration. These three categories are assumed to count for all viable fusion architectures. Given simple sensors, a fully centralized solution is opted for. Having no desire or resources for a central fusion processing, fully distributed architecture pushes the fusion process as part of the local fusion center, possibly a device equipped with

multiple sensors. Hierarchical solutions can be benefited from once having access to fusion both at local and global level.

Some data fusion and track fusion methods based on Kalman Filtering were studied in [75] for two different architectures, adhering to our proposed notation, fully centralized and fully distributed. The fully centralized was found to provide the optimal solution, and thus can be used as benchmark for proposing any other fusion architectures.

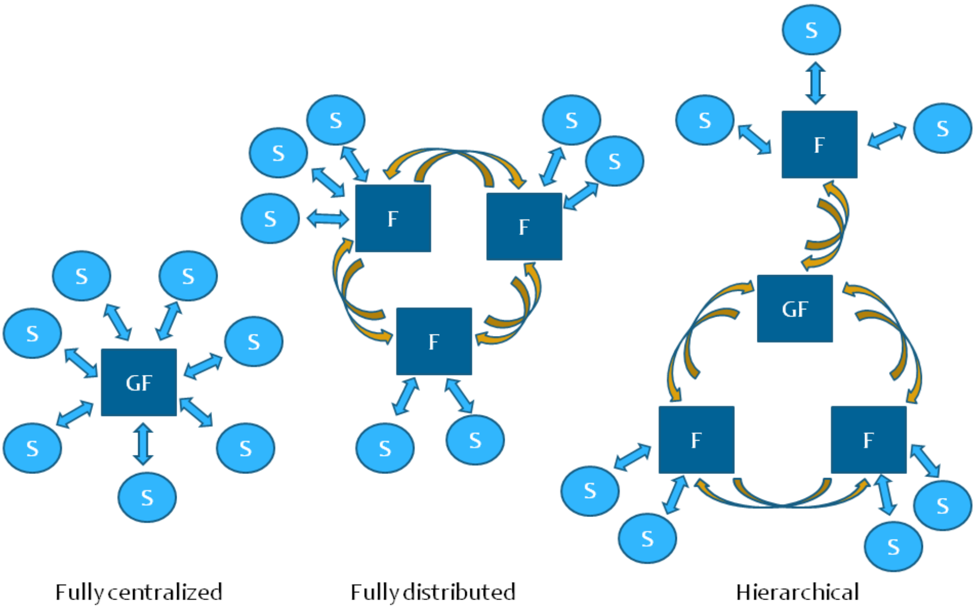


Figure 4.1: Examples of the three proposed fusion architectures. Nodes are marked with S for sensor, F for fusion, and GF for global fusion center which gathers previous fusion results and computes the final estimate.

4.2 Synchronization for Fusion

Data captured from a scene at different times, when expecting samples from the same time instant, can have serious degrading effects on the accuracy of any fusion method. Data transmission might be corrupted and contain lag, mostly due to geometrically dispersed sensors. The longer the data has to travel, the weaker the data carrying signal becomes, making distortions more probable. Sensors may also operate at different frequencies which might require compensation. Or simply the bandwidth of the transmission channel is not sufficient for carrying all the data simultaneously. For example, for cameras it is not uncommon to drop frames while capturing a scene, an issue that can cause serious shifts if left untreated. In Figure 4.2 an illustration of the problems related to data synchronization is provided.

The issues and challenges arising in sensor (vision) networks for synchronization can therefore be accounted for three sources:

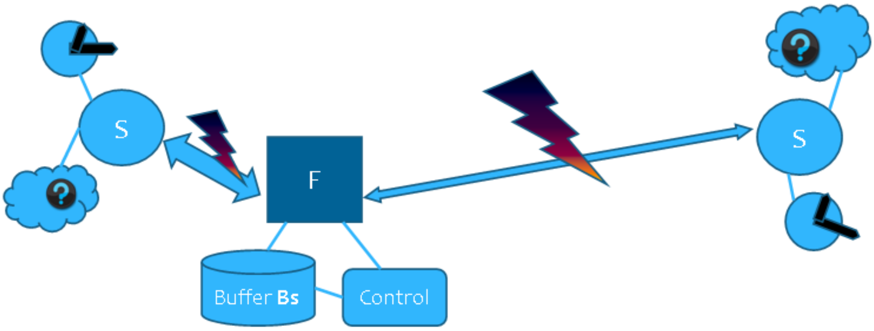


Figure 4.2: *An example situation of two sensors S impacted by the uncertainties in capture and operating frequency, transmitting over noisy channels of varying bandwidth, distance and data corruption, received by fusion center F equipped with data buffer and control mechanisms for possibly asynchronously received transmissions.*

- sensor capture
- data transmission
- fusion processing

Due to the difficulties, it is not always possible to guarantee a synchronized delivery of data streams, which is usually considered as a part of quality of service (QoS). If transmission problems stay within an acceptable range of time, they can be ignored, largely depending on the demands from the system processing steps. If ignoring them is not an option, measures need to be taken for assuring that data being processed corresponds to desired requirements. These measures fall under data synchronization.

A common way of assuring relatively closely synchronized data streams, is to revert to a well established protocol for handling the time shifts. Application layer protocols such as Real-Time Transport Protocol (RTP) and Network Time Protocol (NTP) exploit time-stamping of data packets for ensuring a proper handling of data streams. Another more crude and less accurate option has been to simply buffer the data and use the average or median value as the estimate for a given time instant.

In the study presented in [75] for data fusion methods based on Kalman Filtering, synchronous and asynchronous transmission within a fully distributed architecture were compared. Higher uncertainty in estimation, thus higher state covariance, was apparent in the asynchronous system compared to synchronous. This was explained in the paper by having less (new) information available by the asynchronous transmissions.

It is worth noting that for some applications and environments the cameras also do need to understand how the world coordinate system is in relation to them and to the other cameras. This can be achieved by *extrinsic calibration*, see earlier section 2.2.3 for details. By having a calibrated vision network much more detailed information can be provided for the application, e.g., any dimension or distance measures can be provided in absolute values, and with greater confidence as any geometrical dependences on the image plane can be taken into account.

4.3 Architecture Models for Fusion

It is very hard to separate the implemented architecture from the application it will be serving. Many models were introduced in the 1980s such as Joint Directors of Laboratories (JDL) which was later revised in [89], the UK intelligence cycle described by Shulsky [90], and the Boyd-model for decision support [91]. In the 1990s, models called Waterfall [92] and Dasarathy were introduced, shortly followed by Omnibus-model [93] which aimed to combine advantages and eliminate disavanges of each of the previously proposed models.

4.3.1 Boyd

A four stage cycle for decision-support in military information operations was introduced by Boyd in 1987 [91]. Due to resemblance to fusion systems, the decision-support inspired model has been adopted for sensor fusion. The model stages and connections are illustrated in Figure 4.3.

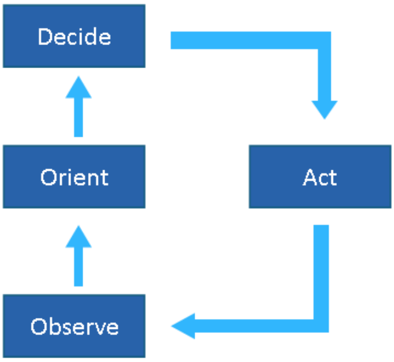


Figure 4.3: *The Boyd-model for data fusion process [91].*

The first stage of *observe* can be considered to include sensor preprocessing, such as prescreening and data allocation. *Orient* includes processes such as data alignment, association, tracking, and identification, followed by object context analysis and situation predictions. *Decide* stage is considered as making final decisions on resources and goals. Having a responsive system consisting of actuators that operate on the environment, the loop is closed with stage of *act*. The visibility of the processing tasks is not directly evident when examining the Boyd-model. Therefore it is hard to identify the separate sensor fusion tasks. The overall flow is though well represented.

4.3.2 Waterfall

A more precise model for portraying the lower level processing functions was introduced by the Waterfall-model in 1997 [92]. The six stages are depicted in Figure 4.4. The separation of processes is much better than in the Boyd-model, but the Waterfall-model does not explicitly include the feedback loop. Loop that represents the actuation on and observations from the environment.

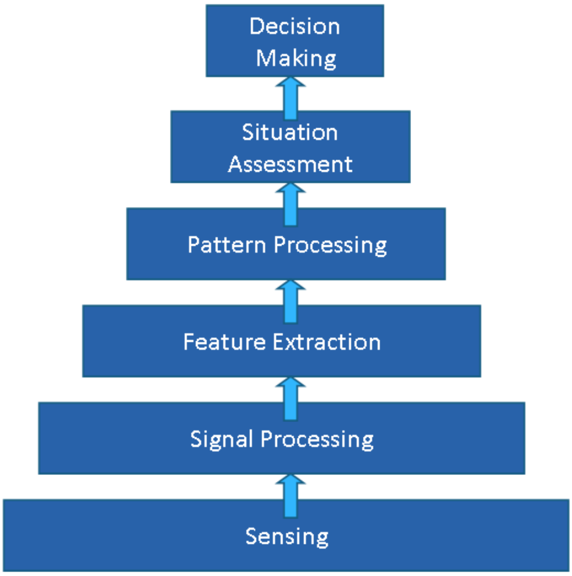


Figure 4.4: *The Waterfall-model for data fusion process [92].*

4.3.3 Omnibus

A unified model, the Omnibus-model, was presented in 2000 [93]. It aimed to capture the cyclical nature of the intelligence cycle and the Boyd-model, and to provide finer defitions for the processes like presented in the Waterfall-model. The Omnibus-model is depicted in Figure 4.5.

Fusion Level	Bandwidth	Performance	Advantages	Limitations
Data	high to very high	potentially optimal	possible to use physical models	high bandwidth restricts use to single platform systems
Features	moderate	good to high	high performance	difficult to select correct features
S Decisions	low	often good	bandwidth performance trade-off	compensation for correlated sources required
H Decisions	very low	depends on system	simplicity for larger systems	poor performance for small systems

Table 4.1: Omnibus: The advantages and disadvantages of fusion at different levels [93]. In general, the less bandwidth is required, the less is the expected performance.

The feedback flow was made explicit and loops within loops are acknowledged by the model, enabling the use of the model in defining both the system aim and its task objective. Related to the Omnibus-model, a short comparison of the challenges related to fusion at different levels was given by Bedworth and O'Brien [93], depicted here in Table 4.1.

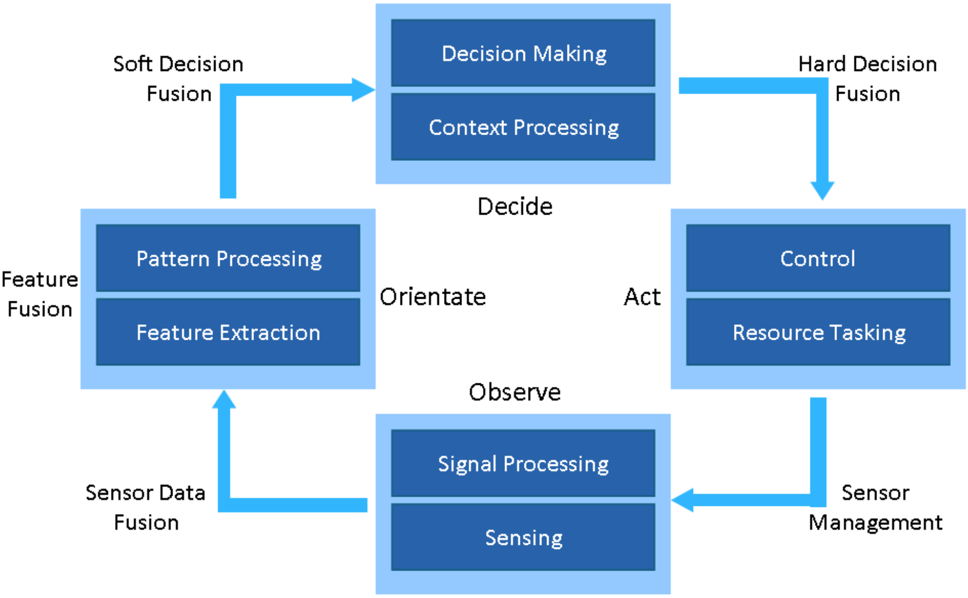


Figure 4.5: *The Omnibus-model for data fusion process [93].*

4.4 Part-I Remarks

This chapter dealt with fusion architectures, challenges in gathering data, and models for architectures that have been proposed for various applications purposes. A chosen fusion architecture does not necessarily limit the options for fusion level and fusion method. Some choices might become less practical though, such as gathering raw images to a central fusion unit easily requires large bandwidths and much computational power, especially as the number of cameras increases. In such a case it would be more flexible to either introduce local fusion units or to compute at each camera data of higher abstraction level, such as features or even class scores, before sending data to the central fusion unit. Selection of fusion architecture should therefore be driven by the accuracy requirements of the application, but one has to limit the architecture options according to the current available resources of and future plans for the vision network.

The Boyd architecture model does well in separating system tasks, identifying the fusion opportunities, and highlighting the feedback to and from the environment. This makes the Boyd-model a good reference for formulating a fusion framework that contains a more structured approach to fusion and that better supports fusion opportunities within vision networks.

Part-I of this thesis provided the necessary background information on vision networks and the three aspects of fusion that play an important role in these networks. The potential of vision networks was discussed through application examples, and the emerging trend of bringing them to private spaces through ambient intelligence solutions was highlighted.

The increased accuracy in scene understanding performed by the vision networks can only be achieved by intelligent fusion of data. The approach to sensor fusion was split into three separate aspects. Fusion architecture defines the manner in which multi-camera data is aggregated, fusion level defines the type of data to be combined, and fusion method defines how the data is combined.

In Part-II of this thesis the presented material is formulated into a common vision fusion framework, and various experiments in sensor fusion across four applications of vision networks are conducted.

Part II

Framework and Experiments

Chapter 5

Vision Fusion Framework

Like was discussed and explored in the chapters of Part-I, the design of a vision network that fully exploits the complementary nature of multi-sensor data by applying suitable fusion methods is a very hard challenge. Such a multi-faceted problem requires a systematic approach for streamlining the design process and for achieving best possible results. To this end, a framework is proposed in this chapter. The framework, referred to as vision fusion framework (VFF), aims to enable a generic and systematic approach to designing, building and using vision networks efficiently based on fusion-friendly approaches.

Section 5.1 starts this chapter by shortly summarizing the relevant research on architectures and frameworks. The structure and the flow of the proposed framework is illustrated in section 5.2 by expanding on how it connects to the common vision network architecture and the three aspects of fusion: architecture, level and method. The properties of vision networks that affect the creation and exploitation of fusion opportunities are discussed in section 5.3. The suitability of different choices in the three aspects of fusion are highlighted and a formulation is given in terms of design rules in sections 5.4 – 5.6. The chapter finishes in section 5.7 by gathering all the rules into one common design structure under the proposed framework.

The later chapters of this thesis will provide empirical studies for making conclusions and recommendations based on the design elicited by the vision fusion framework.

5.1 Related Research

Many of the existing architecture models were presented at the end of chapter 4. These models were originally designed with certain applications in mind, later they have been used to model fusion processes. Models such as Boyd [91], Waterfall [92], and Omnibus [93] were more thoroughly discussed and illustrated. Ombibus especially was an interesting model, because it first explicitly showed the cyclical nature of making observations and acting upon decisions made, and secondly provided a natural separation of processing tasks and thus levels of fusion. These two aspects: cyclical processing and task separation, are also a crucial part of the proposed VFF.

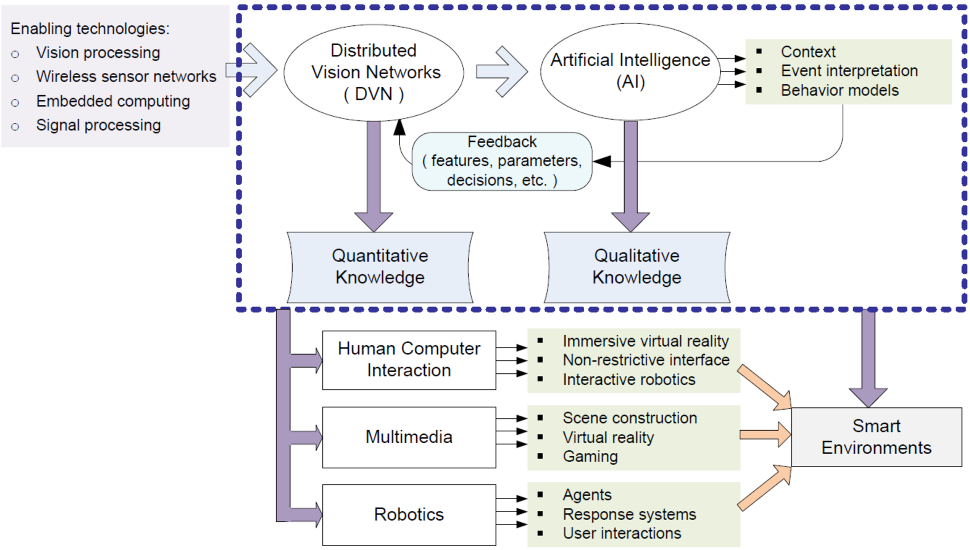


Figure 5.1: *A framework for distributed vision networks possessing reasoning capabilities enabling a wide range of applications. With permission from [94].*

Aghajan et al. introduced a model-based data fusion framework for human posture analysis in [94]. Like VFF, this framework uses a camera network for providing vision-based information from multiple sources. Particularly, their framework is focused on distributed processing, thus no raw image data is considered to be transmitted within the vision networks they address. VFF does not exclude transfer of images, it considers all data levels applicable. Their framework in connection to the enabling technologies and to the application prospects for smart environments is shown in Figure 5.1.

Aghajan's framework covers three dimensions of space, time and features; cameras placed with different viewpoints, each camera capturing data over time, and different subsets of features being combined. The framework differentiates the distributed vision processing from the high-level reasoning in terms of the type of knowledge provided: quantitative and qualitative, respectively. Their framework provides a technology driven outlook into a versatile set of application opportunities, but does not try to address the common structure of vision networks and the approaches for multi-view data fusion. This formalization will be given by the proposed VFF.

5.2 The Proposed Framework

Vision Fusion Framework (VFF) is largely based on the structure of the common vision networks. VFF covers the details from a part of a vision module to the connections between the system modules. VFF does not exhaustively cover all possible topics, connections and effects, but it does address the most common factors having high impact on fusion-friendly design of vision systems.

5.2.1 Definition

The proposed fusion framework is cyclical in nature. The decisions made by the vision system can be used to adapt the environment or to give feedback to the user, both which may alter the behavior of the person. This possible change in user location, activity or behavior is again detected by the multi-camera observations captured by the vision system.

Vision networks exploit the complementary data by aligning the observations both temporally and spatially, and by associating corresponding observations within and between views. The aligned and associated data can be directly processed by the three-layer data fusion approach, and the combined result is returned to vision network for later processing steps. A graphical representation of the proposed VFF is given in Figure 5.2.

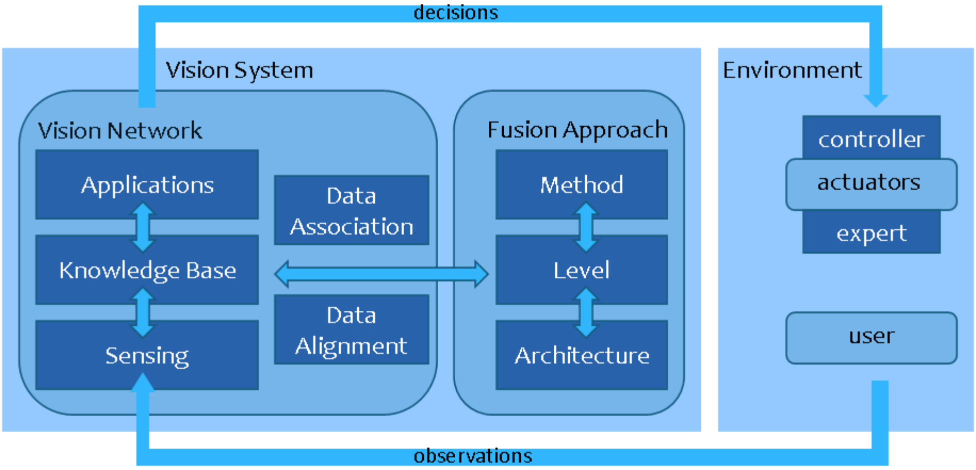


Figure 5.2: *Application-driven Vision Fusion Framework for the design of fusion-friendly vision systems.*

The main modules for vision networks, data fusion, and the environment are presented by VFF. Each of the modules may cover a variety of smaller details that themselves can have significant impact on overall system accuracy. The presented modules are there to provide a division of tasks that has been found most suitable, and thus hopefully enable a successful breakdown of system aspects that can be individually considered in system design. By following the division of tasks and addressing each of them within VFF, it is the aim of the framework to provide a systematic approach for building fusion-friendly vision networks.

5.2.2 Fusion Design

The goal of a fusion process can commonly be considered as one of two approaches. The first fusion approach is to provide most consistent estimation of a value with the highest joint certainty in the estimation. The second approach is to detect a phenomenon-of-interest (POI) that might be only visible in a view or a subset of the views.

Design Rule 1. *Apply fusion either to build consistency or to increase responsiveness to a phenomenon.*

The consistency approach provides a good solid accuracy, e.g. the percentage of positive classifications, over the entire range of situations arising in the observed environment. Providing state-of-the-art results in certain occasions does not necessarily make a system suitable for the challenges imposed by real world scenarios. In this approach, fusion is for building stability and certainty, not necessarily for providing continuously the highest accuracies. With this approach, a solid accuracy of positive fusion results is considered as the most important outcome of a successful fusion system.

The phenomenon-of-interest approach has the same foundation as does the consistency approach, that is to provide stable results through fusion. The difference with POI approach is in increased sensitivity for detecting certain properties of human behavior. For example, a delicate hand-gesture might be only visible in one of the observing views, and in this case a majority based scene analysis would not register the gesture, but POI approach would do so.

Theoretically speaking, POI approach appears to be a reasonable option. However the realities of the challenges that computer vision faces in tracking deformable objects in dynamically changing situations and environments make this approach much harder to implement robustly than it first might seem. The main issue is how the system can robustly separate a real POI instance from an outlier. For example, when detecting movement of the upper body within a ROI the camera that reports a major shift of the upper body, is actually been fooled by the movement of the person standing behind the observed person, because they both happen to fall inside the ROI covering the observed person.

Both fusion approaches should include a mechanism for detecting outliers, or at least compute an estimate in such a manner that is not heavily influenced by outliers. For example, one option is to use a median-operation instead of a mean-operation for computing an estimate from multiple values. Outlier elimination is much more difficult of a task for the POI approach, as sometimes only one camera might provide a contradicting observation compared to the other views, and this contradiction might be due to the real POI or the outlier. Therefore the POI approach has one major issue to find a solution for: How strongly a single camera can override others without having been supported by other observation(s) of similar type. Either some support will be required, or certain valid assumptions and checks will have to be performed by the vision algorithms.

As depicted in Figure 5.2, all the choices made in each of the fusion design modules influence each other. The selected fusion architecture, such as fully centralized or fully distributed, will have impact on which fusion level, such as feature or decision-level, can be expected to provide the best accuracy, commonly the environment having an impact on the choice too. According to the structure presented in VFF, the following order of fusion design procedures is most often applied:

1. define fusion approach
2. define fusion architecture(s)
3. define fusion data level(s)
4. define fusion method(s)

The choice on fusion architecture is often limited or even defined by the physical implementation of the camera network. The physical implementation has to adhere to the limitations in resources and location dimensions. The required compromises will be reflected in the fusion architecture options. Similar limitations may apply, e.g. to the amount of transferred data, thus limiting the choices on fusion level. The choice on fusion method often comes down to providing tractable, fusion approach-specific results while exploiting a prior information, if such is available.

In general, the greatest potential of fusion is in providing stability of processing in varying conditions, simultaneously increasing certainty in the results achieved. The second potential of fusion is to enable detection of phenomenon that might otherwise be missed. In the next sections each of the vision system modules is broken down, and the effect of the major properties to fusion opportunities is discussed and formulated into a design rule structure.

5.3 Effects from the Vision Network

As illustrated in Figure 5.3, there are a multitude of topics that a VFF needs to consider from both software and hardware side of a vision network. More precisely, the topics that arise from the areas of sensing, vision and providing services.

5.3.1 Sensing Tools

The tools responsible for sensing, and thus creating observations, in a common vision network include cameras for capturing video, networking for transferring the captured data, and processing units for all the data handling operations.

Cameras
When considering cameras, three aspects have great influence to system design: resolution in pixels (RES), frame-rate (FPS) and field-of-view (FOV).

Resolution is defined as the number of individual points on the image sensor in both vertical and horizontal direction in which scene emitted light is sampled. Thus resolution is a limiting factor on how small of an object, area, or movement can be detected on the image. *Frame-rate* tells us how many images, or frames, the camera captures per a second. Therefore the fastest detectable change in e.g. size, movement, appearance, or any other property is limited by the frame-rate. Hardware specifications do limit the range for both resolution and frame-rate, but commonly these operating settings are defined by the software within the applicable range. *Field-of-view*, which almost always is required as wide as possible, is again very limited due to the available camera lens and image sensor hardware. FOV defines the largest detectable object, area, or movement; thus the opposite of limitations given by RES.

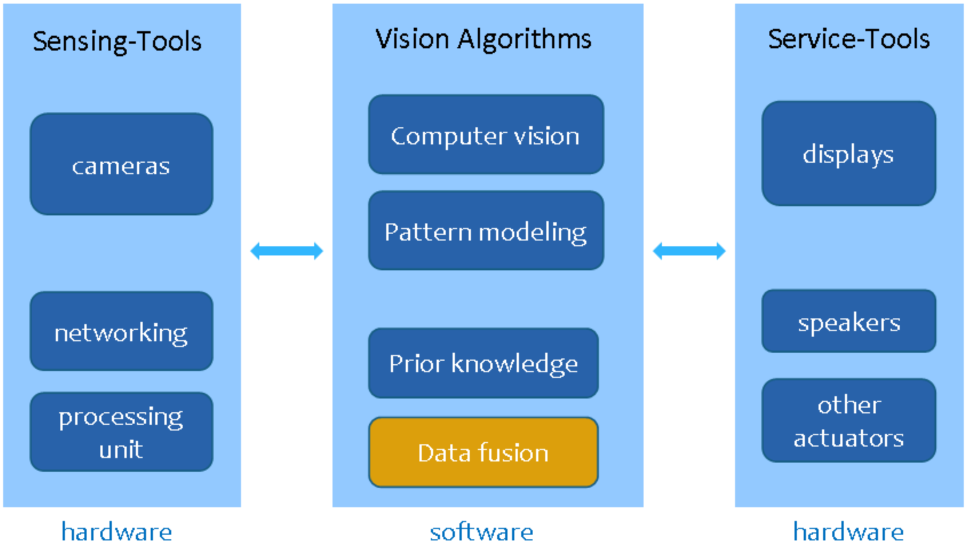


Figure 5.3: *Illustration of the Modules of Vision Networks, each having impact to fusion processes, thus part of consideration in the VFF.*

Networking

Networking similarly has limitations from both hardware and software tools: physical transport medium and transport protocols.

The *physical transport medium*, such as USB or fiber optics, present limitations to the achievable data-rates. This can limit directly the fastest change detectable by the vision network. *Transport protocols*, such as TCP, IP, or RTP, are required for transferring the data robustly and in timely manner. Each protocol has its own characteristics in dealing with packet losses and providing synchronized delivery. Synchronization can be enabled by deploying timing protocols, such as Network Timing Protocol (NTP). NTP helps designers in setting maximum shifts in time allowed between different data sources.

Processing Unit

The unit responsible of computations, referred to here as processing unit, has especially three properties that define its performance: clock frequency, processor architecture, and circuit-board architecture.

Clock frequency defines the sampling rate of the processor, the memory-elements, and the data transferring buses. *Processor architecture* includes the number and type of processor cores. *Board architecture* defines the type of interfaces, memory-blocks, and computation parallelism existing on the processing unit. All three properties influence the computing power and thus the performance available for the vision network.

Effects on Fusion

By understanding what to look for and what to address with sensing tools, VFF aims to enable the direct comparison and thus direct fusion of multi-view data. Enabling is achieved by eliminating the need of additional steps in the latter processing steps, by systematic assessment of the sensing tools.

By using identical cameras for the same function, such as person tracking, a stability of data that is critical to optimal fusion, is achieved. For securing this stable foundation for imaging, it is advised for cameras to have similar resolution, frame-rate, and field-of-view. It is common, especially in image pre-processing performed off-line, to alter the resolution and the frame-rate of a camera. However for this purpose, it is recommendable to have high resolution and high frame-rate. By downsampling from better quality the result will be of higher fidelity, than with upsampling from inferior settings. This may create unnecessary need for upgraded cameras, as the weakest link, the camera with the slowest frame-rate and lowest resolution, will define the performance of the entire vision network. Additionally any compensation performed by software for the inequalities between the cameras takes away computational resources from the vision network.

With a stable and uniform basis the algorithms can be mapped to all hardware in a straightforward manner. No alteration on algorithm parameters or assumed constants is needed. If fixing these parameters, such as the number of size-iterations of a template-model or range limiters for a search window, no further analysis is required.

Design Rule 2. *Apply uniform design over all sensing tools, thus creating data that is stable and directly comparable.*

Having observations from the same time instant is crucial for fully exploiting fusion potential. Options for data fusion are best supported by having access to synchronized data. For example, to successfully use feature vector concatenation, the separate feature vectors must be available at the same frame-rate [55]. Latency in the start of a session can be avoided by global triggering methods. Shifts in data streams between multiple cameras are avoidable by introducing time-stamping of data and regular device-clock updates. Data streams should be transferred through same physical medium for all the cameras, thus providing similar probabilities for data congestion and traffic constraints.

Design Rule 3. *Apply a uniform transmission on multi-view data, thus ensuring synchronous delivery that enables maximal fusion opportunities.*

By providing comparable data, in a synchronized manner, with computational resources meeting the demands for processing, will data fusion mechanisms have full effect on the commonly heavy amount of multi-view data.

5.3.2 Service Tools

Vision networks provide scene understanding through visual analysis. In most cases reported in this thesis, it is related to people and their activities. As the smart environment understands what people are up to, it can better react to and accommodate the user. This feedback can be provided by a multitude of actuators, such as displays, speakers, and other mechanical devices.

Actuators
The design of the form of an actuator has to address at least the three aspects: size, shape, and integration to existing surroundings. An actuator that conveys application information must meet specifications for operating frequency, latency, and accuracy/resolution. Displays can be provided by various technologies, such as screens, projectors, holographic projections and electronic paper. The content can be rendered in monochrome, in color, or with depth. Speakers can be integrated to existing devices, in a form of a sound bar, or distributed with several small monitors. Sounds reproduction can be based on mono, stereo, or surround sound.

Effects on Fusion
All above mentioned specifics on physical form and rendering quality are driven by the application requirements; what needs to be shown and in which manner for the best desired effect. Visual rendering of feedback or content might introduce new dynamic artifacts into the scene. Fusion of multiple views can cope with these artifacts, e.g., by exploiting the understanding of scene geometry given by the multiple views for removing the correlating effect of the artifacts, Rule 4. Artifacts may include, e.g., a blinking screen that is used to provide user-support. By fusion it is sometimes possible to isolate and ignore this correlating error source.

Design Rule 4. *Prepare for the effects that user-supporting rendering has on the performance of the user-observing vision system.*

Rendering techniques often re-sample the given data, thus it is not necessary to provide them with high fidelity input data. Rendering can interpolate between given samples and thus estimate the behavior of the data between the samples. Therefore, the sampling accuracy and sampling rate demands on low-level vision processing can be relaxed. This frees up computation resources to be used elsewhere, most importantly on fusion algorithms.

5.3.3 Vision Algorithms

Two especially important groups of algorithms in a vision network are computer vision and machine learning algorithms. Computer vision (CV) algorithms process the raw captured images in our studies for detection and representation of people. Depending on the application, machine learning (ML) has been used either for modeling of activities or goals and intentions.

Effects on Fusion
For achieving high performance and accuracy one important factor is generic view-independent vision processing. In other words, it is practical and tractable to use the same vision algorithms across all cameras, at least between the cameras that perform the same exact function, such as person-identification or tracking. In the context of this thesis, the ultimate target of computer vision is to provide features that are maximally robust against changes in orientation and scale of the observed person. With such an approach, data fusion is possible at each system level, with credible expectation for high accuracy given the controlled manner in which the features are computed.

Design Rule 5. *Apply features with lowest dependency on person scale and orientation, thus increasing both stability of data and potential for fusion.*

Given uniform processing, any violations of view-independency can be detected to a certain extent, which enables the vision system to count for such situations in data fusion. In addition, machine learning offers powerful tools, e.g., for finding the true trend in feature data, and helping fusion in this manner in coping with the inherent variability that exists in all multi-sensory data.

5.4 Effects of the Fusion Architecture

When defining the underlying fusion structure in which fusion operates, it is beneficial to study the suitability of the three proposed fusion architectures in chapter 4. Additionally, some practical issues should not be forgotten, as they can have indirect effects on the fusion process.

5.4.1 Practical Issues

The major drawback of the fully centralized fusion architecture (FCFA) is that it is not a fully flexible architecture, that could be changed to use an arbitrary number of sensors. Certainly, some additional sensors can be added to FCFA, but the computational load and amount of data-traffic faced by the central unit can increase to unmanageable levels. Thus, depending on the central resources available and the type of data being transmitted, FCFA typically provides rather limited flexibility on increasing sensor population. The same concerns on centralized architecture of visual sensor networks w.r.t. the scalability and high costs when deploying many cameras were mentioned by Karakaya et al. in [95].

The fully distributed fusion architecture (FDFA) uses a subset of all the sensory information in each of the local fusion processes, while consulting the other local fusion centers about their results. This approach offers more flexibility for adding additional sensors, due to the distributed nature of computations. On the other hand, the more complicated communication paths and synchronization issues make this approach more problematic than the FCFA. The properties of scalability, robustness against sensor failures, and modularity were stated by Durrant-Whyte and Stevens as the most attractive properties of the decentralized fusion of data [96]. The hierarchical fusion architecture (HFA) has similar issues with communication as does FDFA, but it does offer added robustness against sensor and fusion failures, due to its hierarchical nature.

5.4.2 Suitability

The selection of the fusion architecture to be deployed is always affected by two basic factors, when dealing with sensors:

- *sensor failure*: a sensor fails to provide data
- *sensor noise*: observations given by a sensor are affected by noise

The suitability of each of the three proposed architectures is studied for four different scenarios. In case of sensor failure, a failure detection mechanism can be deployed, defined as scenario $FailureDet$, or not deployed, $noFailureDet$. In case of sensor noise, a noise compensation mechanism can be optionally used, defined respectively as scenarios $NoiseComp$ and $noNoiseComp$. Having such mechanisms has a great impact on the predictable accuracy of an architecture. Estimates on the suitability of each architecture-mechanism configuration are presented in Table 5.1.

Table 5.1: Fusion Architectures ranked by a comparative suitability.

scenario	noFailureDet	FailureDet	noNoiseComp	NoiseComp
FCFA	poor	best	poor	best
FDFA	good	good	good	good
HFA	best	poor	best	poor

Considering the cases in which a mechanism is in place for coping with sensor failure and sensor noise, FCFA is expected to perform best. This superiority is due to one factor. Because central unit gathers all the sensor data, it has access to all observations; good, bad and missing. This is expected to provide ideal situation for the fusion center for detecting outliers, which are observations numerically distant from others. One applicable iterative method for detecting outliers that has been widely used is Random Sample Consensus (RANSAC) [97]. For the same reason of having access to a wider set of data of the same level than HFA, also FDFA is expected to outperform HFA. The following guidelines can help in the selection of the fusion architecture:

Design Rule 6. *Apply fully centralized architecture, if the main goal is to have maximum fusion potential and cases of sensor failure are handled.*

Design Rule 7. *Apply other than centralized architectures only if prior information or initial data analysis supports it.*

In case of having no mechanism for coping with sensor failure and sensor noise, the same wide access to data that helped FCFA and FDFA to outperform HFA, is now their downfall. Without data handling mechanisms the effect of erroneous data, within otherwise good observations, can have major negative impact to the overall fusion result. Having no detection and response to sensor noise or sensor failure, FCFA is most negatively affected. This provides a clear motivation for Rule 6. In case of FDFA the negative effect can be expected to be slightly less than with FCFA, due to FDFAs distributed nature mitigating the effect.

For supporting another architecture than centralized, prior information, e.g., on the camera hierarchy in the sense of fixed distances to the observed person, can be considered as valid information to group certain cameras together, this way giving rise to HFA. A similar tendency between a group of cameras can be, e.g., detected in the initial data analysis of the multi-view feature-data, see Rule 7.

It is not difficult to design and implement the mechanisms for sensor failure and noise. However, if these mechanisms are not in place, the stability and performance of the selected fusion architecture can be seriously degraded. In overall, the choice of a fusion architecture has to be supported by the available mechanisms and by any prior or initial data that is or can be made available.

5.5 Effects of the Fusion Level

Considering the previously presented four fusion-levels: images, features, class-scores and class-decisions, many level-combinations for exploiting fusion exist. Each level offers a different abstract viewpoint to the original sensory data.

5.5.1 Abstraction of Data

A higher fusion level corresponds to fusion of more abstract data. Raw data has been transformed into another type of data having more qualitative information in it. The effect of noise has favorably been canceled, and the qualitative properties of information are of high standard and possessing discriminative power.

It has been argued that by having access to all of the raw data, the best approach to data fusion is achievable [35]. This is because the level of detail in the raw data is highest. On the other hand, also the corruption due to noise is also highest, making the choice of appropriate level more difficult, than it originally seems.

Design Rule 8. *Apply fusion at the lowest single fusion-level with consideration for the tradeoffs between handling of noise and quality of information.*

Do not expect the level of accuracy with a single fusion-level to carry onto other environments. In the experiments sections it will be shown that depending on the tracked object and the environment under observations, it is different fusion levels that based on the same fusion methods, can provide the best accuracy.

5.5.2 Level Configurations

In all of the proposed architectures, such as FCFA, fusion can be implemented at any combination of levels or at any individual level. With FDFA, fusion at any level is achievable once agreed on between the fusion centers; the same applies to HFA. Thus, the exploitable fusion level or combination of levels are not limited by the chosen architecture.

Considering the entire chain of data from sensing to decision, multiple options for data fusion exist. The level of fusion and the number of levels involved can all be changed. In [35] Dasarathy illustrated the full potential of fusion based on his five input-output type characterizations for data fusion, presented here in Figure 5.4. Based on the four levels of data proposed in this thesis, all the options for type of fusion based on data are listed in Table 5.2, with remarks on the assumptions and requirements for a particular option.

Of all the 15 fusion-configurations presented in Table 5.2, there is no single configuration that could be expected to always perform better than the rest. Regardless, some discussion topics do arise. Having limited resources and flexibility, any of the single fusion-configurations should provide increased certainty in observations. For image-level fusion, calibration of the camera network for both intrinsic and extrinsic properties is required. Calibration is a laborious task, sensitive to mistakes and easily broken, which makes fusion at image-level less practical. Calibration is also required for fusion at the feature-level for some algorithms, such as the feature-point based depth estimators using temporal images [98]. In general, the other single fusion cases require less initial work, a mechanism for data association and alignment is though often required, but the ranking between the configurations depends on the environment and application. This will be shown in the experiments presented in sections 6-9.

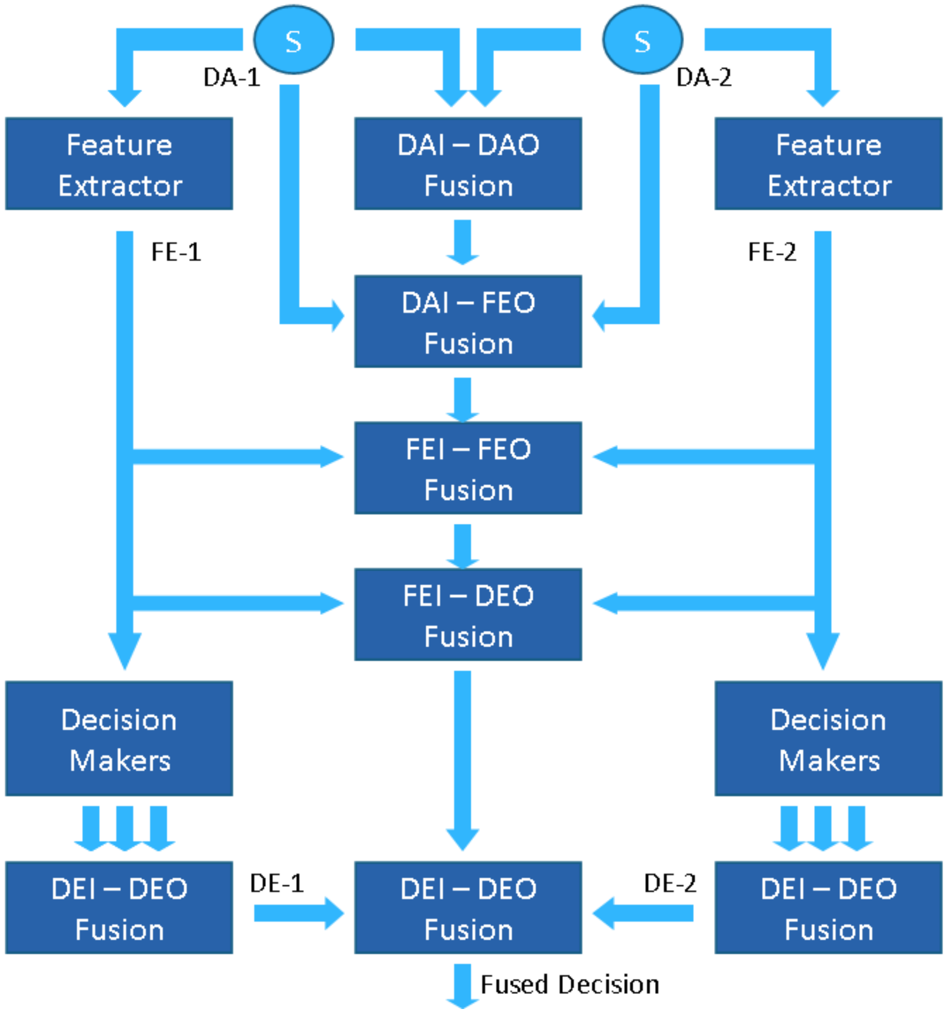


Figure 5.4: *Illustration of the fusion potential for Dasarathys I/O-based fusion characterization. DA stands for data, FE for feature, and DE for decision. [35]*

Excluding the 1x-fusion cases, all the rest of the fusion-configurations exploit some level of exhaustive processing, in which multiple fusion processes take place as data passes through the vision system. The quadruple 4x-fusion, in which fusion is performed at each fusion level, exploits all three of the following techniques:

1. *feature extractor suite*: the same feature is computed by different algorithms from the same source.
2. *classifier suite*: a class is inferred by a number of classifiers based on the same features.

3. *temporal accumulation of data*: the same feature is observed over a sequence, from which an estimate for the feature is computed.

Table 5.2: Fusion opportunities based on the fusion level: fusion is marked by 'x'. I stands for image, F for feature, CS for class-score and CL for class-label.

scenario	I	F	CS	CL	remarks
1x-Fusion					
1	x				images assumed registerable
2		x			no requirements
3			x		..
4				x	..
2x-Fusion					
1	x		x		$Note^1$: assumed classifier-suite
2	x			x	$Note^1$
3		x	x		$Note^1$
4		x		x	$Note^1$
5	x	x			$Note^2$: assumed feature-extractor-suite
6			x	x	$Note^3$: assumed temporal-fusion
3x-Fusion					
1	x	x	x		$Note^1$, $Note^2$
2	x	x		x	$Note^1$, $Note^2$
3	x		x	x	$Note^1$, $Note^3$
4		x	x	x	$Note^1$, $Note^3$
4x-Fusion					
1	x	x	x	x	$Note^1$, $Note^2$, $Note^3$

It is most probable that no single fusion level configuration can guarantee optimum accuracy in all environments and applications. Each technique above presents an opportunity to fully exploit the fusion potential. The idea is to not narrow down the options, if no analysis or expertise has been provided to back up such a procedure. For example, Mansoorizadeh and Charkari showed that a hybrid fusion of both feature and decision-level fusion results, can provide better accuracy than unimodal classification, or fusion of either features or decisions [53]. They experimented on the use of face and speech information for emotion recognition.

Design Rule 9. *Apply fusion both across fusion levels and on estimates within levels to provide more stable accuracy, if lacking sufficient domain knowledge.*

On the contrary, a suite of different techniques performing the same task can be applied. The estimate is expected to have captured the essential behavior of the value under fusion, because estimates with different assumptions behind them are combined. The effects of the assumptions of each technique are expected to cancel each other out. For canceling to happen, some understanding of the behavior of the techniques is required, as otherwise the effects may actually re-enforce each other.

5.6 Effects of the Fusion Method

The type of data to be combined largely limits the possible fusion methods to be applied. Having prior information of the observed object or environment, can help in initializing and directing the method. Additionally, the method might have to be able to learn on-line some facets of the observed process. It is beneficial and sometimes even necessary, for the results to be tractable by a (human) expert, so that the method can be adjusted through a feedback loop. This is achievable, e.g., by so called Elucidative Fusion Systems (EFS) proposed by Dasarathy in [99]. EFS aims to create transparency to the contributions of each sensor involved in a fusion process. In EFS the importance of a sensor, the relative influence on the fused result, is made explicitly visible.

5.6.1 Relative Influence of Sensors

In order for relative influence to have an impact, the fusion method has to provide a tool for weighting the different sensors based on their importance, Rule 10. The weighting can be given by prior information, or can be learned on-line as the system is running. For the best accuracy, a combined effort of defining initial settings by the prior information and updating them through on-line learning should be opted for.

Design Rule 10. *Apply to the fusion method preferably an adaptive mechanism of relative influence.*

The challenge in conforming to the expected properties of the observed phenomenon and adapting to any changes observed is that, if done poorly, fusion accuracy deteriorates and may end up giving results worse than with majority voting of sensors, Rule 11. For example, by assuming a close proximity camera to provide the best posture estimate due to full and close visibility to a person, may not always hold. The proximity of other people and moving objects on close-by tabletops can make this originally best view the most difficult one to process by vision algorithms. If not correctly adapting to the changed situation of camera preferences, the positive results will diminish and a catastrophic fusion may take place.

Design Rule 11. *The fusion method has to control the quality of prior information, on-line learning criteria, and expert feedback in order to prevent poor fusion.*

Quality assessment is enabled by tools of quality control. One possible approach is to use a suite of fusion methods to provide a benchmark of average fusion accuracy. Equal weights are used for each fusion method, and estimates from each method are combined into suite estimate by finding the peak of the estimate distribution. The quality of each method is then defined by comparing the estimate provided by the method to the suite estimate.

One should notice that the suite estimate is not expected to provide the best accuracy at all times, Rule 12. A single method might be more responsive to certain changes in environment conditions, making it the best method for that given time instant. In contrast, the method suite will adapt more conservatively, because each of the contributing methods have their own characteristics, drawbacks, and benefits.

Design Rule 12. *A fusion method suite can provide consistently good fusion results, especially when applying correlating fusion methods with non-correlating errors.*

The selection of fusion methods provides in an ideal case a set of techniques whose fusion results correlate, and the errors they make are uncorrelated. Most often this correlation of results will increase the accuracy, and the non-correlation of errors diminishes the overall system error. If each of the fusion methods has the same weakness for a certain change in data, the fusion results will be negatively affected. Having methods with complementary reactions in terms of the errors they make, helps the suite to cope with various situations.

5.7 Conclusions

As it has been noted, the most suitable data fusion system depends on various vision network aspects. Aspects covering the environment and the observed people, the physical camera-system, and the purpose of the application all have an impact. The fusion itself can be defined in four parts: fusion approach, fusion architecture, fusion level and fusion method. Each part has its own properties, all of them influencing the overall fusion accuracy.

By defining the proposed framework, the aim is to help in making a systematic analysis of the vision network and the potential for fusion within. Analysis highlights the common pitfalls and directs the efforts for most efficient workflow and positive fusion results.

5.7.1 Rules of Design

Based on the analysis on different parts of both the vision system and the environmental factors, important aspects for creating and exploiting fusion potential were raised in this chapter. The major modules were presented in the proposed vision fusion framework, and for each module design rules were formulated. The fusion framework was illustrated in Figure 5.2. Correspondingly, the rules for driving fusion-friendly design within the framework were defined as follows.

Fusion Approach
Design Rule 1: Apply fusion either to build consistency or to increase responsiveness to a phenomenon.

Vision Network
Design Rule 2: Apply uniform design over all sensing tools, thus creating data that is stable and directly comparable.

Design Rule 3: Apply a uniform transmission on multi-view data, thus ensuring synchronous delivery that enables maximal fusion opportunities.

Design Rule 4: Prepare for the effects that user-supporting rendering has on the performance of the user-observing vision system.

Design Rule 5: Apply features with lowest dependency on person scale and orientation, thus increasing both stability of data and potential for fusion.

Fusion Architecture
Design Rule 6: Apply fully centralized architecture, if the main goal is to have maximum fusion potential and cases of sensor failure are handled.

Design Rule 7: Apply other than centralized architectures only if prior information or initial data analysis supports it.

Fusion Level

Design Rule 8: Apply fusion at the lowest single fusion-level with consideration for the tradeoffs between handling of noise and quality of information.

Design Rule 9: Apply fusion both across fusion levels and on estimates within levels to provide more stable accuracy, if lacking sufficient domain knowledge.

Fusion Method

Design Rule 10: Apply to the fusion method preferably an adaptive mechanism of relative influence.

Design Rule 11: The fusion method has to control the quality of prior information, on-line learning criteria, and expert feedback in order to prevent poor fusion.

Design Rule 12: A fusion method suite can provide consistently good fusion results, especially when applying correlating fusion methods with non-correlating errors.

In the following chapters 6 – 9, experiments for data fusion will be presented. Each experiment will be tackling a different application domain. Depending on the application, different fusion aspects and questions will be explored. In the discussion-section of each of these chapters, the findings of the experiments are connected back to the VFF.

Chapter 6

Remote Visual Communication

A common task of a vision network is to detect and track a person moving within an environment. An application designed for visual communication between remote sites is presented in this chapter [1]. The application requires the detection and localization of a person within the designated area. To this end, a set of distributed cameras is used as sensors for reconstructing the shape of a person within the volume-of-interest (VOI).

Section 6.1 starts the chapter by providing background for the visual communication application and common related methodologies for occupancy testing. This is followed by a description of the methodologies used in the proposed vision network in section 6.2. The experiments on implementing such an application and vision network are presented in section 6.3.

Application experiment is followed by two fusion experiments on shape reconstruction in section 6.4. The first experiment studies the effects of camera configuration; that is where and how the cameras are setup for the shape estimation. The second experiment examines the accuracy and behavior of five proposed occupancy fusion methods on two noise conditions: imaging sensor noise and camera calibration noise. This chapter concludes in section 6.5 with a discussion on the major findings in relation to the vision fusion framework.

[1] This chapter is (partly) based on:
Määttä, T., Aghajan, H., Härmä, A. (2009). *Home-to-home communication using 3D shadows.* 2nd International ICST Conference on Immersive Telecommunications.

6.1 Related Work

A system for persistent and ambient visual communication based on capture, transmission, and rendering of 3D shadow representations of users is presented. The shape of a person is captured using a distributed camera array, compressed, and transmitted over the network. In the receiving end the shape is projected as a shadow on a surface using a lighting device. The 3D representation of the shape makes it possible to control the 2D visualization at the receiving end in many interesting ways. For example, when controlled by tracking of the observing user the shadow may create a visual illusion of a 3D shape on the wall.

In video conference applications it is usually desired to transmit a high-quality video image to imitate face-to-face communication. Some of the most important factors for the face-to-face experience are the preservation of facial expressions and eye contact. In a home video telephony system installed in a PC or a TV these can be typically only preserved when the user is seated close to and facing the terminal device. The *terminal-centricity* of video communication limits the user from doing other things while having the call and causes fatigue in long communication sessions. Consequently, video calls are often short in duration. Due to the flat-rate pricing model of home broadband communication it has become economically feasible to have visual home-to-home connections always open, enabling applications to support *persistent* presence.

The requirement for the face-to-face quality can be relaxed, if the goal is to build a visual communication device supporting primarily the social connectedness and awareness of the other side [100]. Understanding the activities and status of others is achieved through social connections and communications. The idea of the system introduced is to convey the awareness not by direct interaction or sharing the same physical space, but by mediating it through an open-ended, or *persistent* visual communication channel. Such communication systems have been earlier studied in the collaborative working environments and it has been found that they increase the awareness and support other forms of communication [101]. Users of Instant Messaging (IM) are already familiar with indicating their presence and keeping the link on for longer periods of time. Persistency and constantly low bit-rate of data when compared to videoconferencing are the main characteristics of the system discussed.

If the goal is not to reproduce the physically accurate visual representation of a remote person there are plenty of alternatives. The remote person can be represented in stylized form as an avatar which may take any shape. It is also possible to map the visual representation to some abstract or symbolic representations such as geometric patterns, movements, colors, or even sounds. Which kind of visual information is transmitted from their home to the other side could be controlled by the users themselves. Based on user's preferences to different people at different times the user could have several options for communicating, options on devices (projector, screen) and presentations (shadow, avatar) to be used. Realistic reconstruction is required by application, but point-to-point accuracy is not necessary for the expected effect. The visual requirements from the application are:

- lack of visual artifacts: no additional blobs
- smooth representation: few jagged edges
- geometrically correct visualization: correct anthropometric proportions

In the proposed system the focus is on using a silhouette representation, which is projected on a surface, such that it looks like a shadow of a person placed between the light source and the surface. Shadow or silhouette representations of remote people have been proposed earlier in [102, 103]. The shadow representation preserves many of the subtle non-verbal signals about our mood, attitudes, and feelings. Therefore, people living far apart from each other might still like to maintain the feeling of living close to each other, like neigbors who see each other daily through the windows of their homes. A spatial system for bringing distant homes together by merging the two spaces has also been introduced in [104]. The benefits of shadow representation are:

- low bit-rate
- low rendering complexity
- more privacy while still conveying presence

6.1.1 Scene Occupancy

Methods for capturing the geometry and occupancy of an environment commonly rely on occupancy maps and certainty grids [36]. Both occupancy maps and certainty maps are based on dividing the observed space into regular cells. Given a two-dimensional floor, the observed cells are squares, see Figure 6.1. For a three-dimensional space the cells are defined as cubes. Based on the observation performed by the multiple sensors the occupancy of the cell is determined. Cell is occupied if the fused sensor data agrees that an object/obstacle relies within that particular cell.

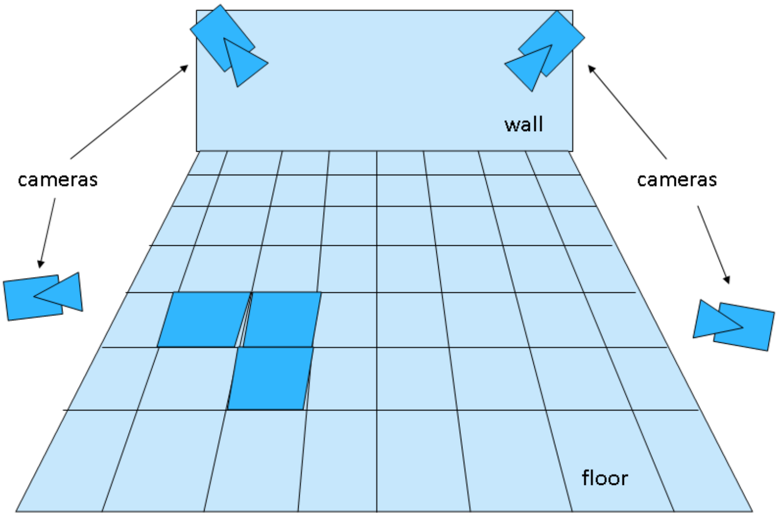


Figure 6.1: *A room observed by multiple cameras, floor level divided into regular cells for generating an occupancy map [105].*

Occupancy Map

Occupancy maps contain a binary value, either a zero or one, for each cell indicating if the cell is occupied by an obstacle or not. A map can be generated in two different manners:

1. by finding the objects
2. by finding the free unoccupied space

By assuming the entire environment is free, the *objects* that are found to exist within the environment are used to define the corresponding locations on the occupancy map as occupied. The locations of the objects can be found by e.g. object triangulation. Whereas by assuming the entire environment as occupied space, the aim becomes to find the areas with free space and mark these locations as not occupied. Because no object detection/tracking is needed, the find-free-space approach is less complex and more robust. Additionally in case of a sensor failure, less area will be defined as free space, which does no affect e.g. the safety in obstacle avoidance.

Hoover et al. built camera-specific occupancy maps in [105] by first taking a difference image, which was then transformed to map space, having done the required calibration for the vision network as a preliminary step. The occupancy map was initialized as fully occupied. If any of the camera-specific occupancy tests declared a cell free, that cell was declared as empty in the fused occupancy map.

Certainty Map

Certainty maps are similar to occupancy maps with one major distinction; the value of a cell is not anymore a binary value, but a value representing the probability that the cell is indeed occupied by an object/obstacle [106]. The value of certainty can be achieved by various value fusion methods. A common approach to assigning the certainty to a cell is performed based on Bayesian inference by assuming that conditional independence exists between sensors and that there is no preference on cell state.

For the perception and navigation of a mobile robot, certainty maps were reviewed in [107]. The collection of single sensor readings, referred to as sensor views, taken from a single robot location formed the local sensor map. For each sensor type there was a separate sensor map, which fused together yielded the robot view. The robot view thus contains all the sensor information of the robot's surroundings from a single location. The global map of the environment was composed by fusing the robot views taken at different locations as the robot travels through the surroundings. Two different modalities, sonar and scan-line stereo, were used as sensors. The grid estimates from both modalities were combined by the *independent opinion pool* method [108]: each cell was multiplied by the corresponding cell in the other sensors grid, and the resulting cell-value was multiplied by an appropriate normalizing constant.

The communication application discussed in this chapter requires the space to be labeled as either occupied or free. Based on these hard decisions can the system estimate the shape and location, and use these directly in rendering. Therefore, an occupancy map is needed for the application.

6.2 Proposed Vision System

The proposed system combines multi-video processing and free-viewpoint visualization to achieve peripheral awareness and persistent connectedness to a close one who will be portrayed as a shadow-like character. As the user's 3D shape is captured, walking around the 3D shape is made possible thus enhancing the effect of the other one being here, at ones home. The silhouettes of the person are extracted in one location with multiple cameras observing the location from different viewpoints; the silhouettes are combined into a 3D silhouette. This 3D silhouette or derivation of it is coded into a more compact form, transmitted over the network and rendered in another location interactively based on the observing user's position. The flow of data in one home is presented in Figure 6.2.

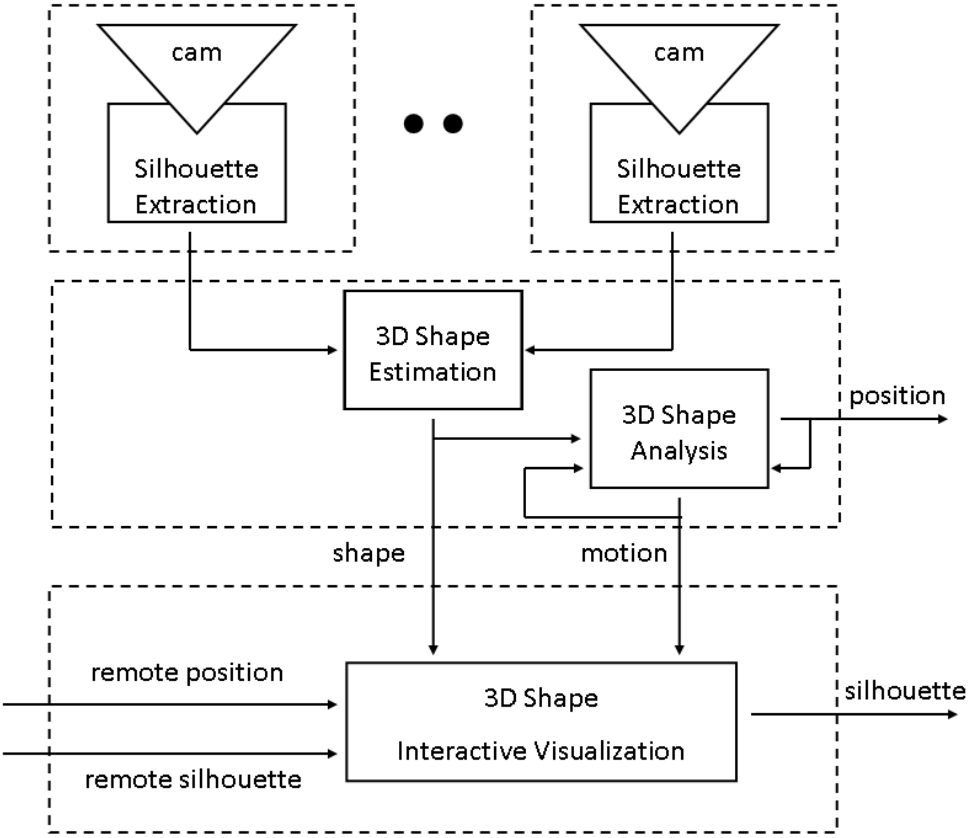


__Figure 6.2:__ Data processing and flow presented for one end of the vision system. Both application ends will have an identical system.

6.2.1 Person Capture: Silhouette Extraction

Human motion capture is to understand human behavior through detecting, tracking and recognizing user's actions. Capture can be performed with only a single view [109], with a pair of cameras [110] or by multiple cameras to achieve even higher robustness. Capture systems are either based on special markers or special suits to mark important spots on the user or they are based solely on the video content.

Multi-view, video-based user capture was chosen for reasons inherent to the system designed for the home environment. User capture has to be done unobtrusively to offer ease on long-term use of the daily communication application. In addition to improved robustness a larger area can be covered with multi-view thus giving the user more freedom to move.

The proposed system is designed to detect where the user moves and how fast. The multi-camera system provides video feeds from different viewpoints of the same scene. These feeds are processed locally to a much simpler form by segmenting the user from the background. This process of binary segmentation is hereafter called silhouette extraction. The approach used in our system is based on the following assumptions. The scene is situated indoor and is decently illuminated by ordinary lighting. The cameras are fixed, they do not support zooming, panning or tilting therefore keeping the scene stationary. Only the user will generate movement in the scene. The silhouette extraction is based on using adaptive statistical pixel model to describe the recent history of color at each observed pixel by normal distributions of luminance and green and blue chrominance components [111]. The binary silhouette image is the result of subtracting the model from the video image and thresholding this difference image w.r.t. a predefined threshold.

6.2.2 Multi-View Geometry: Camera Calibration

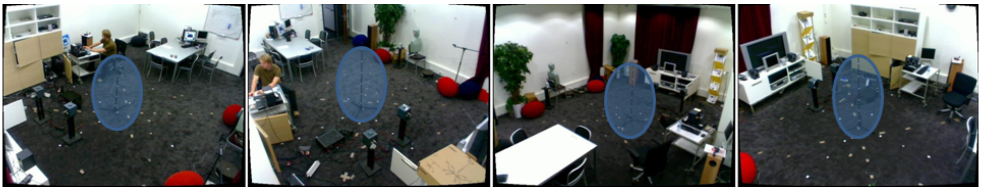


Figure 6.3: *A black microphone stand marked with white tapes for external calibration observed from four different viewpoints. The images are undistorted for radial distortion based on intrinsic calibration.*

To be able to do the 3D reconstruction of the user with the desired metric accuracy, the cameras need to be calibrated. Cameras capture the light, emitted by the 3D environment they are focused on, with their image sensor forming the image plane. Through camera calibration one is able to determine which ray in 3D space is associated with which pixel on the 2D image. There are several methods for multi-camera calibration which use different calibration objects, treat the cameras individually or together [31] and are fully-automatic or require manual work.

The classical calibration approach by manually defining the points of interest in the 3D scene was used for the extrinsic calibration. The intrinsic calibration for radial distortion was performed by the camera calibration toolbox given by Graphics and Media Lab (GML) of Moscow State University. Calibrations were done by assuming the pinhole camera model. The pinhole model is the simplest approximation of camera geometry used in computer vision. The calibration was performed by capturing the projections of known 3D scene points, points whose world-coordinates (x,y and

z) were fixed with a calibration object, on the image plane, see Figure 6.3. These 3D-to-2D point correspondences were used by the Direct Linear Transformation (DLT) algorithm [97] to calculate the projection matrices defining the mapping from 3D scene to 2D image for each camera independently.

6.2.3 3D-shape Reconstruction: Visual Hull

There are many approaches for reconstructing a 3D shape differing in what image features they use, how the shape is represented, and how much detail is preserved of the original object. The shape-from-silhouette (SFS) [112] technique was used in the proposed system. SFS is a method for estimating the 3D non-textured shape of an object by using the silhouette images taken from this object from multiple viewpoints.

SFS is based on the fact that the separated foreground in the form of a silhouette together with the camera viewing parameters can be projected back to the 3D space as a cone that contains the actual foreground object. As every viewpoint forms its own silhouette cone, the intersection of these cones forms the bounding geometry of the actual 3D object. The result is called Visual Hull (VH) [41]. VH is an approximation of the true 3D shape and it depends on the number of views, positions of these views and complexity of the object, see Figure 6.4.

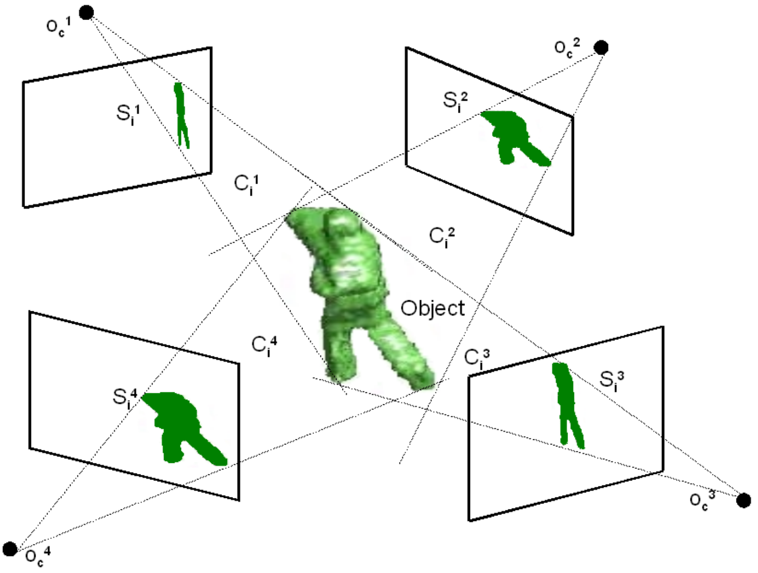


Figure 6.4: Formation of Visual Hull with four viewpoints: Object forms silhouette image $S_i{}^k$ on camera k at time instant i.

A straight forward way of implementing SFS is to define a volume-of-interest (VOI), observed by the cameras, within the environment. The VOI is divided into a grid of 3D voxels. Each voxel's occupancy is independently tested against the silhouette images from available viewpoints. Occupancy testing is performed by projecting the voxel onto the image plane and checking if the voxel projections lie inside the silhouette, see Figure 6.5. This testing is done for all views. For speeding up the on-line processing, the voxel projections per view were stored in a lookup-table structure for fast access enabling real-time volumetric reconstruction.

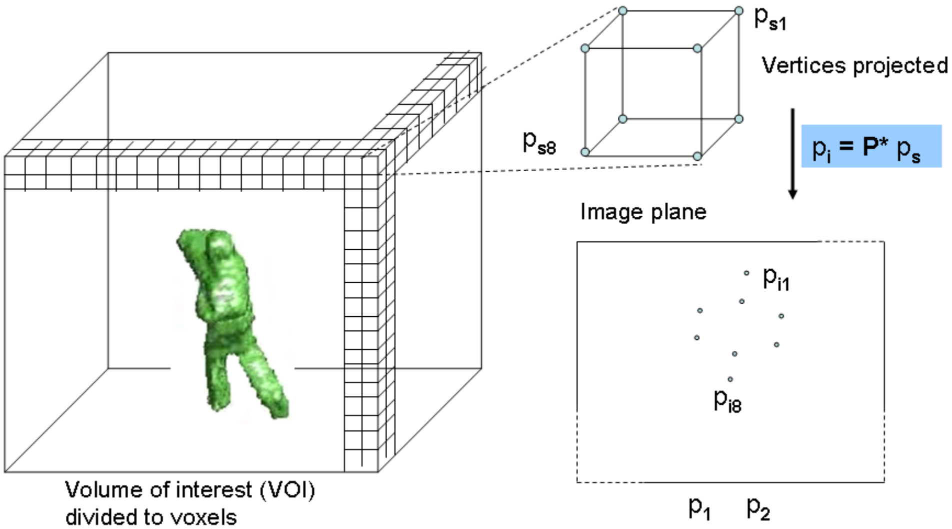


Figure 6.5: *Volume of interest enclosed by a cube formed by regular grid of voxels, whose eight vertices are projected to the image plane. The projections are represented by two uniform points p_1 and p_2 used to test voxels occupancy.*

6.2.4 Visualization: Interactive

In the proposed system the capture of user position and level of activity is performed by analysis of the Visual Hull. By having the 3D shape, the view can freely rotate around the remote user and go closer or further away. The idea is to give the observing user the power to define the angle and proximity by his natural movements. In this fashion the shadows are shown on the screen, which can be considered to act as a see-through window. For example, if the local user moves to the right, he will be shown more of the right profile of the remote user. If the observing user moves further away from the screen used for visualization, his distance to the shadow is significantly increased, see Figure 6.6.

This way by tracking the user's position and showing the corresponding view, the 2D silhouette looks as a 3D object with strong perspective effect. This 3D effect is expected to make the shadow more life-like and further enhance the feeling of the remote user being in the same space with the local user. In addition, based on the amount of activity of the remote user the users shadow is shown in different colors, to serve as a status indicator for peripheral awareness.

In the proposed system the user has the option to move around the shadow of the remote person to a maximum of 120 degrees to each side. The users position is

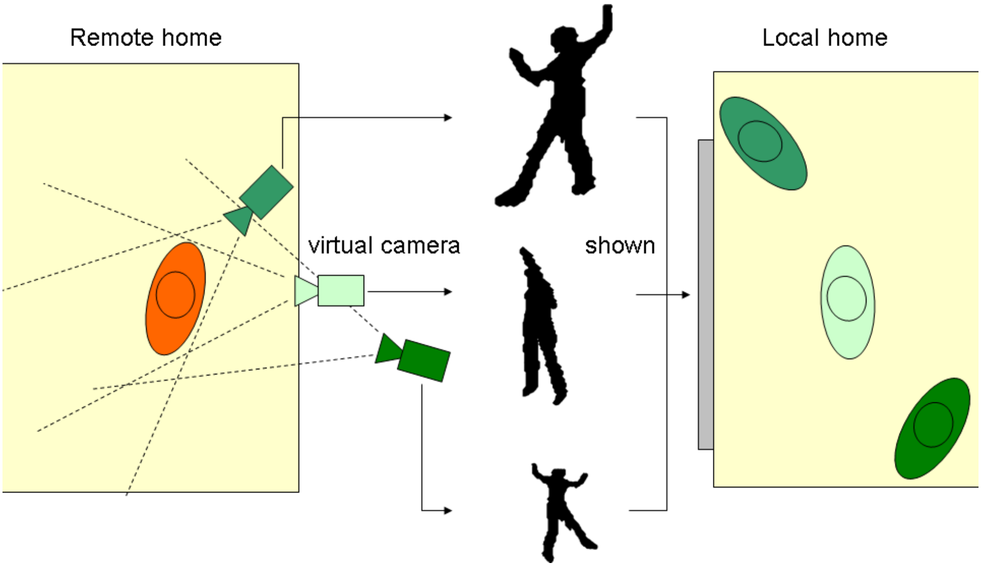


Figure 6.6: *The remote user is shown to the observing user on the right from different angles and proximities based on observing users position. The three user positions and shown viewpoints are illustrated with same color. The roles of observed and observing are here represented for one way case.*

tracked through shape analysis by calculating the centroid of the voxels forming the VH and using this centroid as the center of mass of the user. Based on the centroid the 3D shape can be kept stationary, thus keeping the shadow on the same spot even as the observed user walks around in the VOI. Based on users natural movement the corresponding view is shown. See Figure 6.7 for an example on how the viewer position is determined and Equations 6.1, 6.2 and 6.3 for how these values determine the viewpoints horizontal and vertical rotation γ_{hor} and γ_{vert} and proximity $zoom$.

$$\gamma_{hor}(dx) = \frac{dx}{Dim_x} \times max_{hor} \tag{6.1}$$

$$zoom(dy) = \frac{dy}{Dim_y} \times DoC \tag{6.2}$$

$$\gamma_{vert}(dz) = (dz - CoM) \times s \tag{6.3}$$

max_{hor} is the maximal horizontal rotation angle (120 degrees), DoC is the maximal displacement of camera closer to or further away from the center of VOI (200 cm), CoM is the pre-defined height of center of mass in ordinary standing posture, (60cm), Dim is the VOI dimension in centimeters from the center of the VOI to the border of the VOI either in x (200 cm) or y (200 c,) direction and s is the scaling factor (3) for z-wise movement. The values given for the variables are the ones used in the experimental section. These variables are used in determining the displacement of the virtual camera from its reference position and orientation, which is initially

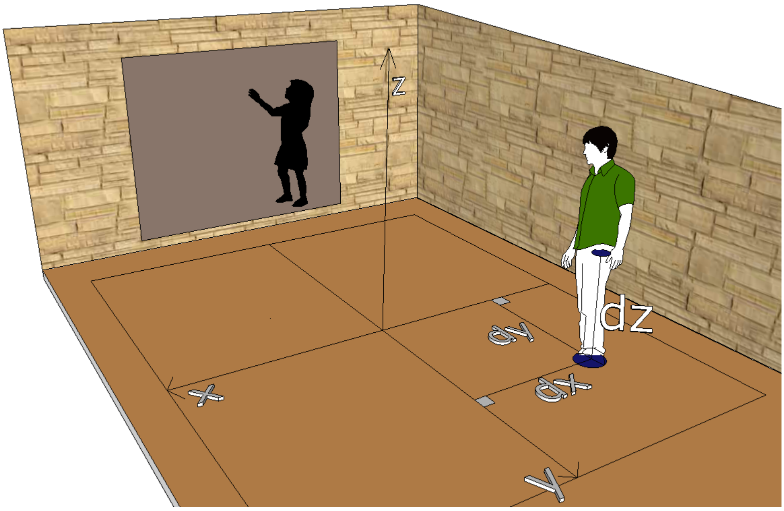


Figure 6.7: *Definition of VOI within an environment: The user position is tracked within the VOI based on the fixed world coordinate system* (x, y, z).

defined at the end of negative y-axis, just outside the VOI with orientation towards the center of VOI.

6.2.5 Visualization: Distortion Free

Projection techniques using existing surfaces of the environment, such as the walls, to display the remote shadows are expected to provide a more life-like see-through effect as the projections can be life-sized and exist in the natural setting. To retain its real world feeling, by not being deformed by geometric distortions, the projection should take into account at least the most basic geometric challenges, the horizontal and vertical tilting of the projection surface w.r.t. the projecting device as well to the user, see Figure 6.8 for illustration of the problem. If the projection surface has a tilting angle α away from the perpendicular orientation against the projector, the relation between original projection width $pwidth_{orig}$ and actual projection width $pwidth_{wall}$ is:

$$\frac{pwidth_{orig}}{pwidth_{wall}} = \cos\alpha \tag{6.4}$$

Thus for the projection on the wall to have the width of the original, the projection width $pwidth_{proj}$ has to be corrected for angular tilt α before projecting on wall as follows:

$$pwidth_{proj} = pwidth_{orig} \times \cos\alpha \tag{6.5}$$

The same correction for redefining the width dimension of the projected shadow can also be performed for the vertical height w.r.t. the vertical tilt of the surface.

For compensating also for the effect of users position the similar correction in horizontal direction is required. Otherwise as the local user moves more to one side to see more of the side profile the projection he sees has shrunk in horizontal direction. To compensate for this the projection width on the wall has to be increased accordingly.

$$pwidth_{seen} = \cos\beta \times pwidth_{wall} \tag{6.6}$$

In order to eliminate the cosine term and thus retain the same width for the observing user the $pwidth_{wall}$ has to be stretched as follows:

$$pwidth_{wall_{new}} = \frac{pwidth_{wall}}{\cos\beta} \tag{6.7}$$

When combined both the shrinking for the angular projection α and expanding for the angular observing β the projection process has to be corrected in horizontal direction as follows:

$$pwidth_{proj} = pwidth_{orig} \times \frac{\cos\alpha}{\cos\beta} \tag{6.8}$$

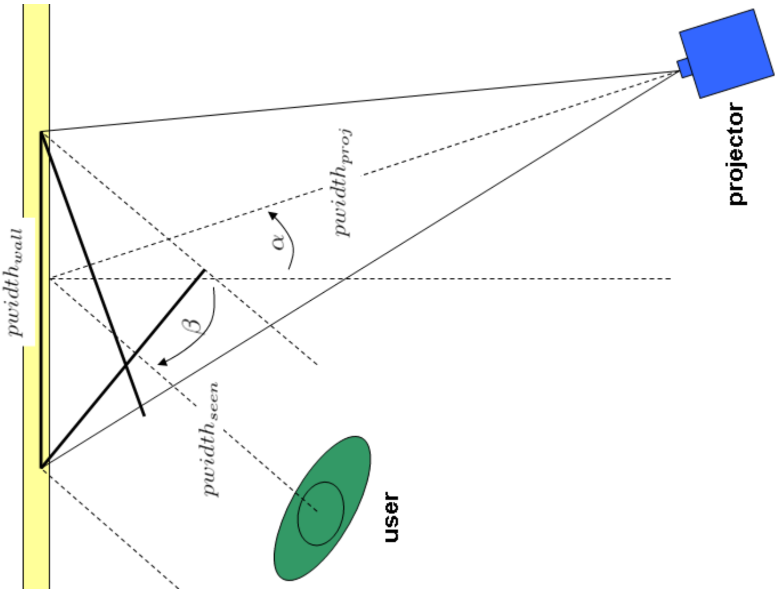


Figure 6.8: Horizontal case: Geometric compensation for projector and observer positions.

6.3 Application Experiments

The proposed system consists of two identical ends, see Figure 6.2, of which one is here used to describe both ends.

6.3.1 Hardware

In the experimental setup built, the four cameras are not situated exactly at opposite corners to avoid redundant views. As the proposed system is for the home, the cameras used were ordinary mid-price webcameras. All cameras are at same height and their views have no occlusions w.r.t. the VOI, see Figure 6.9.

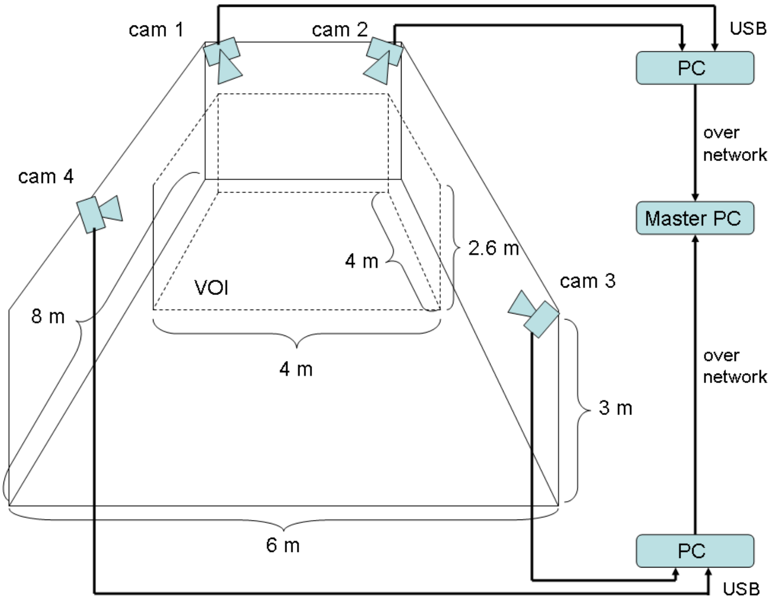


Figure 6.9: *The vision system: modules, cameras and connections.*

The basic blocks of the system are Logitech Fusion USB web-cameras working 20 fps with YUV 444 format in 320 × 240 resolution with 73 degrees of horizontal viewing angle, personal computers (PCs) with 3 GHz Intel IV processor and Windows XP handling data processing/transmission and the network for inter-PC communication. The two PCs run in real-time two simultaneous applications using the connected cameras in generating silhouette image then sent over the network to the master PC having dedicated graphics card for OpenGL rendering of volumetric reconstruction based on the analysis results on the shape of the observing user. The visualization was performed with a projector in this system; in other scenarios also televisions or screens could be used.

The preliminary studies on the effect of available silhouette views showed that by adding a fourth camera an improvement of 25 to 140%, depending on image content, is achieved in the number of outlier voxels. The outlier voxels were simply considered as the voxels, which were present in the three-camera reconstruction, but not in the four-camera shape estimation. In general, the more cameras would be added, the tighter the estimated shape would follow the real shape. However, as a drawback the shape estimation, based on AND-operation of silhouette occupancies, would occasionally falsely reject occupied voxels due to limitations in calibration accuracy and computation accuracy of projective transformations.

The same preliminary study showed that robustness against uncorrelated silhouette noise increases with SFS, when increasing the number of views. Therefore, even with very noisy silhouette images a good shape estimation is possible, when increasing the number of intersecting views significantly.

6.3.2 Data Communications

Each silhouette image is sent over the network in one data-packet. Each of the silhouette packets is time-stamped before sending. The average delay between capture of the scene and the volumetric reconstruction visualized locally was between 0.5 to 1.0 second. This is due to the 0.5 second delay in frame buffering and the rest dealing with camera driver and data transmission and coding. This delay would be visible for the user only when he is moving and the visualization of the remote site would not immediately adapt to his location. Based on a small number of informal user experiments, this effect did not appear annoying and the effect of immersion was still evident.

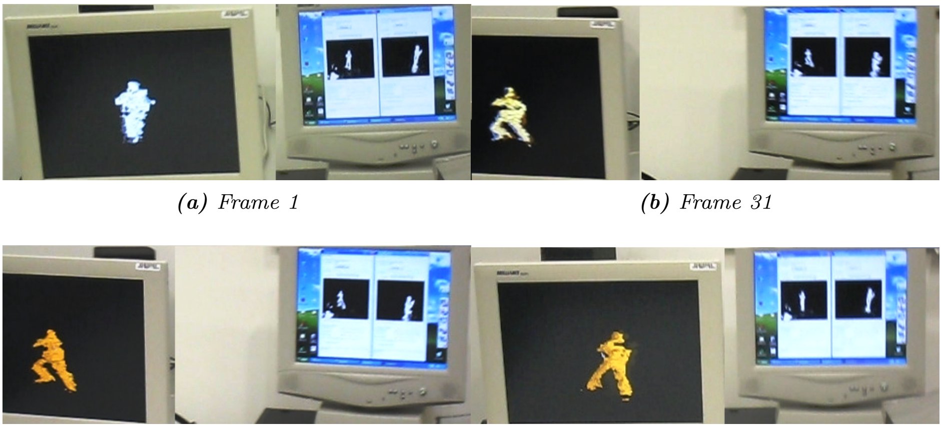
Camera ID	Kilobytes per 30 seconds	Bitrate kB/s
CAM_1	412	13.7
CAM_2	316	10.5
CAM_3	372	12.4
CAM_4	256	8.5
$CAM_{avg-all}$	339	11.3

Table 6.1: Used bytes and corresponding bitrate for a 30 seconds long video sequence with varying amounts of silhouette content.

Average bandwidth required for a silhouette stream was 11.3 Kbytes/s; see Table 6.1 for bitrates in the 4-camera setup. Such a small local network load is achieved by compressing the binary silhouette data by algorithm called Bzip2. The required data-rate is within the limits for, e.g., Zigbee wireless communication which might have a considerable stature in future home networking. Transmission bitrate of compressed shape, sent between homes, is expected to vary between 7 to 20 Kbytes/s depending on the human shape and its noisiness.

6.3.3 Visualization

A real-time demo was built to demonstrate the interactive concepts of the system. In Figure 6.10 a snapshot of the running system is provided, highlighting the activity level based colored visualization. The freedom of user mobility and natural changes of given viewpoint are illustrated in Figure 6.11.



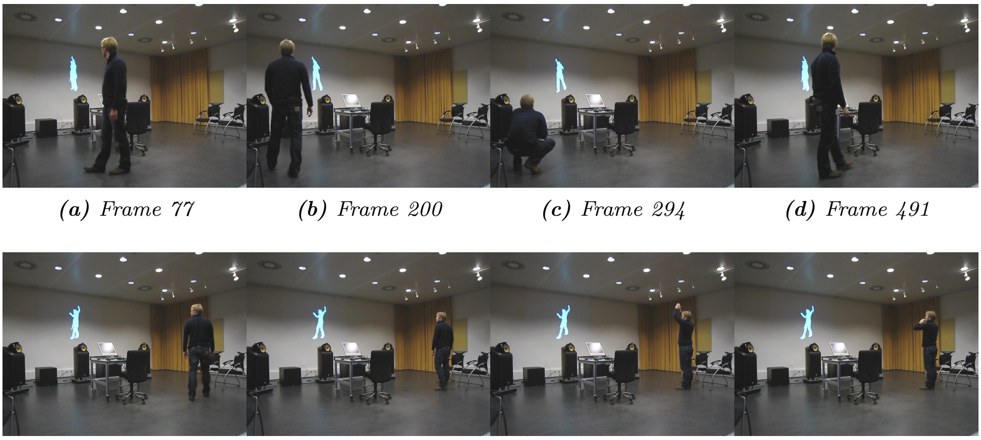
(a) Frame 1

(b) Frame 31

(c) Frame 68

(d) Frame 98

Figure 6.10: *Local user's shape is shown on the front screen and two silhouette masks used to reconstruct the shape on the screen behind on the right all rendered with real-time software. When the user steps to the side, the activity level increases and the VH is correspondingly rendered in yellow color. Frames a-c) user takes a step to the right, d) user hops back to center.*



(a) Frame 77

(b) Frame 200

(c) Frame 294

(d) Frame 491

(e) Frame 694

(f) Frame 797

(g) Frame 894

(h) Frame 942

Figure 6.11: *Simulated action clip of the local observing user walking around the stationary 3D shadow projected on a 2D surface. The first row: observing user goes left and then down to frogs view by crouching, the second row: user goes right and then up to birds view by reaching to the ceiling.*

6.4 Fusion Experiments

The vision network proposed for the application relies completely on the accuracy of the fusion of the silhouette images. Each camera provides the images with same resolution and frame-rate, which makes the direct comparison and fusion of data easier [Design Rule 2]. The data is gathered in one central location, thus a fully centralized architecture is exploited. By performing fusion on the silhouette images, two fusion benefits are apparent:

- *3D-shape reconstruction*: appearance and localization in 3D
- *3D-shape backprojection*: filtered silhouettes in 2D

If noise is present, this will negatively affect the fusion process, and thus the estimated human shape. Common noise sources for shape estimation can include calibration noise, imaging noise and environmental noise.

Given poor calibration the correspondences from 3D scene points to 2D image points will not match between views, resulting in poor joint decisions. Imaging noise affects the image segmentation by creating false and missing hits for foreground areas. False segmentation easily creates errors in shape estimation. The environmental noise is generated by other objects within the scene, either existing within or outside VOI. Depending on the physical location of the noise generating object, such as window showing a weaving tree outside, the noise might be only seen by one camera (no correlation), by some of the cameras (partly correlating), or by all the cameras (fully correlating).

A set of interesting error measures for the accuracy of the responsible vision network could be shown, e.g., by checking the following three comparisons:

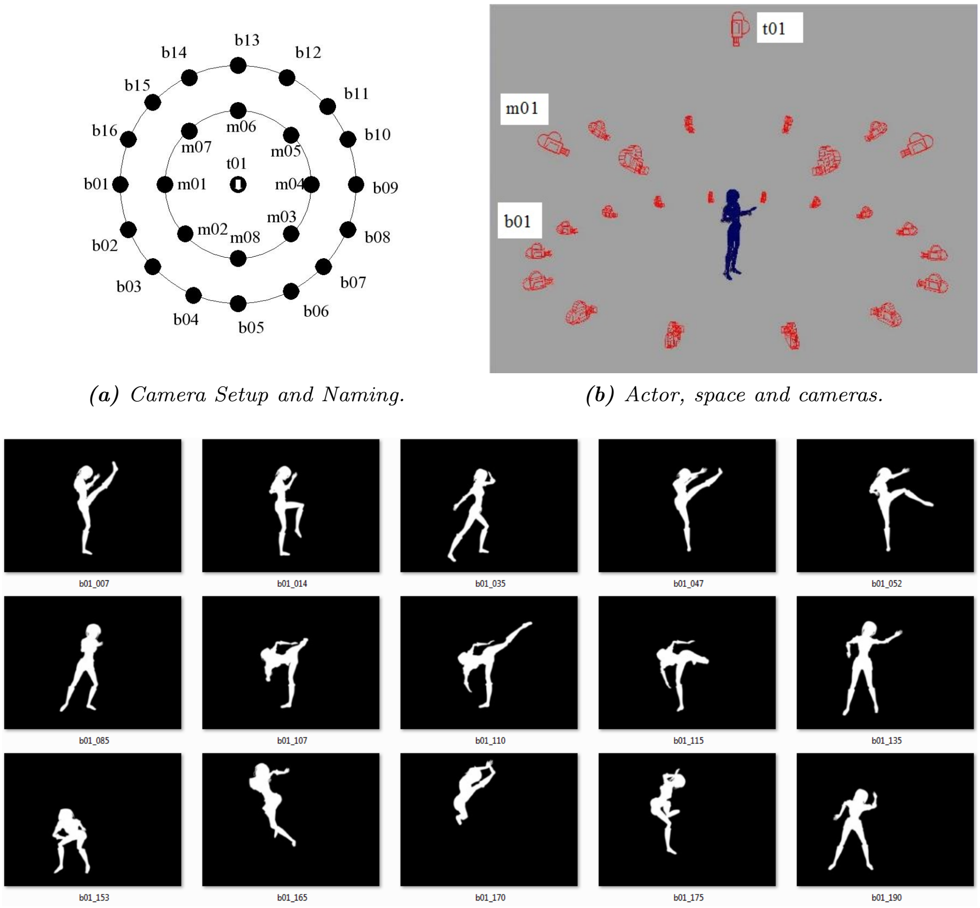
- ground truth silhouette to backprojected silhouette
- ground truth shape to estimated shape
- ground truth tracking to estimated positions in 3D/2D

6.4.1 Ideal Silhouette Dataset

A synthetic test sequence captured by 25 cameras providing both textured and silhouette video streams has been provided by Graphics-Optics-Vision group from Max-Planck-Institut für Informatik [113]. The sequence is 200 frames long, containing various postures and movements, recorded in Maya environment with a 320×240 pixel resolution. This dataset provides an ideal basis for the proposed fusion experiments, as the ground truth data is available and different noise sources can be added to the data in a straightforward manner. An illustration of the dataset setup and examples of the movements involved in the kung-fu sequence are shown in Figure 6.12. The lowest cameras are marked as base-cameras b, the middle tier cameras as m, and the top camera as t.

6.4.2 Study 1 - Camera Configurations

Given such a wide array of cameras to choose from, a three-part study into the significance of camera placement and coverage was conducted. The original ideal



(a) *Camera Setup and Naming.*

(b) *Actor, space and cameras.*

(c) *Example frames from 12 distinctly different time instants by camera b01.*

Figure 6.12: *Ideal Image-Fusion Dataset: Kung-Fu Girl [113].*

silhouettes with no added noise and the optimal camera calibrations provided by the Maya software were used in all the three parts.

Considering a single camera, the occupancy of a voxel is given directly as the value (0 or 1) of the pixel corresponding to the projected coordinates (x,y), value 1 corresponding to foreground and thus representing an occupied voxel. For the entire camera configuration the occupancy estimates were combined by a simple product rule, implementable by a binary AND-operator. The ground truth (GT) shape was computed by using all the base and middle cameras, thus intersecting 24 cameras.

Due to the nature of silhouette based shape reconstruction, the estimated shape is always larger than the GT-shape, given clean conditions. The errors shown on the *top-left* corner of each result figure are the number of incorrectly accepted occupied voxels normalized by dividing it with the number of GT-voxels. This reconstruction error is referred to as the false acceptance rate FAR. For example, FAR-value one corresponds to the estimated shape having as many voxels falsely accepted as are residing in the GT-shape.

A further normalization to the error distributions was carried out by finding the configuration of cameras that for that instant of time had the minimum error. These minimum configurations are shown in *top-right* corner of each result figure. The minimum error was subtracted from the errors of each camera configuration of that particular moment of time. The same procedure was performed over the entire sequence. The results with minimum-error subtraction are shown in *bottom-left* corner of each result figure.

All the error distributions are presented also with a boxplot on the *bottom-right* corner of each result figure. The central mark of a box is the median, the edges of the box are the 25th and 75th percentiles, the whiskers extend to the most extreme data points that are not considered outliers, and the outliers are plotted with individual markers.

Part 1 - On Redundancy

In the first part of the configuration studies seven different two-camera setups (CR1-CR7) were used as source of ideal silhouettes. First base camera b01 was fixed, and the second camera was rotated around the person to seven different locations. Each configuration was studied over the entire 200-frame sequence, by comparing the reconstructed shape to the GT-shape. The camera configurations are provided in Figure 6.13. The shape estimation errors for Part-1 are shown in Figure 6.16.

From top-right of Figure 6.16 it can be seen that the configurations CR2 and CR6 take turns in providing the tightest Visual Hull, thus the least amount of error. By further reviewing the general error distributions of each of the configurations once adjusted by the minimal errors, the configuration CR4 has the biggest error, whereas configurations CR2 and CR6 provide the best results.

Based on these results, it is evident that opposite cameras (CR4) provide very little new information. Having little new information, due to the redundant nature of the viewpoints, results in bad shape estimation. Depending on the posture and action, either one of the orthogonal camera-configurations (CR2 and CR6) will provide the least amount of redundant information. Given the fully complementary nature of the two camera viewpoints, the best shape estimate can be achieved by relying on an orthogonal camera configuration.

Part 2 - On Surrounding

In the second part of the configuration studies, six different three-camera setups (CHO1-CH06) were used. Camera-setups were designed to study the effect of surrounding the person with cameras, instead of having the cameras only on one side of the person. For each two configurations two of the cameras were fixed, and the third camera was located between them either at the same or the surrounding side of the person, see Figure 6.14 for illustration. Otherwise the procedure of the experiment was the same as before, corresponding results shown in Figure 6.17.

All of the three-base-camera configurations do provide minimum error at some point, configurations CHO2 and CHO5 slightly more often than others, but with only a very small margin. The same is more apparent when considering the general error distributions per configuration. No consistent reason for favoring camera configurations that cover the person from both sides seem to exist. Thus a one-sided camera-setup can provide similar accuracy for visibility than a person surrounding one, when considering cameras at the same elevation.

It is worth to mention that in a broader context, such as detecting faces or other posture-direction related person attributes, a one-sided camera-setup is severely limited in detection and tracking of such features. Additionally, camera configurations that cover the person from opposite corners of a room can be expected to provide better accuracy, when a person from a certain side looks similar to a part of the background.

Part 3 - On Visibility

In the third part of the configuration studies six different three-camera setups (CVO1-CV06) were used. Setups were similar to the configurations in the previous part, with the exception of having the in-between camera as one of the middle cameras, which is a camera with a higher observation viewpoint. See Figure 6.15 for illustration. Otherwise the procedure of the experiment was the same as in the previous parts, corresponding results shown in Figure 6.18.

When considering these base-middle-base camera configurations, there seems to be a general consistent reason to favor a surrounding camera setup. The configurations that have the elevated camera on the surrounding side (CVO1-CV03) provide consistently a smaller error in shape estimation, when compared to the base-camera setups (CHO1-CH06) and counterpart configurations (CVO4-CV06) having the elevated camera on same side as the base cameras.

It can be concluded that having an elevated camera by itself does not guarantee a better shape estimation. The benefit of elevation is only apparent in a configuration that is surrounding the person, as it thus creates a view with new information on the scene due to increased visibility of a complementary nature. For example, an otherwise shadowed area between the body and a limb can only be carved away, if there is another camera view from a higher vantage point having a non-occluded view to that specific part of the decision space.

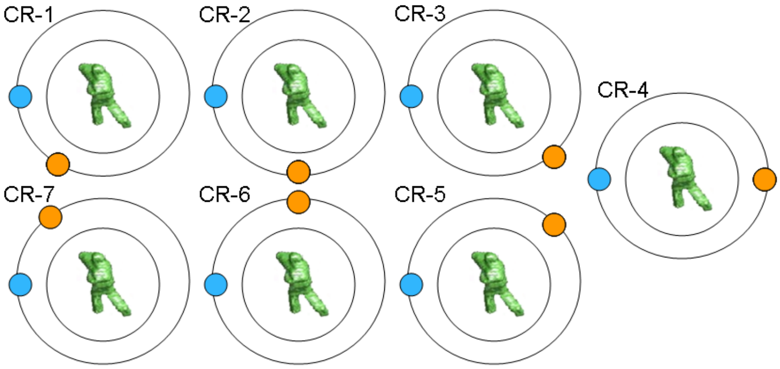


Figure 6.13: *Camera configurations: Part-1 for view redundancy.*

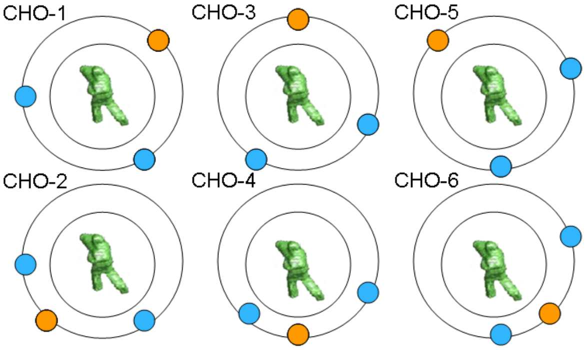


Figure 6.14: *Camera configurations: Part-2 for horizontal view occlusions.*

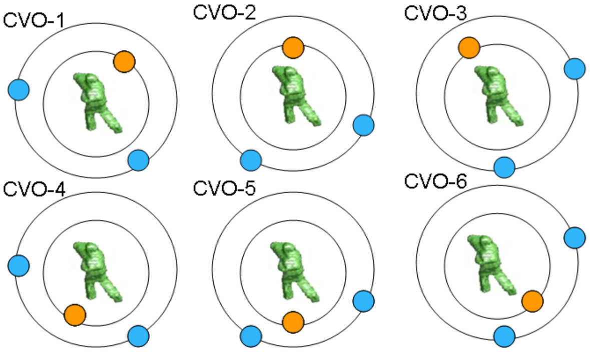


Figure 6.15: *Camera configurations: Part-3 for vertical view occlusions.*

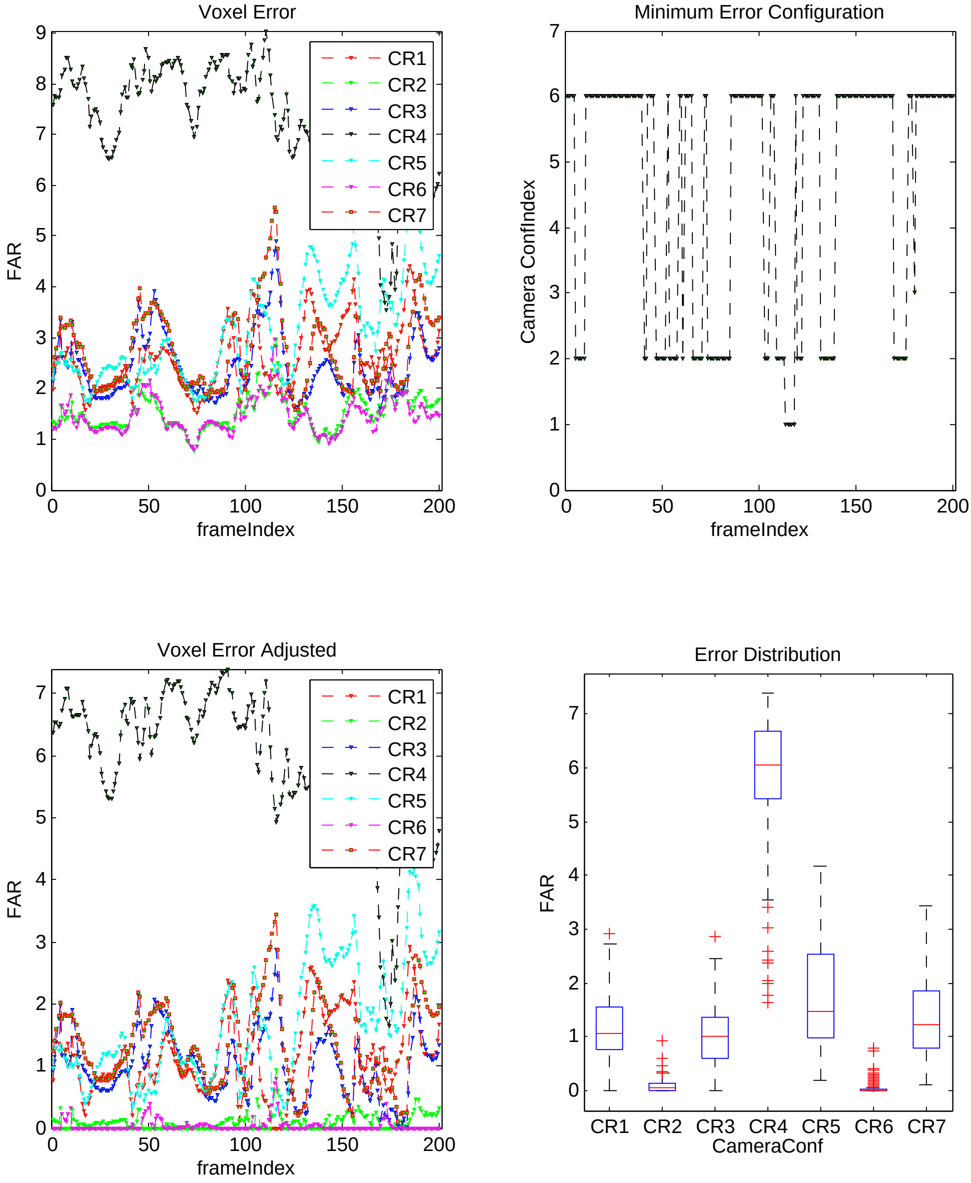


Figure 6.16: *Configuration Study Part-1: Redundancy camera configurations errors for shape reconstruction.*

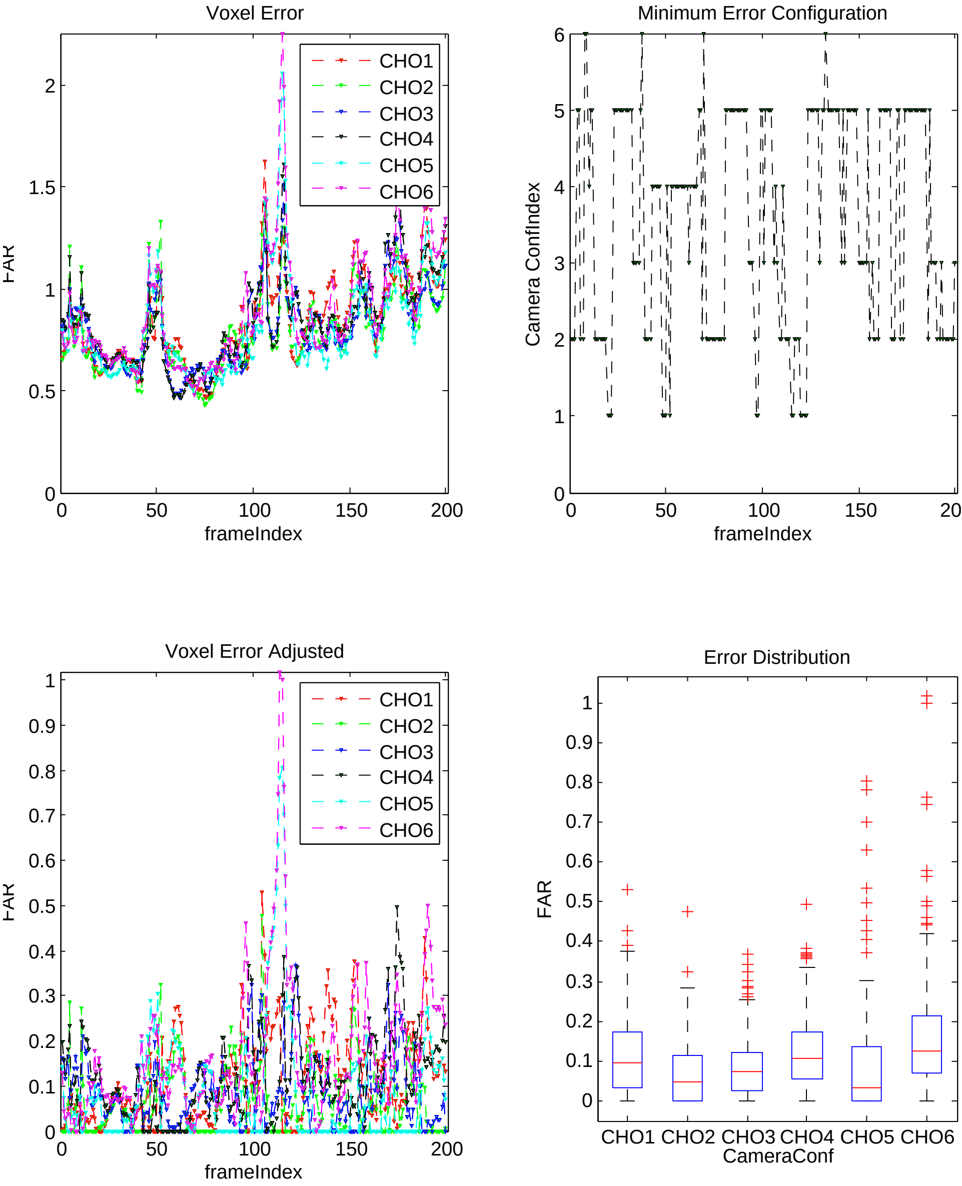


Figure 6.17: *Configuration Study Part-2: Horizontal camera configurations errors for shape reconstruction.*

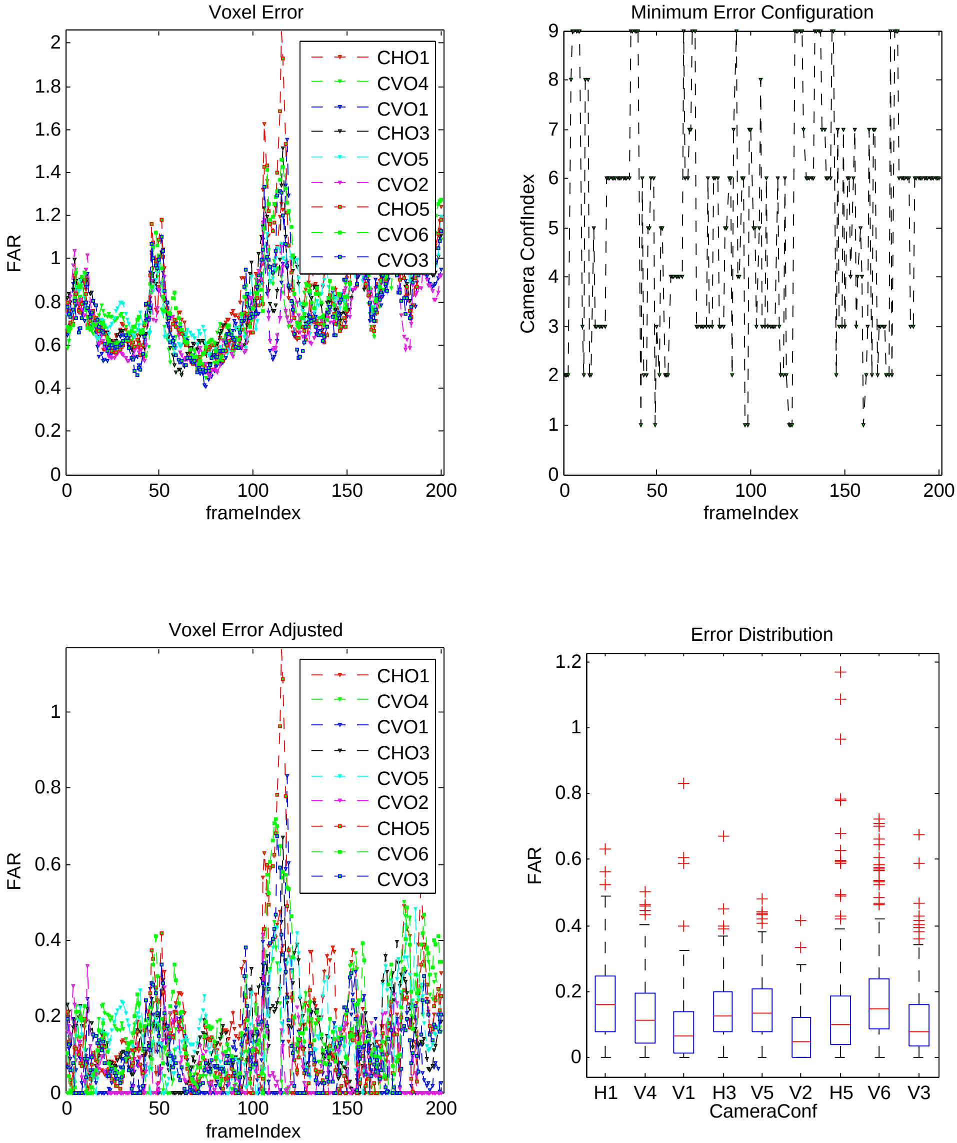


Figure 6.18: *Configuration Study Part-3: Vertical camera configurations errors for shape reconstruction.*

6.4.3 Study 2 - Fusion Methods

For making a joint decision on the occupancy of a voxel, various fusion methods can be used. In the following the accuracy and behavior of five different methods is studied:

1. Binary Product Rule / AND-operator
2. Binary Majority Voting / K-out-of-N Rule

3. Certainty Product Rule (CPR) with threshold $0.51^{nroCameras}$
4. CPR with threshold 0.51
5. CPR with threshold 0.51 and cameras weighted by a prior

For studying the behavior of fusion methods a two-part study is conducted. The first part of the study introduces noise into the silhouettes, second degrades the quality of extrinsic camera calibration.

From Binary to Continuous Data

Depending which method is used, either binary or continuous value is provided by a single cameras occupancy testing. For methods 1 and 2 a binary value, one for a silhouette pixel and zero for background, is provided by the single-camera occupancy testing. For methods 3,4 and 5 a 3×3 filter-grid centered on the voxel projection location, is used to compute an occupancy certainty value as the ratio of occupied pixels to all pixels within that grid.

Including Prior Information

Camera weights can be computed based on many criteria, such as proximity of a camera to VOI-center, vertical angle of camera w.r.t. the VOI-center, or the number of possible sources of dynamic error sources within the FOV (such as windows and doorways). Prior information based on camera proximity is used in this study. Distances to VOI-center are computed on the floor level (x,y), and the priors are scaled such that closest camera has a prior of 1.0 and the cameras further away a value less than 1.0.

Because the noisy silhouettes have holes in them, the shape reconstruction will eventually carve away some of the GT-voxels. Thus in addition to falsely accepted voxels (FAR), we will also have falsely rejected voxels (FRR). The combined false estimations, by a simple addition of FAR (red curve) and FRR (blue curve), are presented as general false rate (GFR) (green curve).

Part 1 - Under Imaging Noise

The effect of low-light conditions on imaging and incorrect thresholding values on image segmentation is simulated by adding noise blobs of size 2×2 to random locations in the silhouette image. This noise is considered as uniform salt-and-pepper noise, removal of which from binary images has been studied, e.g., by Al-Khaffaf et al. in [114]. That is, if the noise is added to an area belonging to the foreground, the value of pixels is set to zero, thus creating a false background hole in the silhouette. Correspondingly when adding noise to a background area, the pixels are marked as foreground. Another option would have been to corrupt the silhouette images by partial occlusions, such as vertical or horizontal bars, which prevent the silhouette from being fully visible. However, by adding salt-and-pepper noise with a variety of noise densities, the results are much more concrete, if not very realistic.

The amount of random imaging noise is defined in percentages of image pixels. All in all six different noise conditions of 0, 1, 3, 5, 7, and 11 % are studied for the five fusion methods. The results of shape estimation in false voxels per noise condition and method are shown in Figure 6.19. The corresponding distributions of error across the sequence over all the noise conditions are shown in *top-row* of Figure 6.21.

The first fusion method, using the binary product rule, performs robustly over all the noise conditions by having no dips/rises in the shape estimations accuracy. Second method, majority voting, is capable of decreasing the number of falsely rejected voxels, thus maintaining a larger amount of the GT-shape than method 1. This is because only a majority of views need to agree a voxel is occupied, therefore limiting the effect of small holes within a silhouette. A side-effect of this approach is the increase in FAR which creates a largely oversized shape.

The third method, CPR with low threshold, manages to further decrease the number of falsely rejected voxels, but on the other hand accepts more false voxels than methods 1 and 2. The fourth method, CPR with higher threshold, maintains a lower FRR than method 1, but FAR is still much higher than with method 1, thus resulting in worse combined accuracy than method 1, but better than the methods 2 and 3 achieve. The fifth method, the prior-weighted CPR, maintains a low FRR and additionally provides lower FAR than with any of the previous methods. By adding the prior-weighting method-5 is able to decrease FAR while maintaining a low FRR. Therefore, method 5 achieves the best combined accuracy of all the methods, when considering all the imaging noise conditions.

Part 2- Under Calibration Noise

As the experiment dataset included calibration information stating the exact positions and orientations of all the cameras, a study on the effect of poor calibration on the fusion methods and on shape reconstruction was carried out. Four calibration noise conditions are defined. The first condition has the GT-settings of cameras. The three other conditions have deviations from GT-position (x,y,z) by either 3, 7, or 13 centimeters in all directions. Based on the noisy camera locations, new projection matrices are computed for each of the cameras and are used in projecting voxel locations for occupancy testing.

The results of shape estimation in false voxels per noise condition and method are shown in Figure 6.20. The corresponding distributions of error over all the noise conditions are shown in *bottom-row* of Figure 6.21.

Across all the calibration noise conditions method 1 has the worst FRR, other fusion methods having better or similar FRR performance. Therefore, by using only a majority of views or by introducing softer certainty values, better FRR performance can be achieved. On the other hand, method 1 provides best FAR, method 5 following closely as the second best. The softness in occupancy decisions introduced in methods 2-5 clearly hinders their accuracy in rejecting false voxels. In combined accuracy it is method 1 that has a slight advantage of 0.2 over method 5, other methods trailing behind.

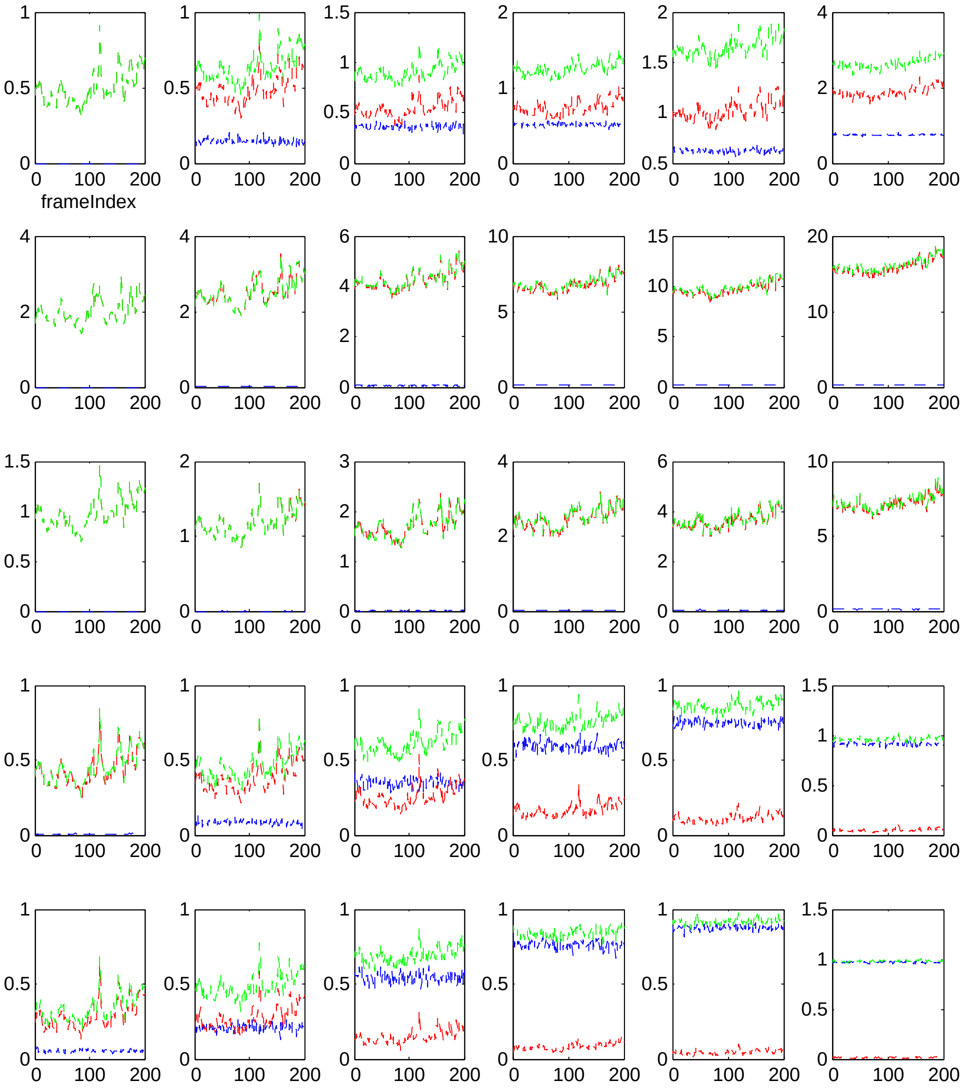


Figure 6.19: *Fusion Methods Study Part-1: Reconstruction Errors of Fusion Methods under different Imaging Noise. Rows represent different fusion methods, with first row as method 1, last row as method 5. Starting from left column the noise percentages are respectively 0, 1, 3, 5, 7, and 11%.*

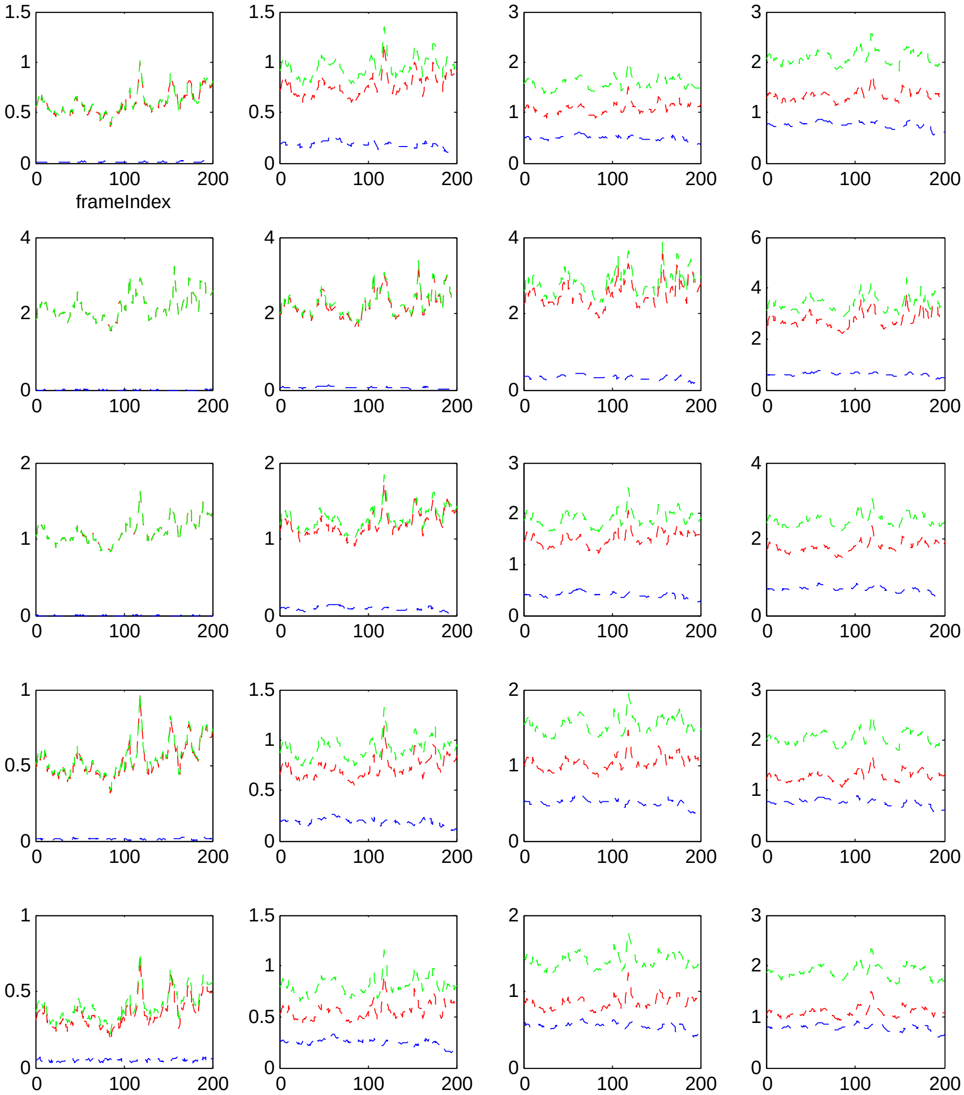
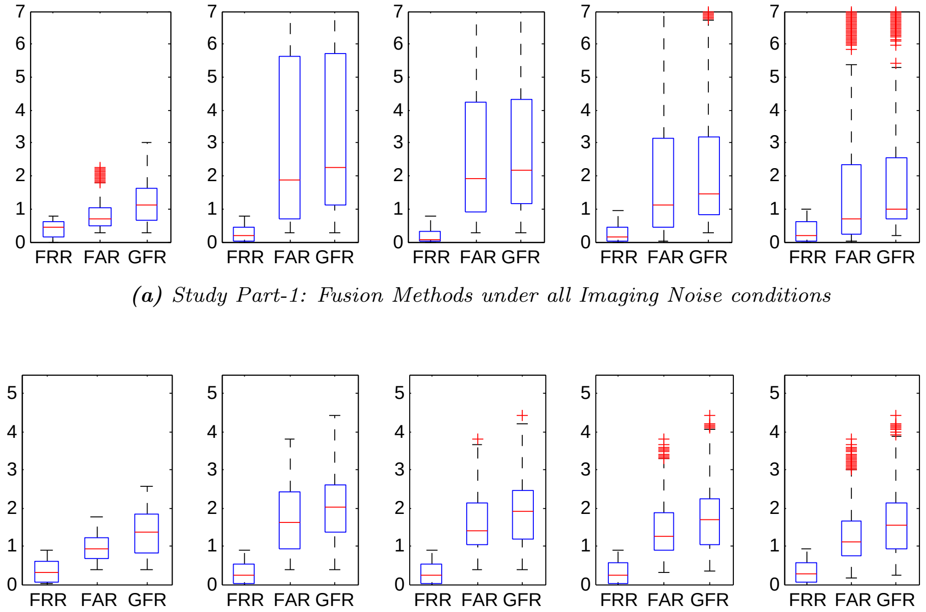


Figure 6.20: *Fusion Methods Study Part-2: Reconstruction Errors of Fusion Methods under different Calibration Noise. Rows represent different fusion methods with first row as method 1, and last row as method 5. Starting from left column the noise in centimeters is respectively 0, 3, 7, and 13.*



***(a)** Study Part-1: Fusion Methods under all Imaging Noise conditions*

***(b)** Study Part-2: Fusion Methods under all Calibration Noise conditions*

***Figure 6.21:** Fusion Methods Study: Error Distributions of the five Fusion Methods, starting from left with method 1.*

6.5 Discussion

A calibrated vision network for the reconstruction of a 3D shape of an object is a very interesting case for testing how different camera configurations and fusion methods perform. By knowing the 3D-to-2D correspondences between scene points and camera pixel coordinates one can reconstruct the shape of a person based on silhouette images. The resulting shape estimate, or visual hull, is an approximation of the shape and location of the person.

Given clean silhouettes and perfect calibration, the more cameras are used, the more close the carved voxel-volume approaches the true shape. However considering the difficulties in image segmentation, due to changing lighting conditions and definition of algorithm parameters such as decision thresholds, the errors of imaging will propagate and deteriorate the shape estimation. Similar effects are apparent, if calibration is not done accurately. For example, given approximately a 22 centimeter error (13 centimeters in all three directions) in camera position estimation for a VOI of $200 \times 200\times$ centimeters, results in almost complete rejection of the true shape.

Application

Having implemented a communication application that connects remote sites by capturing user's 3D shape and location, and by correctly visualizing the corresponding remote shapes with interactive view-angle, several interesting points can be addressed. By carefully calibrating the camera-system, the construction of a geometrically correct approximation of the shape of the observed object was possible. The dynamic selection of a view-angle based on user location was responsive and created a more immersive sensation, even with the 0.5 to 1 second lag from capture to visualization. The bandwidth needed for transmission of a commonly encountered silhouette representation encoded with Bzip2 was found to be in average 11.3 Kbytes. This is well within the limits for the networking standards designed for homes.

Preliminary studies showed that by adding a fourth camera an improvement of 25-140% is achieved in number of outlier voxels. This is assuming the system has ideal silhouettes to combine. However there are limits to how many cameras can be added. Sure, by adding more cameras better and tighter shape estimation can be achieved. However due to accuracy limitations in calibration and computation of projections an increasing number of voxels would be falsely rejected.

The same preliminary study showed that robustness against uncorrelated silhouette noise increases when increasing the number of views. Therefore, even with very noisy silhouette images good shape estimation is possible when increasing the number of intersecting views correspondingly.

Camera Configurations

A study of three parts into the effects of camera configuration to 3D shape estimation was conducted. Based on the two-camera results of the first part it can be concluded that opposite cameras at low elevation provide very little new information due to the redundant nature of their viewpoints. Having little new information results in poor shape estimation. The best shape estimates can be obtained by relying on a geometrically orthogonal configuration of cameras.

The three-camera experiments of part-2 further showed that for shape estimation there is no reason to favor camera configurations that cover the person from all sides, when assuming cameras at the same elevation. However in a broader context, such as detecting faces or other attributes dependent on the direction a person faces, a one-sided camera-setup is severely limited in detecting such features.

Based on the three-camera results presented in part-3 it can be concluded, that having one of the cameras more elevated by itself does not guarantee better shape estimation. The benefit of elevation is only clearly apparent in a person surrounding configuration, because in that case the more elevated view will have the most new information on the scene, due to the increased visibility of highly complementary nature.

The findings of the three-part study can be summarized as follows. The camera configuration used in providing the coverage and visibility to a person, is recommended to be a surrounding setup that has no opposite cameras, and that places the cameras by maximizing the vertical displacement between adjacent cameras.

On Fusion

Lets consider the fusion results in shape estimation presented in this chapter in relation to the proposed vision fusion framework (VFF).

Fusion Approach The shape reconstruction application required the cameras to jointly assign the most probable occupancy for a 3D part of the observed space; these building blocks were referred to as voxels. The communication application was not effected more negatively either by missing or additional voxels. In other words, neither one of these cases were preferred, so the cost for a incorrect occupancy decision was the same for both cases. Therefore the approach in fusion was to declare a decision with the best certainty based on the observations given by the individual cameras [Design Rule 1]. Whereas if the application had required the vision network to never loose track on a person, a reasonable approach would have been to more easily declare a voxel as occupied. This approach would introduce more falsely accepted voxels around the person's true shape, but would provide more robustness in shape estimation and localization against noise sources and data delivery problems.

Vision Network The remote visual communication application example clearly highlighted the need to set up the cameras that match each other in hardware, software settings and data delivery. All the cameras used in the application experiment were the same models. They were set to operate with same parameters for frame-rate and resolution, and automatic gain and white-balance options were disabled and the same manual settings were given to each camera [Design Rule 2].

If settings, e.g., for white-balance or gain in any of the cameras would have been left adaptive, the camera itself could first of all start to produce badly segmented images. The degraded segmentation would further propagate to occupancy fusion, and depending on the fusion approach the effects of this failure might be catastrophic for shape estimation. Likewise, if any of the cameras had a different frame-rate, the frames from that camera would have to be matched to the frames of the other cameras. This match in time may not be perfectly possible for all the frames depending on the used frame-rates. Therefore, the matched frames might not actually present an observation from the exact same moment of time. If this shift in time happens, e.g., when a person moves his arms, it is very likely that the locations of the arms between the views do not match, and therefore this occupied space will not be correctly declared as occupied.

The data was transmitted over the same medium (USB/Ethernet) from all the cameras, with additional NTP-based time stamping for all the frames. The algorithms applied to each video stream were the same and their settings were matched to each other [Design Rule 3]. If the frames from any of the cameras would have been transmitted over a different medium, this would have likely introduced some troubles in matching these frames to the ones from the other cameras. Time stamping would most likely handle well most of the shifts in time, but sometimes be forced to ignore a time moment or use frames with a minor shift. If having to ignore frames, the quality of visualization could be affected by the sudden jumps in time. If exploiting frames from slightly different moments of time, the fusion of occupancy information could easily fail for fast limb movements.

To limit the effect of the shape rendering into vision algorithms, two different options were considered. First, the (projection) screen could be setup outside the VOI for displaying the remote shape. In this case, some cameras would observe the visualization but others would not. The visualization artifacts would be most likely removed in such a setup, because the fusion of views would usually remove any noise sources that exist outside the VOI. Second, the visualization could be shown within the VOI. In this case, most likely all the cameras would observe the artifact. The shape reconstruction should now cope with the rendering artifact by using the correlation between the views to remove the additional noise source within the VOI. First option was opted for the experiments shown in this chapter, because the rendering noise source can be more easily ignored by the common shape reconstruction without additional modifications to the reconstruction algorithms [Design Rule 4].

Fusion Architecture All the experiments collected the multi-camera data in a centralized unit before combining the data into a single estimate on the space occupancy. Because even the raw silhouette data did not require much bandwidth or memory, there were no resource restrictions to the choice in fusion architecture. Additionally, hardly any camera failures could be expected, and in the rare case of a dropped frame the other NTP-timestamped frames could be combined or ignored. Therefore a fully centralized fusion architecture was applied, which maximizes the potential for fusion by allowing it on any fusion level [Design Rules 6 and 7].

Fusion Level Silhouette images were combined in order to decide if a voxel was occupied. The values used in making the decision were given by the silhouette image pixel, binary value O or 1, or by the 3×3 neighborhood of pixels, a certainty value in the range of 0.0 to 1.0. The binary value could be considered as a decision-level value, and the certainty value as a score-level value. Based on the experiments there was no hard evidence to support the use of softer values, certainties, within the shape reconstruction for all noise conditions. In general, the scores performed better when images were affected by noise, but they were again outperformed by decisions with camera calibration noise. Therefore, it would be advisable to not exploit only score- or decision-level values, but to use fusion on both levels and combine these results, as no conclusive domain knowledge seemed to exist [Design Rules 8 and 9].

Fusion Method The two error sources of imaging noise and calibration inaccuracy were studied in a two-part experiment by testing five different fusion methods. The fusion methods combined the occupancy estimates for each voxel into a final decision of occupancy, and thus defined the entire shape estimate. The methods 1 and 2 used the decision-level binary value for declaring the joint estimate of occupancy. The methods 3, 4 and 5 combined the score-level certainty values from each camera.

Considering all the imaging noise conditions, method-5 achieved the best combined accuracy of all the methods. Method-5 exploited both the use of soft decisions and prior camera position information. Across all the calibration noise conditions method-1 that uses binary product rule, had the worst false rejection rate (FRR), with other fusion methods having better or similar FRR performance. On the other hand, method-1 provides the best false acceptance rate (FAR), with method-5 following closely as the second best. In overall accuracy for calibration noise it is method-1 that is slightly better than method-5 with other methods further behind.

The use of a prior weight to each of the cameras increased the accuracy of fusion results. This was shown with method-5 that was otherwise the same as method-4, except for the additional camera-weights. These weights were computed based on the camera distances to the person. Because the assumption on distances remained valid over the entire testing sequence, even these prior and fixed weights were able to introduce a beneficial relative influence between the cameras [Design Rule 10]. The non-correlating results of FRR and FAR with method-1 and method-5 would seem to suggest a good stable accuracy, if the results from these methods were combined by a method suite [Design Rule 12].

In summary The findings of the two-part study on fusion methods can be summarized as follows. With more intelligent fusion methods that exploit both soft values and prior camera information, some of the errors in imaging and calibration can be compensated for in multi-camera occupancy testing. The compensation is unfortunately usually followed with an increased cost in falsely accepted occupied space.

Chapter 7

Recognition of Activities

A vision system can better understand a scene by inferring some actions-of-interest based on its observations. A study on classifying human activities based on multi-view observations is performed with three fully annotated datasets: MAS, VIHASI and HOMELAB, and a combination dataset MAS+VIHASI [1].

This study focuses on five different methods to combine multi-view data from uncalibrated smart cameras for recognizing a set of six activities. Classification is performed based on image features computed from silhouette images with a binary tree structured classifier using 1D conditional random field for temporal modeling. The multi-view fusion scenarios studied are divided into two categories: view selection and view combination methods. Selection uses a single view to classify, whereas combination merges multi-view data either on the feature- or decision-level.

Section 7.1 starts this chapter by discussing previously presented approaches to handling multi-view data for activity classification. The proposed approach for activity modeling is presented in sections $7.2-7.3$. After defining the five fusion-scenarios in section 7.4 and illustrating the three datasets in section 7.5, the classification accuracy results are shown and discussed in sections $7.6 - 7.7$. The chapter concludes in section 7.8 with a summary of major findings in relation to the vision fusion framework.

[1] This chapter is (partly) based on:
Määttä, T., Härmä, A., Aghajan, H. (2010). *On efficient fusion of multi-view data for activity recognition.* 4th ACM/IEEE Int. Conf. on Distributed Smart Cameras.

7.1 Related Work

As people move around in an environment covered by multiple smart cameras, they can be seen in varying number of views depending on the geometry of the place and relative location of subjects and cameras. A single camera might provide enough information to classify the activity of a person with a reasonable accuracy, but having other view(s) providing simultaneously data from other perspectives is expected to improve the accuracy. The information about the activities and behaviour of the subjects can be used to control various services, e.g., in smart homes or shops, or to collect important events, e.g., in care homes or hospitals.

One proposed method to handling multiple views has been to explicitly account for the change in aspect/view [115, 116]. A classifier trained in one view should be put to use in a completely new view. By looking into feature correspondences within the same classes of activities between the source view and the target view a suitable transfer of the model could be found. This method is computationally expensive and requires a fixed person-to-camera angle to work optimally.

Cherla et al. demonstrated in [117] that view-invariant recognition can be performed by using a simple data fusion of minimum of two orthogonal views. This is based on the knowledge that better recognition is achieved e.g. from a side view for a kicking or pointing action. Therefore, by using the appropriate camera to recognize a particular activity should provide better accuracy. In real-world scenarios assumption of view orthogonality is easily invalidated, because the person moves within the scene and rotates around this way, e.g., breaking the assumption of a frontal view. Additionally, choosing an appropriate view is driven by the type of action, which itself is unknown too.

The approach proposed in this chapter aims to provide nearly view-independent image features. All the views are given the same priority and no geometrical assumptions on the scene or cameras are made. This approach aims to develop a single model that works with all the views providing accuracy close to that of view-specific models. The multi-view data for activity classification can be combined by any fusion architecture, level and method. A fully centralized fusion architecture is used to gather data in this study. Two different fusion levels of feature and decision are compared through five different fusion scenarios. Sensor selection, majority voting, and arithmetic average are used as the fusion methods.

The details on the methodologies used in this study for modeling human activities, and for combining data from multiple views according to described multi-camera fusion scenarios, will be presented. After describing the camera scenarios, the performance of these different scenarios is explored.

7.2 Activity Modeling

For classifying activities based on video, the video has to be processed to a simpler form that focuses on the person and gives essential information.

7.2.1 Image-based Features

Regardless of its challenges silhouette extraction is a widely used method in preprocessing images to be able to focus on the object of interest, in our case people. Silhouettes themselves could already be used as representation of people. However due to various artifacts depending on selected foreground segmentation method, we applied background subtraction [111], silhouettes are usually processed further into more robust and simpler descriptions, called image features.

We tested three fundamental categories which contain 29 silhouette-based features combined. First, assuming relatively upright cameras, person's posture would be a vital clue. Second, global motion of a moving person would offer insight into how mobile the person is. And third, local motion of the person would indicate an activity involving the use of person's limbs [118] or minor shifts of balance/orientation. The 29 features are listed in Table 7.1, and the equations referred to in the table are defined below the table in Equations 7.3-7.8. In Figure 7.1 one can see illustrations on the four basis descriptors used. The motivation for the selected features and their definitions are given in the following subsections.

7.2.2 Scaling and Normalization

All the features representing changes over time are transferred from differentials across an observation window to the change of a characteristic in one second. This is obtained by multiplying differentials by scaling factor s. Factor s is defined as the ratio of capture frequency fps and the observation interval l_w expressed in the number of frames.

$$s = \frac{fps}{l_w} \tag{7.1}$$

$$v_j^{new} = \frac{v_j - \mu_j}{\sigma_j} \tag{7.2}$$

In addition to compensating for temporal differences between datasets, when having fixed window length, a mean-variance normalization to each image feature was performed. For each dataset, the mean μ_j and variance σ_j of each feature v_j is computed and used to center and scale the distribution of each feature offline. Goal is to normalize the features into a form that generalizes them better to other environments/datasets.

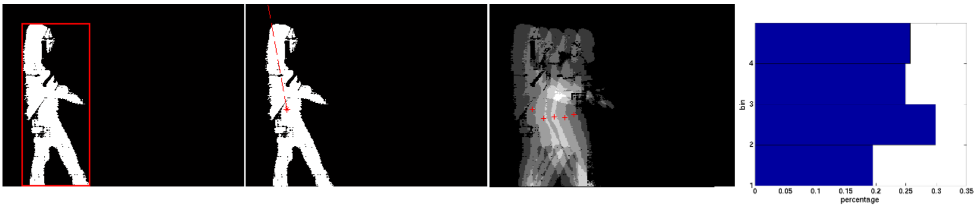


Figure 7.1: *Examples of the silhouette-based features used as the basis, starting from left: bounding box, principal angle, median coordinate (x,y), and 4-bin shape descriptor.*

7.2.3 Features of Posture

To detect the posture and changes in posture two fundamental features were exploited: bounding box aspect ratio and principal component angle. The aspect ratio of a bounding box surrounding the silhouette pixels was used as a rough measure of the posture. In addition to using the aspect ratio of the bounding box AR_{bb}, also its behaviour over time was studied. We focused on observing the change in the aspect ratio ΔAR_{bb}, box height ΔH_{bb}, and box width ΔW_{bb}.

The overall posture was characterized by the angle of the silhouette on the image plane. This angle was estimated using principal component analysis and marked hereafter as PCA. The behaviour of the angle introduced two additional features: the change-in-angle ΔPCA and the change of the change-in-angle $\Delta^2 PCA$.

7.2.4 Features of Global Motion

The overall movement of the 2D silhouette was used to detect global motion $GloMo$. The median coordinate mc of the positions of all the silhouette pixels is used as the basis. By observing the first and second differential of the movement of the median coordinate we were able to detect global motion Δ and acceleration Δ^2 in both horizontal W and vertical H directions. For an additional global motion cue the change of the location of the bounding box in horizontal ΔLW_{bb} and vertical directions ΔLH_{bb} was also computed.

7.2.5 Features of Local Motion

To detect minor changes in the positions and orientations of limbs, a vertical 4-bin histogram on silhouette pixels was computed, called hereafter the Shape Descriptor SD.

The vertical range of the SD is defined by the silhouette. This range is divided into four same length intervals, and based on the amount of silhouette pixels within the interval, the histograms consisting of the four bins is built. See an example of an SD in the right most image in Figure 7.1. In addition to representing how the mass of the person is distributed in vertical direction, the behaviour of SD was also studied. The deformation ΔSD and change of deformation $\Delta^2 SD$ were used to describe how the mass distribution evolves over time.

In addition to detecting the behaviour of the median coordinate on the image plane, the same observations were performed with respect to the bounding box, here referred to as $mcbb$. This results in similar coordinate differential features, but now observed within the bounding box, as cue for local motion $LocMo$.

Another interesting option for silhouette-based features are the features introduced by Barnich [119], which are based on characterizing the human silhouette as the set of all the maximal rectangles that can be wedged inside it. They managed to implement this feature extraction in real-time, which is uncommon for surface-based descriptors, and it managed to provide better classification accuracy than a widely used surfacic silhouette descriptor.

Table 7.1: Features based on Bounding Box (BoB), Principal Component Angle (PriCA), Median Coordinate (MeCo) and Shape Descriptor (ShaDe).

Posture	index	definition
BoB	1	$AR_{bb}(t) = \frac{height_{bb}(t)}{width_{bb}(t)}$
	7	$\Delta AR_{bb}(t) = s \cdot \lvert(AR_{bb}(t) - AR_{bb}(t - l_w))\rvert$
	8	Equation 7.3, with $X = H$
	9	Equation 7.3, with $X = W$
PriCA	2	$PCA(t) = \begin{cases} \arctan(PC_V(t)/PC_H(t))/Pi * 180 & \text{if } \arctan(PC_V(t)/PC_H(t)) \geq 0 \\ 180 - \arctan(PC_V(t)/PC_H(t))/Pi * 180 & \text{if } \arctan(PC_V(t)/PC_H(t)) < 0 \end{cases}$
	12	$\Delta PCA(t) = s \cdot \lvert PCA(t) - PCA(t - l_w)\rvert$
	13	$\Delta^2 PCA(t) = s \cdot \left\lvert(PCA(t) - PCA(t - \frac{l_w}{2})) - (PCA(t - \frac{l_w}{2}) - PCA(t - l_w))\right\rvert$
GloMo	index	definition
MeCo	3	Equation 7.4, with $X = H$
	4	Equation 7.4, with $X = W$
	5	Equation 7.5, with $X = H$
	6	Equation 7.5, with $X = W$
BoB	10	Equation 7.6, with $X = H$
	11	Equation 7.6, with $X = W$
LocMo	index	definition
ShaDe	14	SD_1, 1st of 4 bins of vertical silhouette scanline centered on $MeCo$
	15	SD_2
	16	SD_3
	17	SD_4
	18	$\Delta SD_1(t) = s \cdot (SD_1(t) - SD_1(t - l_w))$
	19	$\Delta SD_2(t)$
	20	$\Delta SD_3(t)$
	21	$\Delta SD_4(t)$
	22	$\Delta^2 SD_1(t) = s \cdot \left[(SD_1(t) - SD_1(t - \frac{l_w}{2})) - (SD_1(t - \frac{l_w}{2}) - SD_1(t - l_w))\right]$
	23	$\Delta^2 SD_2(t)$
	24	$\Delta^2 SD_3(t)$
	25	$\Delta^2 SD_4(t)$
MeCo BoB	26	Equation 7.7, with $X = H$
	27	Equation 7.7, with $X = W$
	28	Equation 7.8, with $X = H$
	29	Equation 7.8, with $X = W$

$$\Delta X_{bb}(t) = s \cdot \frac{|(X_{bb}(t) - X_{bb}(t - l_w))|}{X_{bb}(t)} \tag{7.3}$$

$$\Delta LX_{mc}(t) = s \cdot \frac{|LX_{mc}(t) - LX_{mc}(t - l_w)|}{X_{bb}(t)} \tag{7.4}$$

$$\Delta^2 LX_{mc}(t) = s \cdot \frac{\left|(LX_{mc}(t) - LX_{mc}(t - l_w)) - (LX_{mc}(t) - LX_{mc}(t - \frac{l_w}{2}))\right|}{X_{bb}(t)} \tag{7.5}$$

$$\Delta LX_{bb}(t) = s \cdot \frac{|(LX_{bb}(t) - LX_{bb}(t - l_w))|}{X_{bb}(t)} \tag{7.6}$$

$$\Delta LX_{mcbb}(t) = s \cdot \frac{|LX_{mcbb}(t) - LX_{mcbb}(t - l_w)|}{X_{bb}(t)} \tag{7.7}$$

$$\Delta^2 LX_{mcbb}(t) = s \cdot \frac{\left|(LX_{mcbb}(t) - LX_{mcbb}(t - \frac{l_w}{2})) - (LX_{mcbb}(t - \frac{l_w}{2}) - LX_{mcbb}(t - l_w))\right|}{X_{bb}(t)} \tag{7.8}$$

7.3 Classifiers and Their Structure

As the class categorization under consideration seems to have a natural way to conform under a hierarchical model, a tree is taken as the structure for the classifier, see Figure 7.2. The tree is binary with only two splits per node.

7.3.1 Tree Structure

The root split discriminates between mobile (walk/run) and stationary activities (others). The mobile branch is split again into classes representing walking and running activities. The stationary activities are split into upright (stand/stand-interact) and less upright categories, after which decision on interactivity of the action is made. See Figure 7.2 for an illustration.

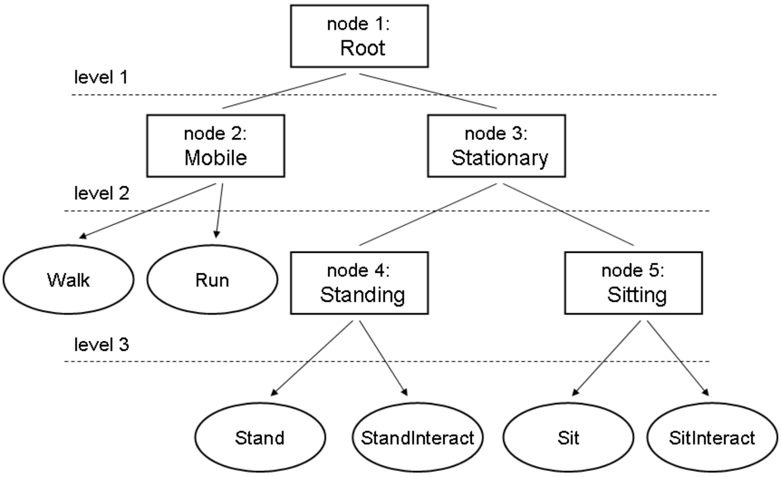


Figure 7.2: *Logical tree structure for the six activities of interest.*

7.3.2 Temporal Activity Models

A 1D model, that could describe the patterns in the features, was needed in order to decide which branch of the classification tree to take in each of the nodes. From widely used temporal models such as hidden Markov models (HMMs) [120] and maximum entropy Markov models (MEMMs) [121], we used conditional random fields (CRF) [122] as representative of these models. See Figure 7.3 for graph on the 1D chain structure of HMM and CRF models.

HMM assumes that a process being modeled is a Markov process, current label Y_i depends only its previous hidden state Y_{i-1} and the current observation X_i. In contrast, CRF conditions on an observation sequence x to find the most likely label sequence y by computing $p(y|x)$ for each label sequence. CRF does not model the properties of the observations themselves like HMM does, notice closed/open observation circles in Figure 7.3. By not having independence assumptions on the observations CRF is able to incorporate bigger set of features than HMM without negatively affecting accuracy even when adding redundant features.

After preliminary experimentation, the observation interval l_w was fixed to 5 frames for all the experiments reported. For model training, iterations were limited to 600 and training was re-computed in case of random initialization preventing model convergence.

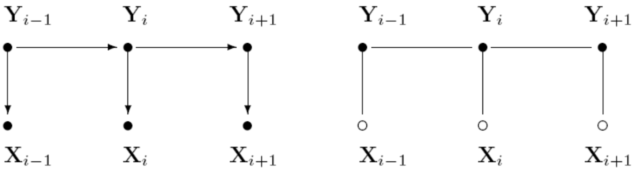


Figure 7.3: *Graphical models of (left) a simple HMM, (right) a similar CRF. Y_i is the current state, X_i is the current observation.*

7.4 Multi-View Scenarios

As the vision networks under study are uncalibrated, see section 2.2.3 of this thesis, each view is independently processed. The output from a view is either a stream of features or labels depending on whether classification is done within the smart camera or by a centralized process.

We introduce five different camera use scenarios to get insight into how multi-camera systems should handle multiple streams of image features of the same scene; see Figure 7.4 for illustration. The camera scenarios are divided into two distinct categories: view *selection* and *combination*. We defined the different camera scenarios in a general manner so that further tailoring can be done if additional information of the environment is available.

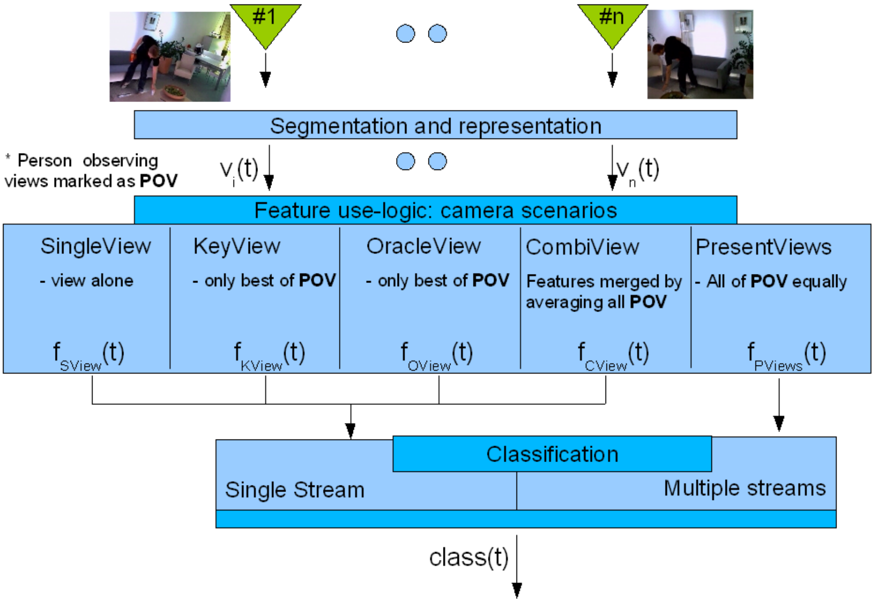


Figure 7.4: *Proposed system with scenarios for feature and decision fusion.*

A view that observes a person (POV) is used to refer to a view that has the person in its field of view (FOV). A person might be only partially visible in a POV. For each time instant t the amount of POVs is given by n, and the feature-vector of size $fL{\times}1$ of view i is given by $\boldsymbol{v}_i(t)$.

7.4.1 View Selection

Scenarios SingleView(SView), KeyView(KView) and OracleView(OView) are methods that rely on features given by an available or selected single view.

SingleView
SingleView represents a case when only one view is used to observe a person. SingleView uses thus one fixed view, no fusion of features or decision labels is necessary. The feature-vector $\boldsymbol{f}_{SView}(t)$ with fixed view i as input for classification is given by:

$$\boldsymbol{f}_{SView}(t) = \boldsymbol{v}_i(t) \tag{7.9}$$

KeyView
KeyView, or shortly as KView, is defined as the POV that fills the silhouette-based three-part criteria the best. The three parts in descending priority are: silhouette fully in FOV (no cut offs), size of the area of the silhouette, and amount of detected global motion. If multiple views fulfill the first priority, the second priority is used for selection and so forth. I is used for an indicator function, that gets value one only when criteria is fulfilled in best manner among views. Therefore, I here selects only the best view by the given criteria as the nonlinear function F. The selected feature-vector $\boldsymbol{f}_{KView}(t)$ is thus given by:

$$\begin{aligned}\boldsymbol{f}_{KView}(t) &= F\{[\boldsymbol{v}_1(t), \boldsymbol{v}_2(t), \ldots, \boldsymbol{v}_n(t)]\} = F\{\boldsymbol{V}(t)\} \\ &= \sum_{i=1}^{n} I_{\text{bestView}}(\boldsymbol{v}_i(t))\boldsymbol{v}_i(t) \end{aligned} \tag{7.10}$$

OracleView
OracleView, or shortly as OView, is a theoretical scenario, in which the system would be able to pick the POV that would give the correct label for a certain activity, if such a view existed for a given time instant. This scenario gives us a reference on how an ideal single camera selection, ideal KView, would perform with the given classifier system.

7.4.2 View Combination

All the scenarios defined so far have been based on the use of a single view, either by choice or circumstance. PresentViews (PViews) and CombiView (CView) scenarios are based on fusion of multi-view data either at decision- or feature-level.

PresentViews
PViews scenario uses all of the POVs individually to first classify and only thereafter combines the decisions labels. Each view has an equal weight/priority. The classifier input $\boldsymbol{f}_{PViews}(t)$ is now a set of all the POV feature-vectors:

$$\boldsymbol{f}_{PViews}(t) = \{\boldsymbol{v}_1(t), \ldots, \boldsymbol{v}_n(t)\} \tag{7.11}$$

The output decision label $class(t)$ is the most common label L of all the classes s among all the individual classifier outputs $c_i(t)$.

$$class(t) = \arg\max_{L \in s} \sum_{i=1}^{n} (c_i(t) == L) \tag{7.12}$$

CombiView

As PViews combines decisions, CView scenario on the other hand combines features of all the POVs into a single feature-vector, which is thereafter used to classify the activity. Feature fusion is performed according to a multiplication with a transformation vector $\boldsymbol{k}$. In our approach $\boldsymbol{k}$ was set as an averaging transformation (Equation 7.14), but other operations such as weighted average may provide improvement. The input feature-vector $\boldsymbol{f}_{CView}(t)$ is therefore defined as:

$$\begin{aligned} \boldsymbol{f}_{CView}(t) &= [\boldsymbol{v}_1(t), \boldsymbol{v}_2(t), \ldots, \boldsymbol{v}_n(t)]\,\boldsymbol{k} \\ &= \boldsymbol{V}(t)\boldsymbol{k} \end{aligned} \tag{7.13}$$

$$\boldsymbol{k} = \left[\frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n}\right]^T_{1\times fL} \tag{7.14}$$

7.5 Datasets used in the experiments

Three different multi-view datasets are used for conducting experiments. Two of them, MAS (Manually Annotated Silhouettes) and VIHASI (Virtual Human Action Silhouettes) offer ideal silhouettes [123]. Silhouettes are without any holes, noise blobs or other artifacts with all the cameras observing the person at all times. MAS is based on recorded people performing actions live in their own pace. VIHASI is based on motion-capture driven avatars, whose appearance differs, but the actions they perform are exactly the same in duration and pace.

The third dataset is called HOMELAB, as it was recorded in a multi-room laboratory space furnished as a home environment; see Figure 7.6 for examples of the environment and views. In the HOMELAB dataset ten subjects (heights between 150 and 190cm) perform the same scripted routine in their own pace. See Table 7.2 for more specific information on the three datasets. Figure 7.5 illustrates the subjects and/or viewpoints of MAS and VIHASI datasets.

As only the HOMELAB dataset had been recorded and annotated with the 5 main activities in mind, both MAS and VIHASI datasets had to be reannotated to match the already existing activity categories. Reannotation was done by merging some primitive activities under a existing activity label and in some cases the primitive activity had to partly or entirely removed from the new reannotated dataset. Table 7.3 illustrates how the original activities of both datasets were handled to fit them under the existing activity classes. In addition to the 5 classes of HOMELAB data a 6th class for running activity was introduced.

Table 7.2: Comparison of the three examined datasets

DATASET	HOMELAB	MAS	VIHASI
Actors: (f)emale,(m)ale	$2 \times f$, $8 \times m$	$2 \times m$	$2 \times f$, $3 \times m$
Orig. activities	5	14	20
Target activities	5	6	6
Views/POVs	8 / 0-3	2 / 2	12 / 12
Frames-per-second	10	25	30
Resolution	320 × 240	720 × 576	640 × 480
Silhouette quality	noisy	ideal	ideal

Table 7.3: 15 action classes from VIHASI and 12 from MAS are mapped to the 6 activities of interest.

Class	original VIHASI classes	original MAS classes
Walk	HeroDoorSlam Walk WalkTurn180(partly)	WalkToLeft WalkToRight
Stand	Collapse(partly) StandLookAround	CollapseLeft(partly) CollapseRight(partly) StandUpLeft(partly) StandUpRight(partly)
Sit	KnockOutSpin(partly)	CollapseLeft(partly) CollapseRight(partly)
StandInt	Granade HeroSmash Punch	GuardToKick GuardToPunch KickRight PunchRight
SitInt	KnockOut(partly) KnockOutSpin(partly)	CollapseLeft(partly) CollapseRight(partly) StandUpLeft(partly) StandUpRight(partly)
Run	Run RunPullObject RunPushObject RunTurn90Left RunTurn90Right	RunLeft RunRight

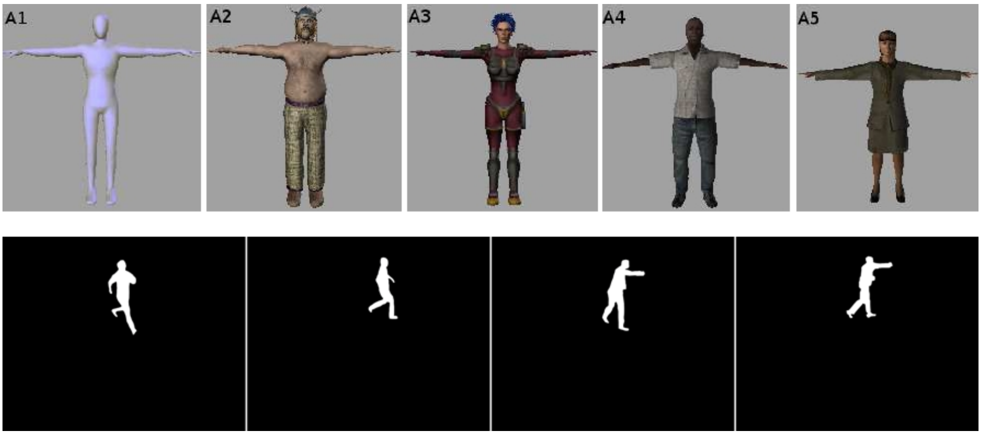


Figure 7.5: *Examples of the observed subjects; top-row) all five avatars of VIHASI dataset, bottom-row starting from left) Actor1 running in two views and Actor4 StandingInteracting in two views of MAS dataset.*

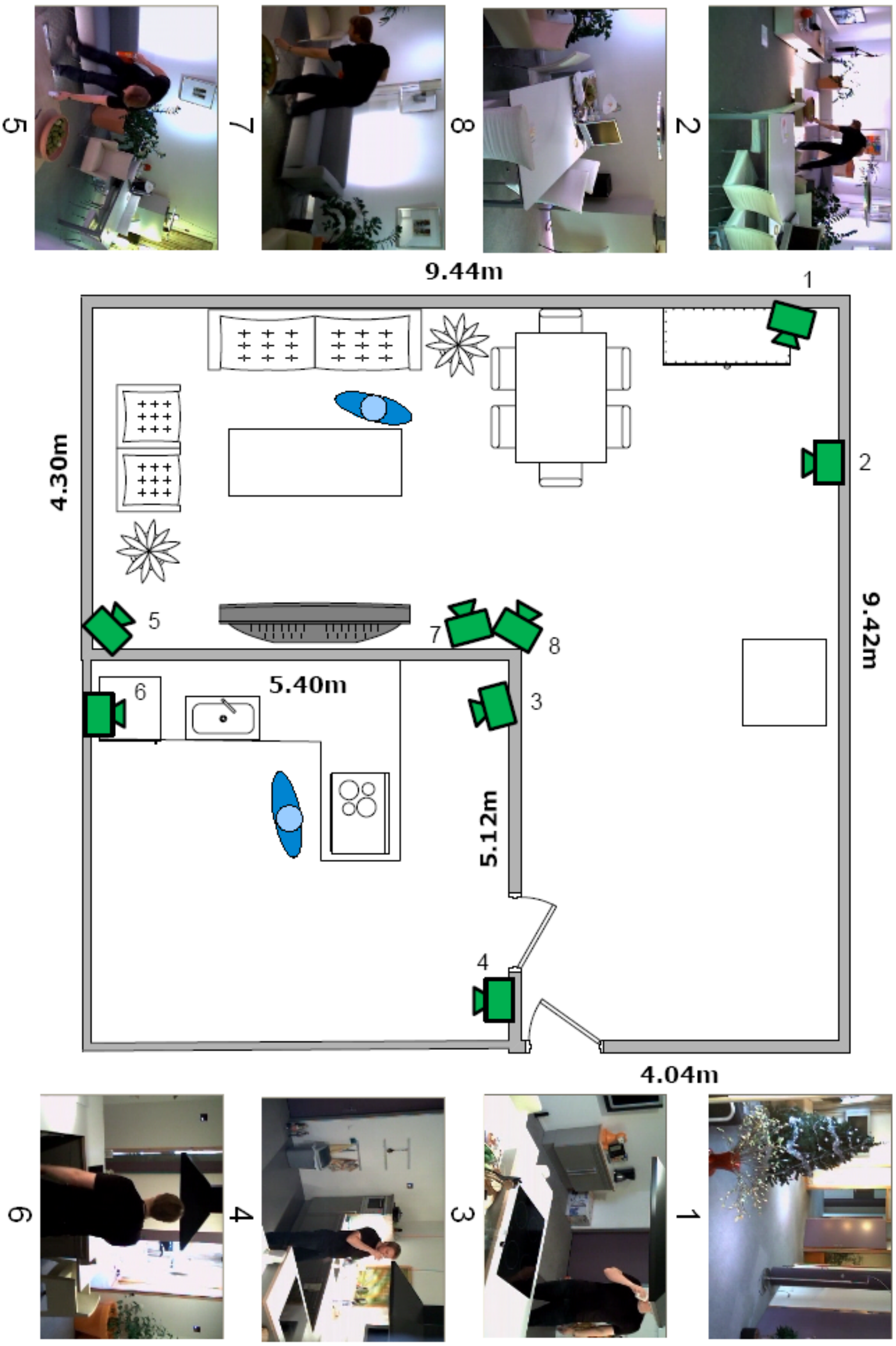


Figure 7.6: *Examples of the 8 partly overlapping views of the HOMELAB dataset with a person taking a drink in the kitchen (views 1,3,4,6), and later putting the drink on the sofa table (views 2,5,7,8).*

7.6 Experiments: Feature Selection

A small study was carried out to see how different features correlate within each of the datasets. The study should highlight the dependences there exist between feature values and the datasets from which they are computed. A feature that strongly correlates with the existing feature-set will not significantly improve the accuracy of classification, because it has most often little new discriminative information to provide. However, if the classifier is robust enough, even this small amount of complementary information can increase the overall accuracy. The direction of the correlation was not considered important, thus only the absolute values are reported here. The color scale of absolute correlation values from blue (0.0) to red (1.0) is shown on the extreme right in each correlation figure.

7.6.1 Full Datasets

The correlation between all the 29 features within the three datasets is shown in Figure 7.7 as separate correlation matrices. Some significant correlation, above 0.8 (orange color), exist in all of the datasets between few different image feature-pairs. Correlation is most evident between features that have connections in the manner in which they are computed, such as change of a dimension or aspect ratio of a bounding box. No other strong correlations exist within datasets. It does depend on the dataset which computationally connected feature-pairs show strong correlation.

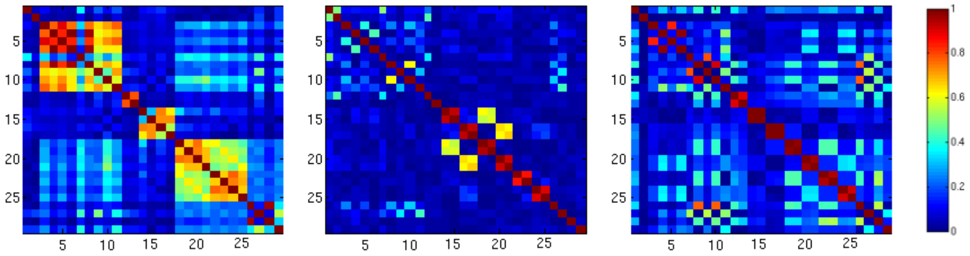


Figure 7.7: Feature correlation within three datasets, starting from left with HOMELAB, MAS and VIHASI.

7.6.2 PFA-Selected Datasets

Based on results shown in Figure 7.7 it is feasible to find feature subsets with low-correlation within a dataset. Therefore, the number of features can be diminished while maintaining a good classification accuracy. A technique called Principal Feature Analysis proposed in [124] was used to find the candidate subsets of features. PFA aims to select features that maintain as much as possible of the variability and spread of the original data from the features given by principal component analysis. The candidate feature subsets were computed for nine different numbers of features: 3, 5, 7, 9, 11, 13, 15, 17, and 29, and for each of the datasets and the combination dataset MAS+VIHASI.

The correlation between the PFA selected candidates of entire HOMELAB dataset is shown in Figure 7.8a with corresponding examples for the entire MAS and VIHASI datasets in Figure 7.8b and Figure 7.8c. The PFA-selected subsets of features offer reduced correlation. For all the datasets a subset with no significant absolute correlation, < 0.6, could be formed until 9 features, when increasing the number of features starting from 3. Between HOMELAB and MAS datasets some overlap in selected features exist. However when examining all three datasets, the overlap is smaller.

In conclusion, in all the datasets a feature-set of 9 features provides already classification results close to what can be achieved by using all the features. However, these features need to selected specific to the environment.

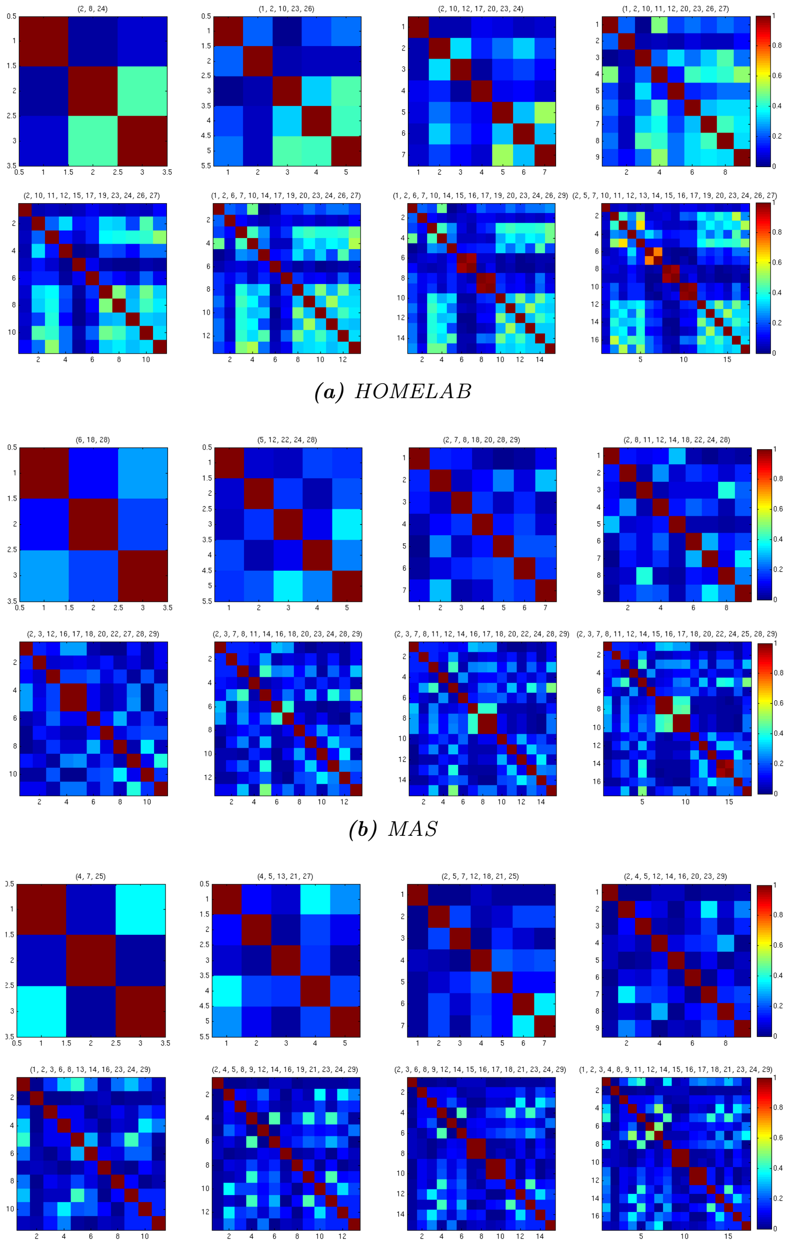
7.6.3 PFA-Selected Node-Specific Datasets

Optimizing the number and type of features used in each of the splits in the classification tree, see Figure 7.2, can provide better classification accuracy. The PFA can also be performed individually in each branch of the classification tree. The resulting candidates are then compared by examining the fit of the model, trained by the candidate subset, to the training data. The candidate whose model fits the training data the best is expected to provide similar accuracy with new data. It should be noted that verifying classifiers by training error might lead to overfitting. In such a case the training data is very well modeled, but the classifier can easily fail with new data.

See Figure 7.9a for the performance of candidate feature subsets selected by PFA on different nodes of the tree for the MAS dataset. The similar experiments for VIHASI, HOMELAB and combination MAS+VIHASI dataset can be see in Figure 7.9b, Figure 7.10b and Figure 7.10a.

For the MAS dataset it can be noticed that for Node 4, in Figure 7.2, the best fit is offered by a 15-feature subset, and for Node 5 by 17 features. Also with VIHASI dataset Node 4 fits best with 15 features. For HOMELAB there is no better fit for any of the nodes than with the full 29 features. Notice that HOMELAB dataset does not contain running action, so there is no need for classification in Node 2. With the combined dataset a slight improvement could be expected when using 17 features for Node 4.

All in all, the expected gain when compared to using the full set of 29 features appears to be small. Overall, the performance of CRF as a temporal model does not decrease even when adding redundant features [122].

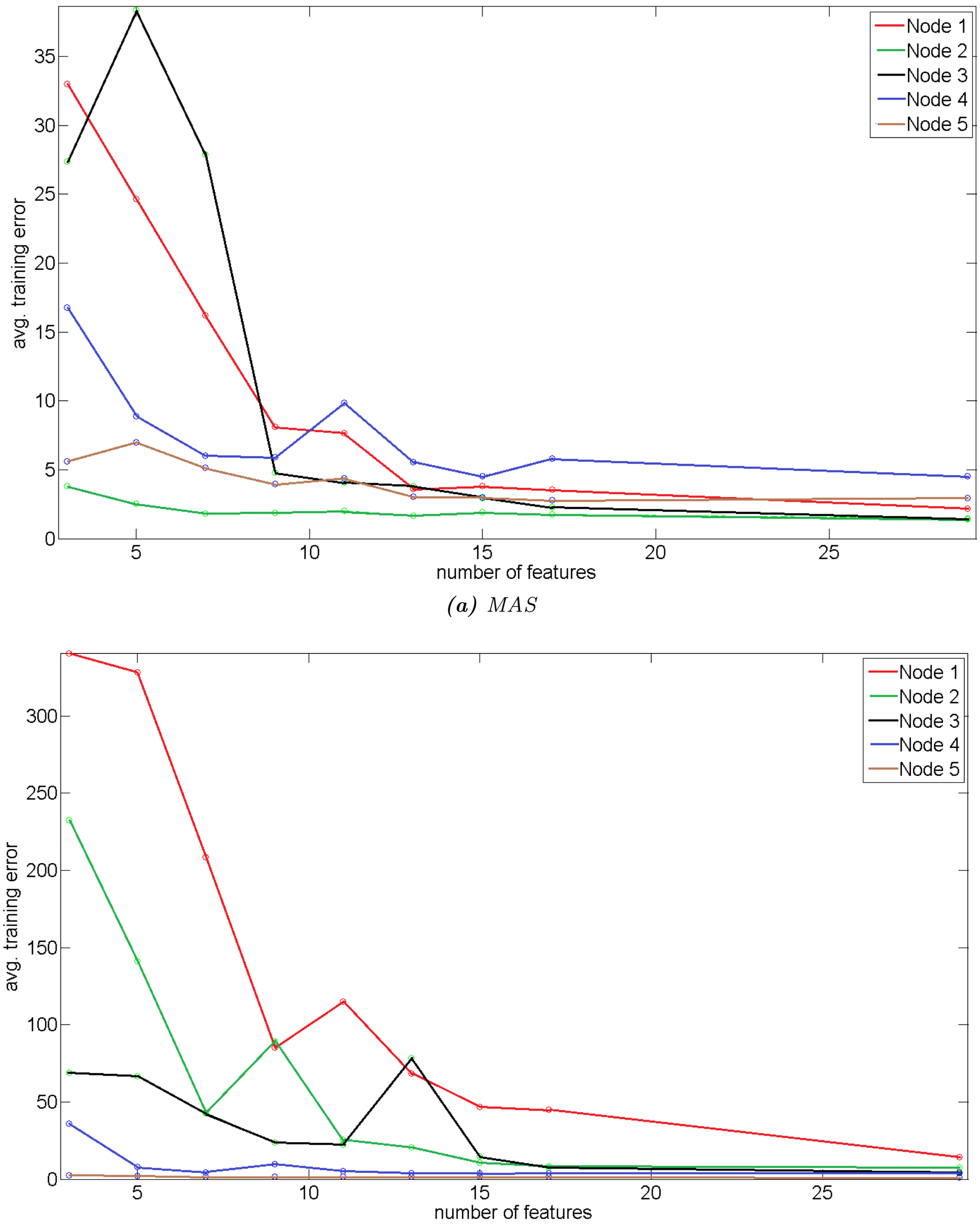


(a) HOMELAB

(b) MAS

(c) VIHASI

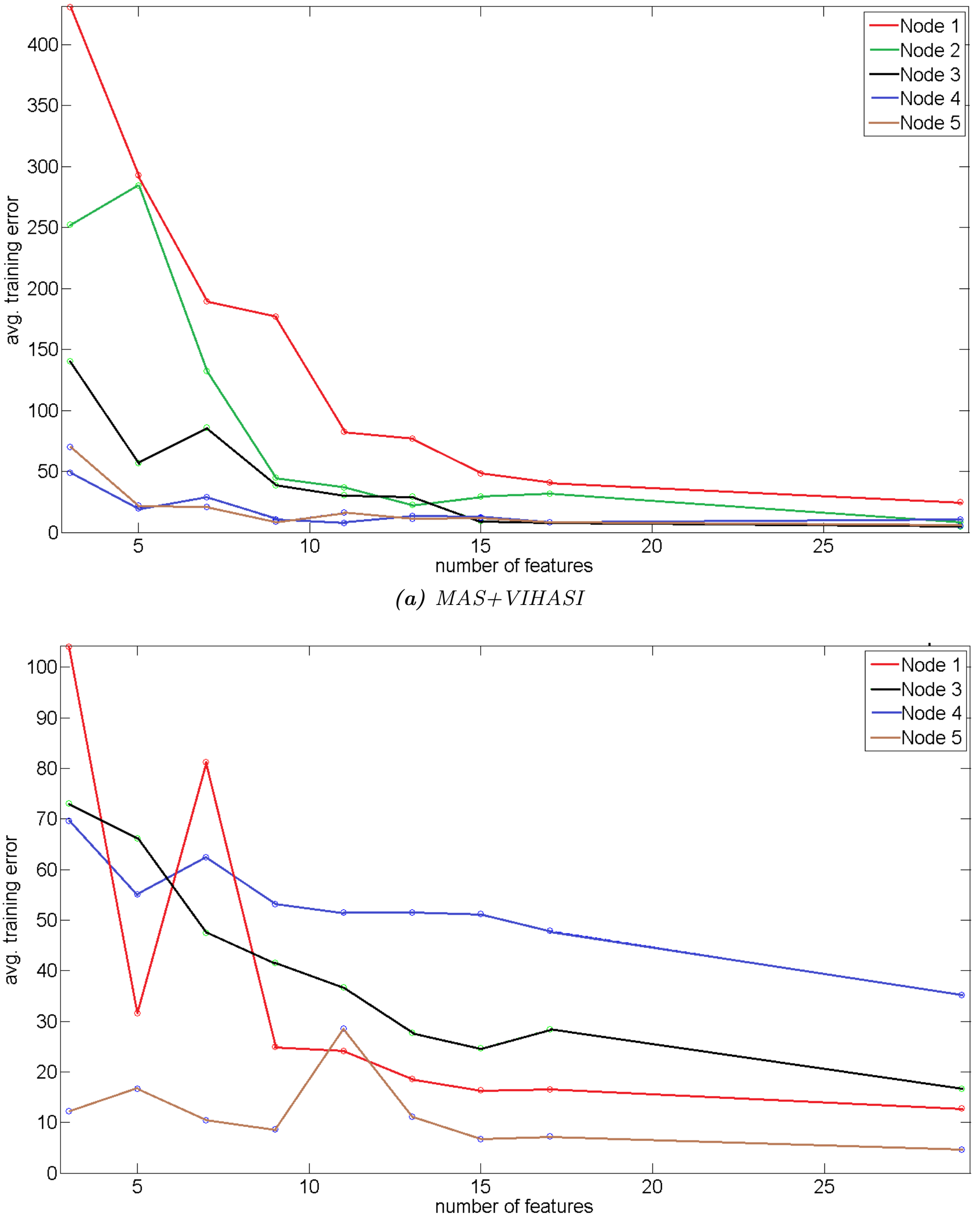
Figure 7.8: *Feature correlation of entire (Node 1) dataset for PFA selected candidate feature subsets of 3,5,7,9,11,13,15,17, and 29 features. The indeces of selected features are shown in top of each sub-image.*



***(a)** MAS*

***(b)** VIHASI*

***Figure 7.9:** Fit of the trained 1D CRF-model for different feature-sets and for each of the tree nodes in terms of training error.*



(a) MAS+VIHASI

(b) HOMELAB

Figure 7.10: *Fit of the trained 1D CRF-model for different feature-sets and for each of the tree nodes in terms of training error.*

7.7 Experiments: Multi-View Scenarios

All the five camera scenarios from section 7.4 were tested with all three datasets from section 7.5. The accuracy values presented are averaged results of leave-one-person-out cross-validations; for each person the data from him/her was used to test the model trained by the data from all the other persons. For results on each dataset-scenario combination see Table 7.5.

7.7.1 Activity Recognition Results

With three out of the four datasets a minor improvement over SView scenario is already visible in the KView scenario that selects a single view based on criteria. Even though the silhouette-based criteria for selecting the best view do not guarantee that the POV with the most accurate classification is selected, KView performs better than restricting to a single view.

A greater improvement is achieved over all the datasets when using information from all the POVs. When considering the ideal silhouette datasets MAS and VIHASI the CView outperforms PViews, on average by 2.5%. With visibility and noise affected silhouettes of HOMELAB and with combined dataset having greater variability in the actions and provided views, it is PViews that performs better on average by 1.6%.

HOMELAB is the most difficult environment to classify in; the best scenario PViews reaching accuracy of 69.1%. There are three major factors that cause this. First, the silhouette extraction used in the experiment could not provide full clean silhouettes. Second, people were not fully visible during the entire routine, as they were sometimes occluded by other objects such as tables and chairs. Also the carried objects, book and cup, deteriorate silhouette quality by including non-human pixels to foreground. Third, sometimes people only partially fit in the field-of-view (FOV) of a camera. In Figure 7.6 one can see examples of encountered occlusions and FOV constraints.

7.7.2 Separation Of Activities

To provide more details on classification accuracy per activity either a PViews or CView confusion matrix for each of the datasets is given in Table 7.4.

The greatest difficulties with MAS dataset are in separating standing-interacting from standing. VIHASI shows more confusion between walking and both standing activities. The combined dataset shows in addition to these cases more confusion with the running activity. Confusion exist between several activities with the erroneous silhouettes of HOMELAB, worst effect is seen with standing detected falsely as sitting on various occasions.

Table 7.4: Example confusion matrices from different datasets summed over all the actors of the particular dataset for the particular fusion scenario.

MAS:	CView					
	Walk	Stand	Sit	StandInt	SitInt	Run
Walk	534	0	0	1	0	19
Stand	0	199	0	14	0	0
Sit	0	0	98	0	11	0
StandInt	7	71	0	967	30	0
SitInt	0	0	22	0	233	0
Run	20	0	0	0	0	273
avg. acc. [%]	92.2					
VIHASI:	CView					
	Walk	Stand	Sit	StandInt	SitInt	Run
Walk	415	0	0	5	0	0
Stand	92	328	0	0	0	0
Sit	0	0	25	0	0	0
StandInt	83	8	4	839	22	4
SitInt	0	0	0	0	110	0
Run	0	0	0	1	0	659
avg. acc. [%]	91.4					
MAS+VIHASI:	CView					
	Walk	Stand	Sit	StandInt	SitInt	Run
Walk	801	0	0	48	2	123
Stand	138	495	0	0	0	0
Sit	0	0	122	0	10	2
StandInt	268	233	0	1302	39	193
SitInt	0	0	87	0	278	0
Run	5	0	0	30	0	918
avg. acc. [%]	80.2					
HOMELAB:	PViews					
	Walk	Stand	Sit	StandInt	SitInt	
Walk	8807	116	110	1613	324	
Stand	90	412	715	53	215	
Sit	0	195	2664	21	332	
StandInt	984	1008	813	3341	1504	
SitInt	163	9	75	166	724	
avg. acc. [%]	69.1					

Table 7.5: Performance on all the four datasets with the five camera scenarios by the optimized classifiers.

MAS					
Scenario	SView	KView	PViews	CView	OView
acc/person [%]	81 81.5	83 82	90 89	92 92	94 96
acc/view [%]	77.5 85				
avg. acc. [%]	81.3	82.5	89.5	92.2	95.3
VIHASI					
acc/person [%]	76 76 74 78 78	76 79 76 76 80	91 87 85 88 92	92 88 88 93 96	99.8 100 99.8 100 100
acc/view [%]	73 79 81 79 75 76 75 78 76 73 75 77				
avg. acc. [%]	76.4	77.4	88.6	91.4	99.9
MAS+VIHASI					
acc/person [%]	66 71 69 73 71 59 57	71 76 76 77 75 59 62	84 87 83 88 83 70 68	84 85 82 89 85 71 65	98.70 99.6 100 99.4 98.8 80.6 79.8
acc/view [%]	64 73 75 77 72 71 64 71 72 68 66 55 61				
avg. acc. [%]	69.2	71.1	80.6	80.2	93.8
HOMELAB					
acc/person [%]	67 55 60 65 64 70 61 51 73 59	57 55 49 57 64 61 59 59 52 66	80 63 64 71 72 71 72 53 71 73	73 59 63 66 71 69 69 58 66 68	90.5 89.7 93.8 90.4 90.8 97.4 96.1 92.5 92.2 97.1
acc/view [%]	70 72 71 39 80 38 70 61				
avg. acc. [%]	62.6	57.8	69.1	66.2	93.4

7.8 Discussion

A silhouette-based, multi-view system for recognizing human activities is an interesting case for testing how fusion on different fusion levels and with different fusion methods performs. The proposed vision system relies on three different categories of image features and on temporal modeling by CRF to distinguish between 5 to 6 different activities.

Application
Experimental results demonstrate that good classification accuracy, above 91%, can be achieved by the proposed system when dealing with clean conditions as provided by MAS and VIHASI datasets. With the combined dataset MAS+VIHASI accuracy above 80% was still achievable. With the more challenging dataset of HOMELAB an accuracy of 69% could still be reached.

A few important computer vision problems and limitations were encountered with the HOMELAB dataset. Because the person is not always fully in the FOV of the cameras and occlusions with other objects exist, it may be impossible to continuously provide clean silhouettes of the person. Whatever the feature representation, there will also be variability within the same activity class. Variability is due to the differences in appearance, the pace of action of different users, and the changing relations between the users and the observing camera system.

On Fusion
The focus of the experiments was to compare different ways of combining data within a local vision network. Five camera fusion scenarios were introduced: SView, KView, OView, PViews and CView. These scenarios were expected to represent good starting points for further development and optimization for multi-view data handling within the context of temporal activity modeling. The five proposed camera fusion scenarios were examined for dealing with the inherent variability in features across the datasets. Lets consider the fusion results of this chapter in relation to the proposed vision fusion framework (VFF).

Fusion Approach The human activity was categorized as one of the six classes based on the features computed from the silhouettes given by a varying number of cameras. None of the classes could be considered more probable. There was no preference to be more sensitive to a subset of the classes, and none of the cameras was considered more ideal for detecting certain activities. Therefore, no preference to detecting certain aspects better than others was included in the vision network. Increasing common certainty in the observations made was the primary goal of fusion [Design Rule 1].

Vision Network All the cameras used in the individual datasets were the same models with the operating settings as the same for frame-rate and resolution. The automatic gain and white-balance options were disabled, and the same settings were given manually to each camera [Design Rule 2]. For example, if the camera gain is adaptive and the intensity of the natural light entering the room changes, the camera can easily fail in image segmentation as the assumed static background does no longer exist. As a consequence, the features would be compromised. Depending on the fusion approach, this unbalance in the data could badly affect the accuracy of activity classification.

The data was transmitted over the same medium (USB/Ethernet) from all the cameras. No time stamping on frames was implemented, as the local network was considered stable. The same algorithms were applied to each video stream and their settings were matched to each other [Design Rule 3]. For example, if the assumption of the stability of the local network would have been false, the synchronous delivery of data would have been compromised. These problems would have made the system incapable of correctly classifying in the presence of fast changes between activities and activities that occur over a short period of time, such as standing-interacting activity.

Fusion Architecture There exist no resource limitations on MAS and VIHASI datasets as these had been captured within a controlled environment. A distributed camera recording system was used in gathering the HOMELAB dataset. In all the experiments a fully centralized fusion architecture was used to gather and combine the data. No prior information existed that would have justified the combination of a subset of cameras. No analysis was performed that would have pointed out similarities between the observations among all the cameras [Design Rules 6 and 7].

Fusion Level The fusion was studied on two fusion levels: features and decisions. A similar common opinion fusion method was applied on both fusion levels, this making the results between fusion levels comparable.

Fusion of features (CView scenario) produced the best, most consistent accuracy of 91.4-92.2% with environments that have a similar setup for all cameras and that have people behaving in a similar manner. Whereas, with environments with more variability such as HOMELAB and MAS+VIHASI, fusion of decisions (PViews scenario) provided slightly better decision accuracy.

This would seem to imply that fusion in the feature-level outperforms fusion in the decision-level within more controlled settings. One should notice though that the differences were small (2-4%) but consistent across the datasets [Design Rule 8].

Fusion Method The five camera fusion scenarios studied were based on sensor selection and voting, and linear logical methods. The first set of fusion methods relied on selection of a single view. The SView scenario was used as reference of a single-camera system performance. KView presented an attempt on how a three-part silhouette-based criteria in selecting a single view out of many views, would increase performance. KView did offer 1-2 % better performance than SView, but much more could still be achieved even by single-camera selection like was shown by OView. The numbers with ideal view-selection by OView scenario show that accuracies up to 93 and 99% could be reached by camera selection alone. The quality of the on-line criteria can thus have huge impact on fusion method accuracy [Design Rule 11].

A second set of fusion methods relied on combining multiple estimates into a joint estimate. Two fusion scenarios, CView and PViews, were introduced. CView combined multiple feature streams into a single stream, by approximating the minimum mean square error estimate by the arithmetic average. PViews combined decisions from activity classification of individual views by majority voting.

Both of these combination methods: MMSE feature-transformation and majority voting provided better accuracies in all of the datasets when compared to the practically implementable sensor selection methods (SView or KView). Instead of selecting only one source of information, it appears to be better to use all of the available data, even with some erroneous data among it, in jointly making a decision. Therefore, by finding the consensus between cameras, the fusion method seems to better cope with data that is affected by some imaging noise or viewpoint limitations [Design Rule 10].

In summary The findings of the experiments on fusion scenarios can be summarized as follows. The results show that in reasonable conditions fusion methods that rely on combination of data outperform methods of selection. Selection-based fusion methods can provide very good results, but they strongly depend on how good of a selection criterion is used, and how well this criterion adapts to different environments. Furthermore, fusion of features outperforms other presented scenarios within more controlled settings. However the more variability exists in camera placement and characteristics of people, the more likely improved accuracy can be achieved by combining decisions in silhouette-based multi-view activity recognition.

Chapter 8

Detection of Repetitive Behavior

Many interesting types of behavior are characterized by repetition of actions such as certain activities or movements. A generic methodology to classify and detect repetitions that may occur at different scales or between different types of actions is introduced in this chapter [1]. The proposed method is called Action History Matrices (AHM). The properties and adaptability of AHM for detecting repetitive behavior are demonstrated in analyzing customer behavior in a smart shop application.

Section 8.1 starts this chapter by discussion on application background by covering some of the traditional modeling approaches to behavior patterns. Instead of modeling specific patterns of action the focus is on generic repetitive behavior, for which the AHM method is developed in section 8.2. The architecture and methodologies of the proposed vision system are given in sections 8.3 – 8.4. The features and classifiers used in classifying repetitive patterns are presented in section 8.5. The two different experimental datasets, recordings in a shop environment, and motion path simulations, are illustrated in section 8.6. Finally, the classification results for simulated data and the fusion results on the real data are shown in section 8.7. This chapter concludes in section 8.8 with the discussion of the major findings in relation to the vision fusion framework.

[1] This chapter is (partly) based on:
Määttä, T., Härmä, A., Aghajan, H. (Submitted 2012). *Collaborative Detection of Repetitive Behavior by Multiple Uncalibrated Cameras.*

8.1 Introduction and Related Work

Vision networks with multiple cameras offer valuable tools to cope with covering vast areas and observing people in various environments. A single camera might provide enough information to detect behavioral patterns with a reasonable accuracy, but having other views providing simultaneously data from other perspectives is expected to improve the accuracy. The information about the activities and behavior of the subjects can be used to control various services, e.g., in smart homes or shops, or to collect important events, e.g., in care homes or hospitals.

8.1.1 Behavior Modeling

Behavior is often defined as observed actions or reactions of a person or animal in response to external or internal stimuli [125]. *External* stimuli relates to interaction of a human with the environment while *internal* stimuli relates to the task objective and the emotional and physiological state of the subject. The focus of this chapter is on the development of a behavior recognition system, which estimates the internal and external stimuli that drives the behavior. The generic behavior system is illustrated in Figure 8.1.

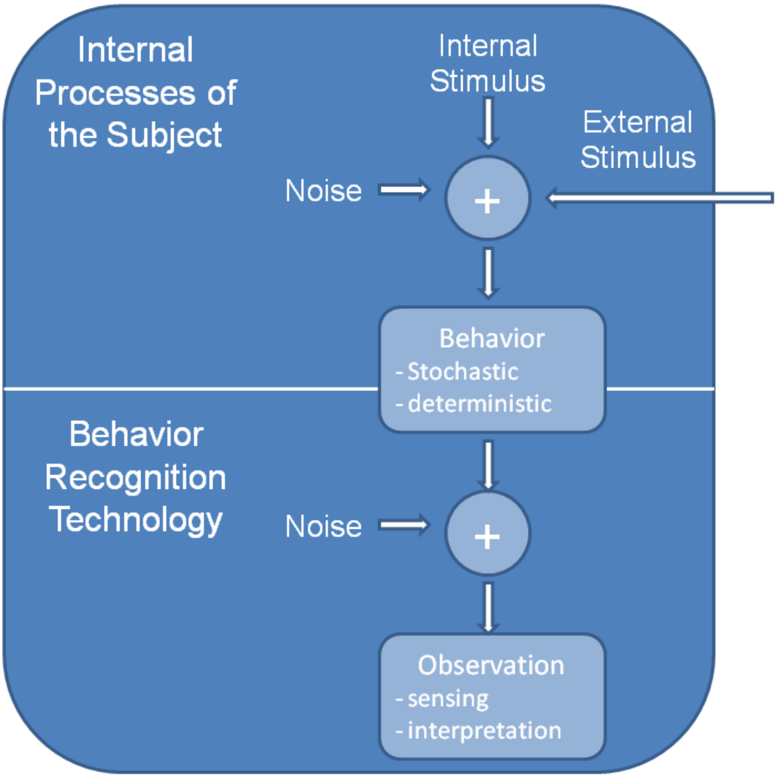


Figure 8.1: *The definitions and relations of behavior and the technology monitoring it.*

In an ideal situation for detection a stimulus would always lead to a deterministic behavior. However, in the real world observed behavior also has the stochastic component which depends on many internal and external factors. One may roughly model the stochastic component of behavior as a response to a noise component that is added to the stimuli. In the behavior recognition system, see the lower part of Figure 8.1, there is another noise component which is related to the resolution, distortions, and additive noises in the environment and the sensors. The design of a behavior recognition system should be optimized such that it eliminates both noise components from the final interpretation of the observation and extracts the deterministic part.

The basis for a behavior recognition system is a mathematical model of behavior. In psychology, mathematical models of behavior have often been associated with the *behavioristic* [126] research tradition best known for the famous works of Pavlov. Most of the early mathematical models were related to the regulation of behavior, for example, through reinforcement or punishment [127], or input-output relations in reaction to external stimuli, see [128,129]. Some of more recent work includes the use of differential equations [130], dynamic programming [131], and sequential pattern recognition methods [132] for complex behavior modeling.

Repeated patterns are most often modeled using graphical models [133] such as Markov models which translate the sequential behavior into state transition probabilities. The graphical models often require large training data which may not always be available. The deterministic part of behavior can be considered as a program that the subject is executing to complete a certain task [130, 131]. However, there may be many actual programs a subject can execute to complete a task [134]. A graphical model consisting of several state transitions cannot necessarily handle this large variability in programs to complete the same task.

8.1.2 From Behavior Patterns to Generic Repetition

The literature on automatic detection of activity patterns mostly focuses on rigid definitions of the patterns. In [135], the activity patterns are discovered by first finding the frequent sets of actions, followed by a search of topology and time relations, and finalized by identifying additional conditions, such as time of day and other contextual information. Benabbas et al. [136] introduced a method where the global motion patterns based on optical flow fields were used to find the most usual patterns and their magnitudes, and further using these in modeling and detecting usual group behavior scenarios such as group dispersion and evacuation. Most often motion trajectories are used in detection of unusual patterns of motion. The common approach is to learn the usual motion patterns and use them in separating away unusual patterns, as was done in [137] by training and applying a hierarchical classifier.

Action repetitions are common in many cases where people are confused, get lost, and need assistance. Therefore, it is also important that a smart monitoring system is able to detect those states. However, in a rich environment the action repetition can take many different forms and cannot necessarily be detected based on a comparison to usual or unusual motion trajectories or activity patterns. Focus of this chapter is on the detection of action repetition *per se* without a need to make a clear distinction between different types of repetitions.

An action may take place at the global or the local level. Movement that is generated by a person moving through an environment is called *global* motion. *Local* motion is defined as motion of the limbs, body and head when subject is e.g. interacting or adjusting himself. In case of both global and local motion, the behavior is represented by repetition or sequential execution of different actions. The main tool for discovering the repetitive action patterns is based on the analysis of the transitions between the actions. The main application context in this chapter is the behavior of a customer in a shopping environment. It is assumed that when customers are confronted by a situation where they are uncertain how to proceed, they often start repeating themselves. For example, a customer may be staying longer periods within the same area, returning to previously visited area, or alternating between two or more previously visited areas. A smart environment can detect such behavior and trigger means to assist the customer.

8.2 Behavior by Transitions: Action History Matrices

A behavior modeling approach is proposed, which allows to work with more limited training data sets and to deal with the variability in the deterministic programs. At the same time the approach allows the inclusion of some insights about the environment into the model. It is assumed that the essential characteristics of behavior (i.e., the program) related to an internal stimuli can be observed from the statistics of the state transitions.

Considering a shopping example where a person who cannot find the right product category alternates between the activities of walking around the shop and stopping to scan the selection in different parts of the shop. However, a subject who has difficulties of choosing between two or more products may walk between a small number of locations and alternate between walking and close examination the products. In both cases the observed complex behavior consist of relatively simple state alterations which can be successfully detected from a camera image.

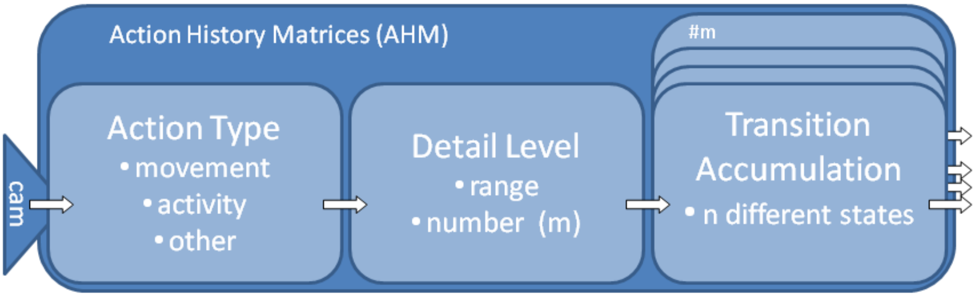


Figure 8.2: *The three-tier approach of Action History Matrices.*

8.2.1 AHM - Definition

A transition matrix is a statistical representation of transitions between different states of the subject during an observation period. Transition matrices have been used earlier in behavior modeling [138], but this chapter aims at a somewhat broader definition of a methodology, enabling various types of data, such as activity and movement information, to be collected in a matrix form. The transitions are accumulated in a matrix, denoted here as

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & .. \\ & .. & \\ & .. & a_{nn} \end{bmatrix} \tag{8.1}$$

where each element a_{ij} represents the number of times the person has moved from cell i to cell j during the observed period. The basic idea of using Action History Matrices, AHM, is illustrated in Figure 8.2. For the construction of AHM one has to define three different characteristics of the data:

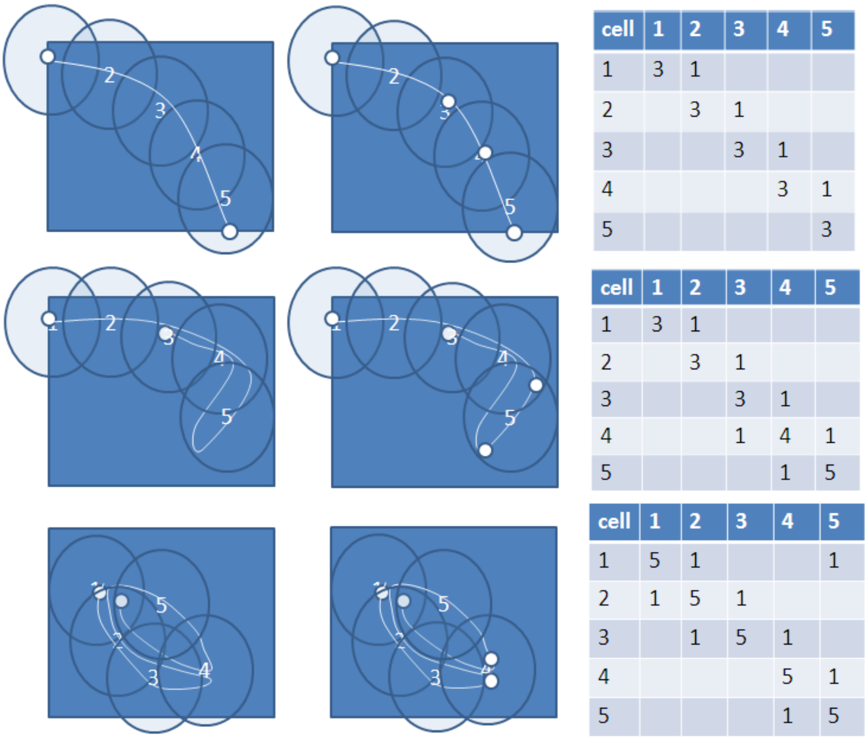
- *type of actions-of-interest*: movement, activity etc.
- *level of detail*: spatial range, time range etc.
- *mechanism of data accumulation*: pre-defined constraints

The type of the actions-of-interest can be defined, e.g., as movement on the floor level. Regardless of chosen action-type, the goal of AHM is to provide analysis of the actions-of-interest simultaneously at multiple scales. Scales include, e.g., different spatial ranges for quantizing space and different lengths of time over which activities are analyzed. The level of detail is thus specified as the range and number of sampling levels used to capture the actions-of-interest. The manner in which data is accumulated can be directed by additional constraints; criteria that action needs to fulfill in order to to be included as part of the accumulated data.

8.2.2 AHM - Movement Example

As a demonstration of the aspects of AHM, let us consider a case of global movement. The transitions between various physical locations of the user in the observed environment are captured from the camera data. In this chapter the location data is quantized to a small number of local regions, cells, which are circular neighborhoods defined in the image plane. When a person enters a new area, a new cell is created to cover that location and the close proximity. This representation of the location data provides a flexible quantization of the space and does not require calibration of the camera system.

Six example scenarios of movement-based action analysis are shown in Figure 8.3. The radius of the cell corresponding to the level of detail is fixed to a single size. No additional criteria for accumulating transitions are included. Thus, regardless of the person making intermediate stops while moving around, the recorded transitions are the same (left and middle column). On average, the person takes a time equivalent to 3 consecutive observations to travel across a cell; see especially top-right matrix of Figure 8.3.



cell	1	2	3	4	5
1	3	1			
2		3	1		
3			3	1	
4				3	1
5					3

cell	1	2	3	4	5
1	3	1			
2		3	1		
3			3	1	
4			1	4	1
5				1	5

cell	1	2	3	4	5
1	5	1			1
2	1	5	1		
3		1	5	1	
4				5	1
5				1	5

Figure 8.3: *Action History Matrices (AHM) examples on motion: top row) person advances directly, middle row) person returns to previously visited cell, bottom row) person loops between previously visited cells . Left column) without stopping, middle column) with intermediate stops, right column) recorded cell transitions in matrix form.*

In general, the cells are created independently per each view; no correspondences between cells from different views exist. Additionally, the cells are directly laid on the image plane without any geometrical corrections based on the part of the scene they happen to overlap with, such as a floor or a wall. With this approach it is not necessary to calibrate the multi-camera system. This is one of the benefits of the AHM methodology. However, if the camera network was calibrated, two opportunities do arise. First, a generated cell could be rotated to match the inclination of the scene plane it overlaps with, so that the absolute scene-area each cell occupies is the same. Second, the semantic labeling of areas-of-interest can be included into AHM. Each location, defined by a cell, could be given a broad semantic label which, e.g., in a shop environment could include the identification of the entrance to the shop and the cashier region.

8.2.3 AHM - Extension

The example scenarios discussed above were using movement patterns as type of actions. Another set of action transitions may take place between different activities.

Activities of interest may include local scale activities, such as 'picking-up-item' and 'closing-a-drawer', or activities of more global scale, such as 'making-a-purchase' and 'arranging-a-closet'.

With activities, a similar multi-resolution approach can be used by gathering observations over periods of different duration of time. Depending on the defined time intervals, observations can be made at various levels of detail. The way in which the transitions between activities are accumulated can be further constrained, e.g., to include only those transitions that based on a prior information may exist. For example, a person that sits can not transfer to walking directly; he must have stood up (stand-up activity) in between.

8.3 System - Architecture

The three-level system approach that includes the proposed behavior recognition detailed in section 8.2 is shown in Figure 8.4. The fusion of data can be performed at any level of the hierarchy, but can also be influenced by feedback from other levels into the fusion mechanism. For example, feedback may contain information about system performance and stability of results.

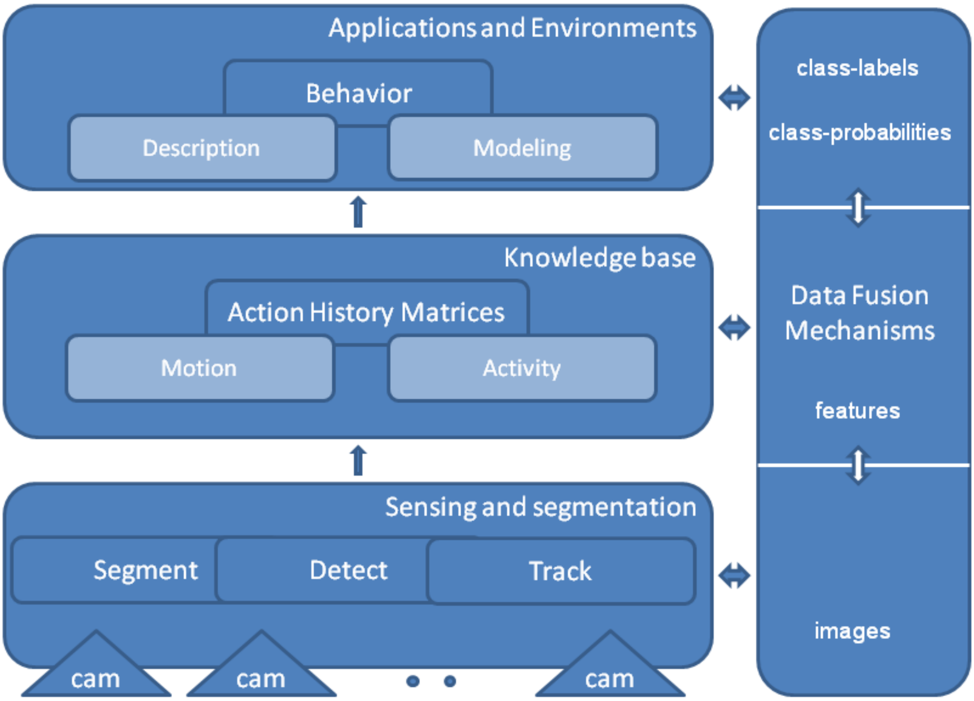


Figure 8.4: *Behavior Recognition Technology: A three-tier layout of a generic behavior recognition system, in this case based on visual cues. Adjacent to the classical modules there are the fusion mechanisms, separately at each level, but contributing together towards improved system accuracy, at each level and as a whole.*

8.3.1 Fusion Opportunities

In the proposed system there are two aspects in which fusion of data becomes an immediate opportunity. First, the motion path cell network was created at three different levels and therefore it is possible to investigate how different resolution levels perform in different conditions. Second, the same motion paths have been observed from different viewpoints, which make it possible to fuse the data between the views. Therefore, data fusion can be performed between different resolutions levels of data or between the camera views. The detection of transitions at multiple resolution scales should provide increased sensitivity to out-of-the-ordinary events without harming the detection of usual events. Fusion between multiple cameras should similarly help in not missing hard-to-detect events, and should provide increased certainty to the observations made.

At the lowest level, different images, such as multi-resolution or from different viewpoints, can be combined into a single image. A multi-resolution approach in spectral domain through wavelet- and fast Fourier transforms has been studied, e.g., in [139, 140] and a general framework for multiple input sensors has been proposed, e.g., in [141]. At the second level features computed from the images, and at the highest level the action and/or behavior labels, can each be further combined when having multi-resolution and/or multi-sensory data. In [142] and [143] fusion mechanisms in the context of activity recognition were explored for fusion of feature values and activity labels.

8.3.2 Chosen Fusion Approach

The data is gathered in a fully centralized manner such that a central system receives the data from each of the cameras. The fusion is performed at two different data-levels. in order to produce a joint estimate, the data is combined either at the feature or at the decision-level. These two levels have been commonly compared in the literature and therefore the two levels were chosen for the current experiments.

The two basic fusion mechanisms of selection and combination are studied. In the selection, a single data entity or a subset of entities is chosen to be used in further analysis. In combination the entities at the certain level, as in features or labels, are combined into a single value best representing the original entities. Three different approaches to fusion methods that perform the actual combination of data are explored:

- *majority approach*: computes the overall opinion between views
- *sensor selection*: selects the view with best coverage on person
- *ideal selection*: selects the view (if any) that gives the correct result

Majority approach computes the minimum mean square error, MMSE, estimate for candidate features as the final feature, or defines the final decision as the most common class label among the candidate decisions. The sensor selection approach selects the candidate for feature or decision from all the available cameras based on coverage on the person. The view that has created most cells and has accumulated the most of transition observations is used as the source of data. In contrast, the ideal selection selects the view that has the correct label for the sequence if any of the views has the correct label. These fusion approaches will be discussed in the experiments in terms of classification accuracy per sequence.

8.4 System - Person Tracking

The initial challenge in person monitoring is finding out the location and shape of the person. Many approaches have been proposed earlier for human detection and tracking [144–146]. Typical differences in these methods concern in accuracy, complexity and performance.

In this chapter a hybrid approach is used in which two complementary algorithms work independently but exchange some decisions to support and/or correct the results of the other algorithm. For this purpose silhouette segmentation was combined with template-based shape matching. This solution is illustrated in Figure 8.5, and specifics are discussed in the next subsections.

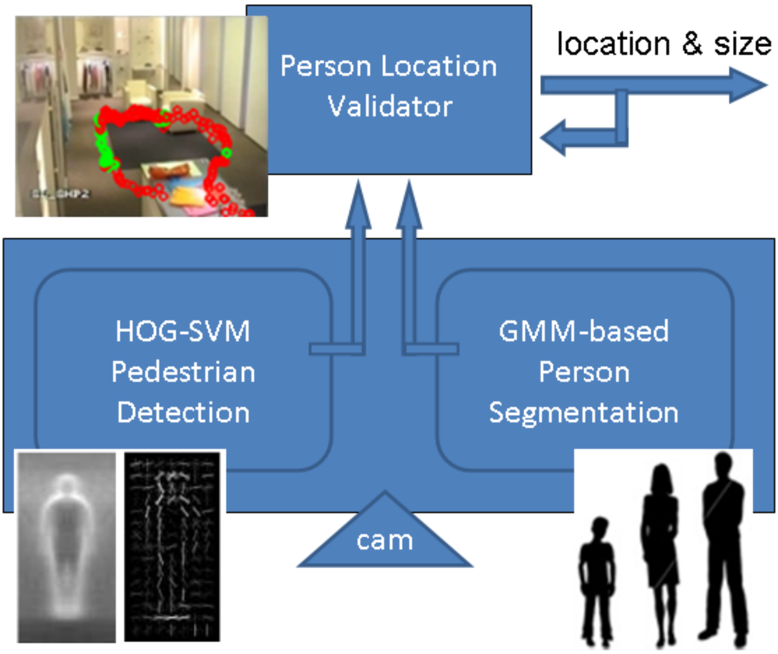

Person Location Validator
location & size
HOG-SVM Pedestrian Detection
GMM-based Person Segmentation
cam

Figure 8.5: *Person Tracking: Hybrid approach to detecting people.*

8.4.1 Silhouette-aided Pedestrian detection

The first task is to segment a person within the image. This is achieved by combining a Gaussian Mixture Model (GMM) background subtraction based silhouette extraction [147] with a Histogram Of Gradients (HOG) template based Support Vector Machine (SVM) pedestrian detector [148]. HOG is used to initialize the detection and as the main authority, if it gives a hit. GMM is used to validate the HOG given pedestrian hits and to update a person's location, when HOG misses the person.

Another possible option to detect people is to use face-detection, [144]. The main inherent problem of these approaches is view-dependency. Only frontal and side-profiles from moderate distances can be detected with confidence. The proposed hybrid method presented in this chapter is very robust against changes in scale and orientation, once an initial hit has been found. In the proposed method the robustness is achieved by temporal data association, and by combining the two mentioned techniques for detection: GMM and HOG-SVM.

Contextual information is exploited in maintaining the track and simultaneously relaxing the computational demands and increasing accuracy. Some methods exploit context further, such as in [145]. In the proposed approach a simple location and size context, defining the search window for next location within a certain range from the previously detected location, was used. This approach also seems to provide an uninterrupted track of a person as long as the subject is visible in one of the views. However, the proposed method does not address the problem of association of multiple individuals between multiple views. The matching of identities and handling crossing paths is left for future work. More intelligent approaches, such as the particle filtering based method for traffic monitoring proposed in [146] can provide smoother tracking results, but that level of accuracy is not necessary for the experiments described in this chapter.

8.4.2 Dynamically Created Location Cells

The second key component of low-level vision is to describe the detected motion of the person as seen on the image plane. For this purpose, the image plane gets divided into dynamically created cells each with unique ID. Cells are created and stored as a person moves through the scene. Dynamic cell generation is done at multiple resolution levels, each level having its own fixed cell radius. In this fashion, the coarse level having the largest radius and the fine level the smallest radius.

One direct benefit of creating cells on-the-fly and storing the transitions into a matrix is the fact that each cell-network is created person-specific. Thus the created cells capture the order of cells entered, and therefore can directly be used to see, e.g., if a person is still moving in original direction, or if he is back-tracing his/hers steps through areas already visited.

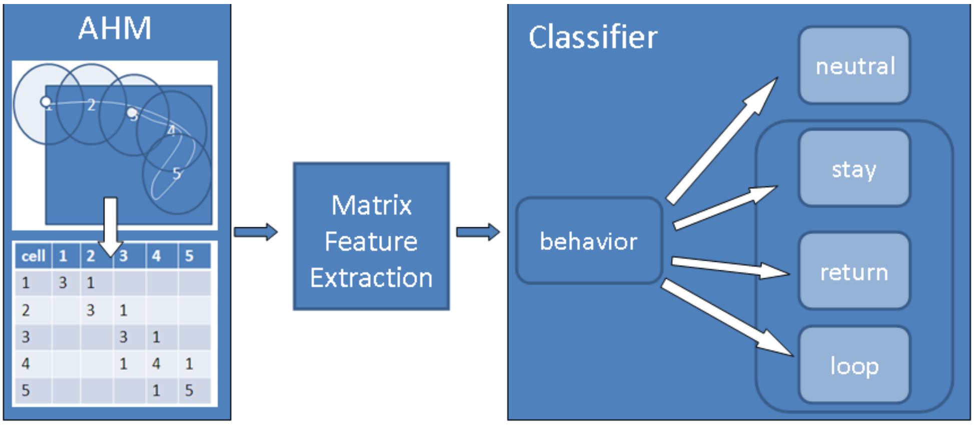
Having cells generated at multiple levels, or resolutions, helps in accounting for different granularities and in building up confidence in the findings. If certain motions are detected across multiple levels, very high confidence can be expected in behavior inference. Sometimes specific motions occur only at a larger scale. It would still be beneficial to detect the coarse scale repetition, but have less confidence in the result. In this manner a more comprehensive understanding of behavior can be achieved.

8.5 System - Transition Modeling

The aim is to model the transitions given by AHM into a set of classes-of-interest. The requirement is to detect and record all the transitions taking place, describe the recorded transitions with descriptive features, and make decision on the status of the person.

Corresponding to the case study, Figure 8.6 shows an example of repetition classification based on transitions between locations that is based on motion patterns recorded by AHM. Following feature extraction, the features are used in detecting if a person is behaving in a repetitive or neutral manner. The classes are:

1. *neutral*: makes a buy or a regular short visit
2. *stay*: stays longer within location(s)
3. *return*: repeatedly returns to a location
4. *loop*: loops between a few locations



cell	1	2	3	4	5
1	3	1			
2		3	1		
3			3	1	
4			1	4	1
5				1	5

Figure 8.6: *Transition Modeling: Proposed system applied to modeling a neutral pattern and three repetitive ones.*

8.5.1 Transition Types and Their Constraints

The transitions in the AHM can consist of, e.g., transitions from one location to another like from a counter to shop-exit, or from activity to another activity, like walking to interacting with an object. In these studies the focus is on location information in order to explain the approach in accumulating behavioral information.

After detecting and dynamically quantizing the location of a person, the manner in which the person moves between the cells needs to be described. For this purpose a transition matrix is utilized to hold the information of transitions between any of the cells. There are two ways to accumulate location transition information:

1. record all the transitions detected
2. record only transitions that meet a certain pre-defined criteria

A pre-defined criterion can, for example, require the person to stop in a location for the transition to be recorded. However the aim of criteria is always the same, to provide the cleanest and most efficient data for transition analysis. For this approach to perform well, the type of transition criterion, such as from standstill to standstill, and how robustly the criterion can be computed becomes crucial. Due to these factors every transition between cells was recorded without constraints in the presented experiments.

8.5.2 Transition Features

A way to approach the modeling effort is to describe some properties of the accumulated transition matrix. These properties can be used as features, on which the discrimination process of different behavior classes is based on.

An interesting option would be to exploit the so called Haralick features [149], which have been successfully used, e.g., in image texture classification, and for which fast calculation algorithms have been proposed, e.g., by Miyamoto in [150]. The Haralick features are usually computed from a gray-level co-occurrence matrix, in which each element value is considered to be the probability that a pixel with value i will be found adjacent to a pixel of value j. This matrix might share a slight resemblance with the AHM. Therefore the Haralick features were experimented alongside with the six features proposed in this chapter.

Equations 4-9 define the proposed six features computed from AHM matrices in this chapter. Notice that each feature is normalized to range [0,1]. These features are based on detecting certain aspects that seem to be useful in describing behavioral events based on the matrix presentation. The following two notations apply for the definitions of the proposed features:

$$A_{sum} = \sum_{i=1}^{N} \sum_{j=1}^{N} a_{ij} \tag{8.2}$$

$$A_{sum}^{-} = \sum_{i=1}^{N} \sum_{j=1}^{N} a_{ij}, i \neq j \tag{8.3}$$

If a certain set of cells has the most of the occurrences in it, that responds to regions/actions that person most often visits/performs. These are characterized by features 1 and 2. Feature 1 represents how much a person stays with the same location by the number of self-transitions normalized by the number of all transitions:

$$f_{DiagonalHeaviness} = \frac{\sum_{i=1}^{N} a_{ii}}{A_{sum}} \tag{8.4}$$

Feature 2 represents the normalized number of all the transitions to the same location:

$$f_{ColumnHeaviness} = \frac{\arg\max_j (\sum_{i=1}^{N} a_{ij})}{A_{sum}^{-}} \tag{8.5}$$

The more symmetrical the transitions have been, back and forth between certain cells, the more likely a person is looping between associated regions/actions. This symmetric property is described by feature 3:

$$f_{Symmetry} = \frac{\sum_{i=1}^{N} \sum_{j=1}^{N} (a_{ij} \times a_{ji}), i < j}{A_{sum}^{-}} \tag{8.6}$$

The less straightforward a person has moved, the more random transitions exist. This will introduce a wider spread in the element entries. This transition directionality is presented by feature 4:

$$f_{Spread} = \frac{\sum_{i=1}^{N} a_{ij}, j = [i-1, i, i+1]}{A_{sum}} \tag{8.7}$$

The more entries exist at the end of the generated cell-path, the more likely the person got stuck in one of his latest locations. This location mass point is defined as feature 5:

$$f_{MassPoint} = \frac{\arg\min_l \left(\arg\min_k \left(\sum_{j=1}^{l} \sum_{i=1}^{k} a_{ij} \geq 0.5 \times A_{sum}\right)\right)}{N \times N} \tag{8.8}$$

Single-way repetition accumulates on the top-side of the matrix diagonal, both ways repetition accumulates on both sides of the diagonal. The repetition direction is defined as feature 6:

$$f_{TopHeaviness} = \frac{\sum_{i=1}^{N} \sum_{j=1}^{N} a_{ij}, i < j}{A^{-}_{sum}} \tag{8.9}$$

This proposed 6-dimensional feature data can be reduced to 1-dimensional symbolic data, e.g., by K-nearest-neighbor clustering and representing a 6-feature vector with a single symbol corresponding to the cluster ID it belongs to in this 6-dimensional feature-space. This way the originally continuous 6-dimensional feature-data can be presented as a discrete symbol, required by some temporal models, such as the discrete Hidden Markov Model.

8.5.3 Classifier Models

The aim is to detect situations in which people are uncertain how to proceed. These situations are defined as moments when people start repeating their actions. In this study case, the actions are the movements of the people. Detecting repetition is a very challenging problem due to the variability of paths and the pace of action. Three different classifier models are experimented with in detecting the repetitive activities described before: neural networks, decisions trees, and naive Bayesian classifiers.

In one set of experiments the classifier was a feed forward Neural Network (NN). NN is a network of interconnected nodes, which takes an input (e.g. six features) and defines an output (one of four classes) as a result of the data passed through the hidden nodes of the NN model [151]. After preliminary experiments 10 hidden nodes in a single layer are used in the experiments. How the hidden nodes relate to each other, is defined by training. Once trained, a new input (feature) is given to the model, and the outputs are the probabilities of each of the available classes. The class with highest probability is considered as the final decision.

Decisions trees (DT) were exploited as a second classifier model. All features are considered for each decision split, and in training the weight of every observation is considered as 1. The resulting classifier is a pruned binary tree, in which each branching node is split based on a value of one of the features. Once reaching a leaf-node the class for a feature-vector is given corresponding to the class symbol of that leaf.

As a third option the naive Bayesian classifier was tested. The naive Bayes classifier assigns a new observation to the most probable class, assuming the features are conditionally independent given the class value. A normal (Gaussian) probability distribution was assumed for all the different features in the training process.

8.6 Experiments - Setup

The proposed system was studied with two different datasets. The first dataset was based on multi-view recordings of actual people acting out as customers in a small L-shaped shopping environment, see Figure 8.7. For more controlled settings in terms of movements and variability a Matlab-toolbox was built for computational simulation of motion trajectories in a square-shaped space recorded by one virtual view, see Figure 8.8. For both datasets, the locations were quantized at various levels of detail; the dynamic cells were created with fixed radius of 30-100 pixels on simulated data, and 10-100 pixels on real data in 10 pixel increments. 30 pixels roughly match the size of half a step, and 120 pixels is roughly the same as the average person height.

8.6.1 Real Data

Real multi-view visual data was recorded in a environment that resembles a small clothing shop. See Figure 8.7 for the room layout and camera viewpoints for an example scenario containing three individuals. Ten subjects were instructed to perform four different scenarios. Each person entered the shop separately, thus providing clean one-person video from three views. The subjects were instructed to perform a given activity in the shop but limit the stay in the shop to approximately one minute. The item that the person was interested in, was described to them by the instructor of the recording session, before the person entered the shop. Once a person enters a shop the accumulation of transitions starts. When they exit, the accumulation stops.

The first routine was described as a successful purchase, in which a person enters the shop, looks around for an item, and after finding it approaches the cashier, and exits the shop after the purchase. In the second routine the person finds the area of the item-of-interest, but fails to pinpoint the item itself. After having looked around in the area, person exits the shop. In the third routine, a person repeatedly returns to a certain area, after which individual exits the shop. In the fourth routine, two to three areas are repeatedly returned to, before exiting the shop.

8.6.2 Simulated Data

In order to test the AHM method in controlled settings an algorithm was designed to simulate the movements of a person in a room. The motion path of a person is simulated on a floor plane of 4×4 meters. The positions that the person walks to are defined by 12 randomly selected points (x,y); see Figure 8.8 for an example on point indices and locations. The floor is divided into four same size squares. Each square contains 3 of the 12 randomly selected points for each simulation run.

The class-specific behavior is achieved by creating a 12-point binary transition matrix, in which the elements marked with 1 (active) are possible transitions to take.

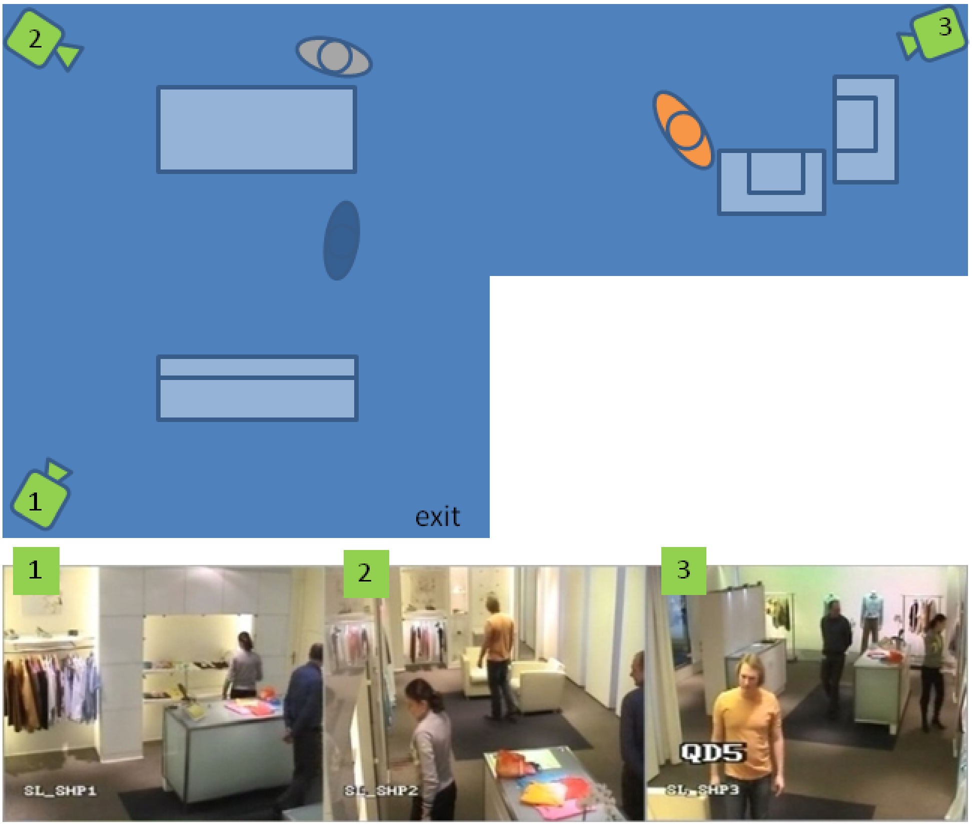


Figure 8.7: *Recorded data: A small shop environment for acting out scripted routines, covered by three cameras with partly overlapping views.*

The randomly generated and modified matrix remains the same during the simulation of a sequence. A fully active matrix is used for creating neutral behavior. All the repetitive classes are based on a randomly filled random-matrix. For stay behavior the diagonal elements of the random-matrix are marked active and for one position all the transitions are marked non-active. One of the columns of the random-matrix is marked active for the return behavior. For the loop behavior a pair of symmetric elements ([i,j],[j,i]) is marked active in the random-matrix and transitions from these positions are marked non-active.

The path of a person is created by sampling randomly the active elements of the class-specific transition matrix of 12 points, and by making a person walk these randomly selected transitions according to the defined step-size. The step-size is randomized before each simulation between 0.5 and 1.0 meters. The simulations runs are terminated after a certain amount of seconds has passed. Six different durations of [180,360,720,1440,2880,5760] seconds are used to gather 10 simulations per each behavior class.

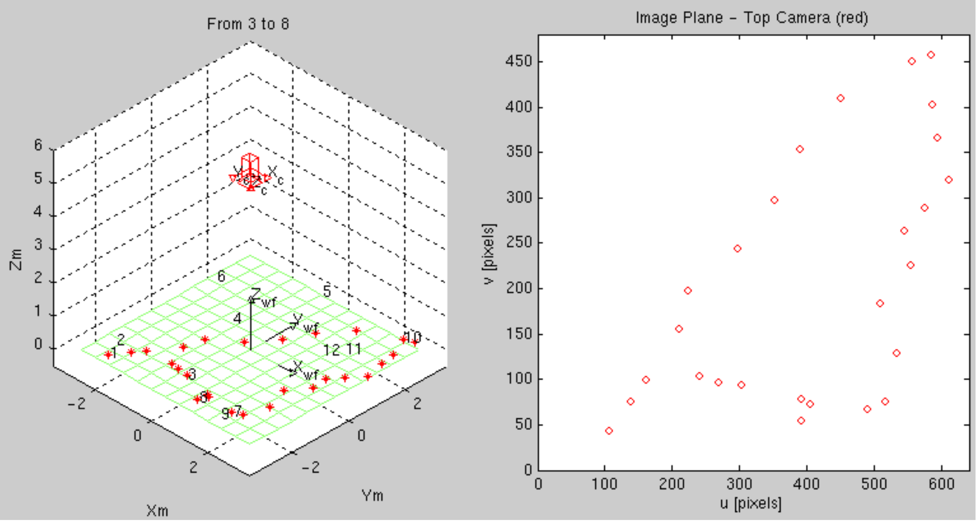


Figure 8.8: *Simulated data: Matlab environment for creating synthetic path data, left) by simulating steps on floor plane, right) capturing them on image with the virtual camera installed above the floor.*

8.7 Experiments - Results

A set of different experiments is performed on each dataset, the recorded and the simulated trajectories. Given a motion trajectory, AHM forms the transition matrix, from which features are computed and the trajectory is classified as one of the four classes: neutral (1), stay (2), return (3), or loop (4). These inferred class labels per sequence are compared against the manual annotation of sequences, to form an accuracy value ranging from 0 to 100%.

8.7.1 On Simulated Data - Classifier and Features

The simulated trajectories are used to study the applicability of AHM to repetition detection with three different classifiers models based on the proposed 6 features. The model that performs the best is further used to study which set of features provides the best accuracy.

In Figures 8.9, 8.10, and 8.11 the results for NN, DT and Bayes classifiers on simulated data over 6 different durations of simulation in seconds are shown averaged over all 40 sequences, presented for each of the 8 cell-resolution levels from 30 pixels to 100 pixels. The simulations runs were terminated (*term* in figure-legends) after one of six durations defined in seconds (value following *term* in figure-legends).

The shortest simulations of 180 seconds offers much lower accuracy (65%-57%) than the longer simulations. With simulation running for 360 seconds the patterns of the classes are already much more developed, thus a significant increase (to above 80%) in the repetition detection can be seen. Considering the longest simulations of 5760 seconds, NN performs the weakest (88%-77%), DT provides better results (90%-82%), and Naive Bayesian outperforms the others (95%-91%) by averaging 93.1%

over all cell radius levels. Based on these results the Bayesian model was fixed as the classifier.

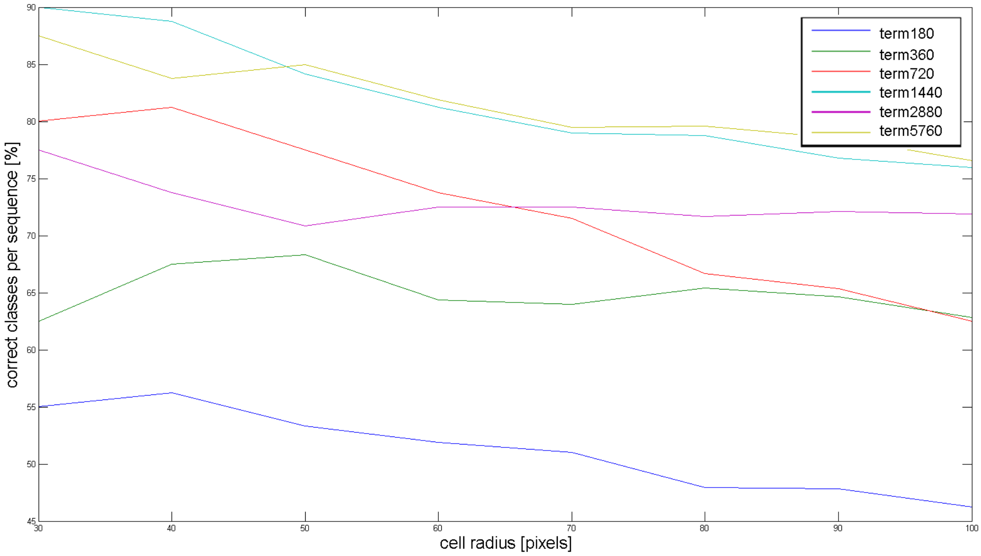


Figure 8.9: *Simulated data with proposed features: Classification accuracy of NN-model as function of the cell radius, presented here for six different lengths of simulation runs defined in seconds.*

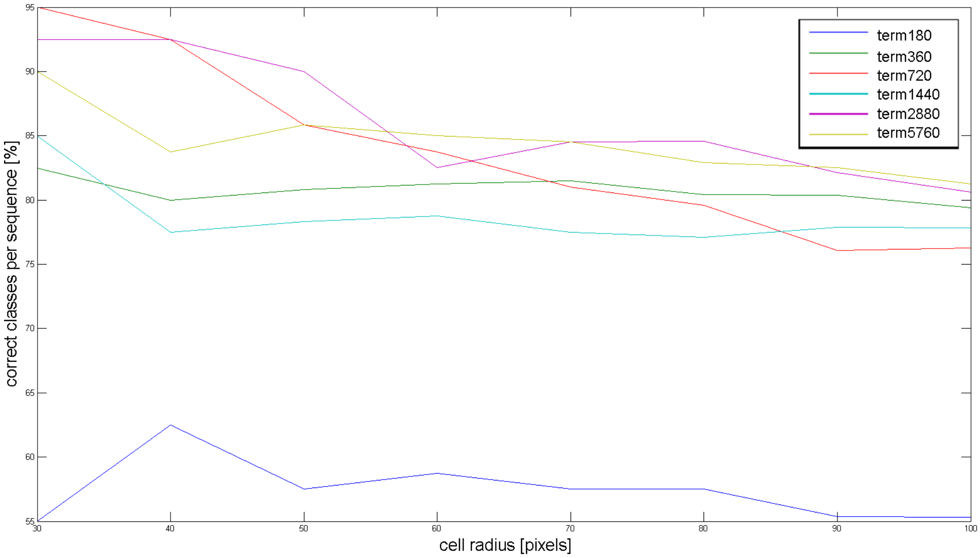


Figure 8.10: *Simulated data with proposed features: Classification accuracy of DT-model as function of the cell radius, presented here for six different lengths of simulation runs defined in seconds.*

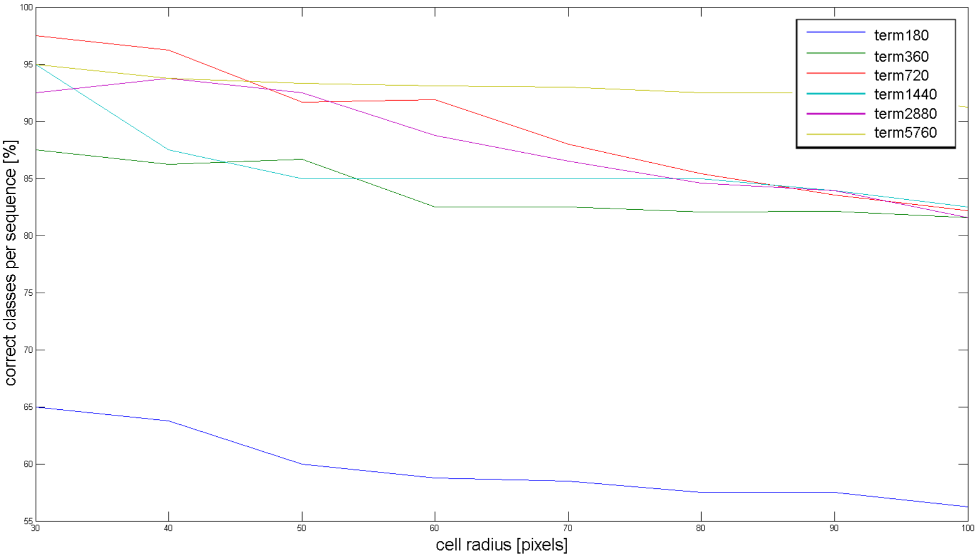


Figure 8.11: *Simulated data with proposed features: Classification accuracy of Bayes-model as function of the cell radius, presented here for six different lengths of simulation runs defined in seconds.*

Respectively, in Figures 8.12 and 8.13 the classification accuracies with the Naive Bayes model using the 13 Haralick-features and combined features (own+Haralick) are presented. Considering the longest simulations of 5760 seconds with Bayes model, the Haralick-features (88%-81%) or the combined features (95%-88%) do not reach the accuracy of the proposed features. Based on these results, the feature-set of the six features defined above was used in the subsequent experiments.

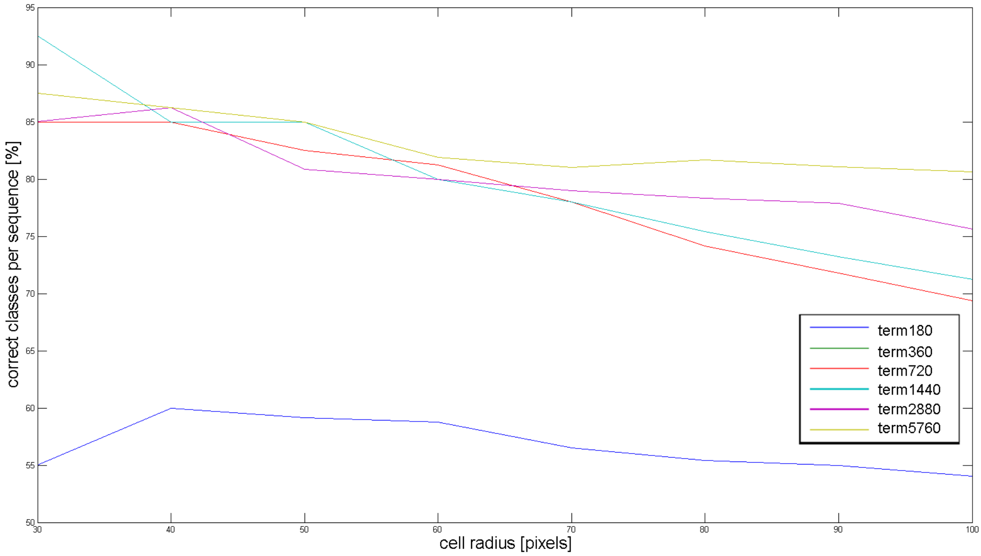


Figure 8.12: *Simulated data with Haralick-features: Classification accuracy of Bayes-model as function of the cell radius, presented here for six different lengths of simulation runs defined in seconds.*

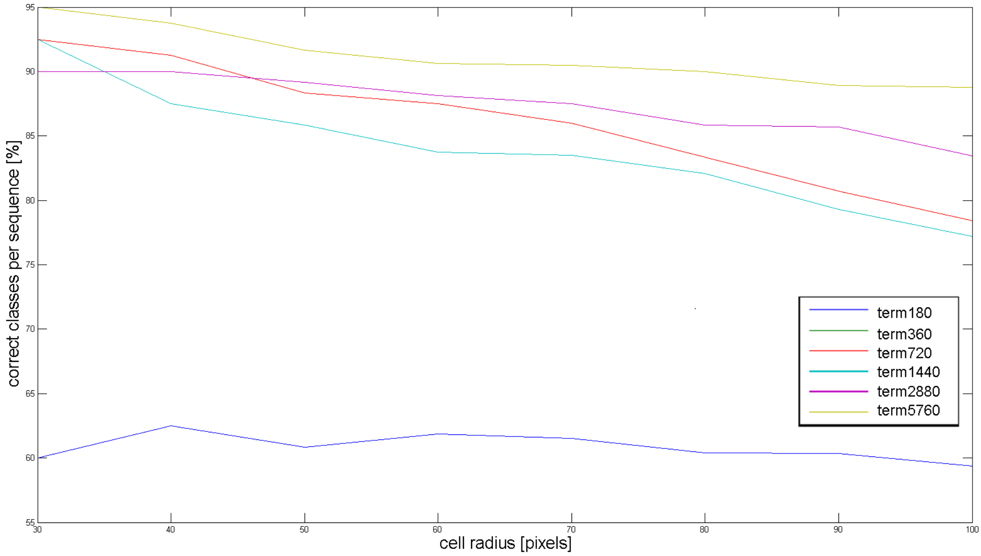


Figure 8.13: *Simulated data with combined features: Classification accuracy of Bayes-model as function of the cell radius, presented here for six different lengths of simulation runs defined in seconds.*

8.7.2 On Real Data - Cell Resolution

The naive Bayesian model based on the six proposed features was used to classify the real sequences. In Figure 8.14 the classification accuracy on real data is presented for the Bayes model for each of the three views, as a function of the cell radius from 10 to 100 pixels.

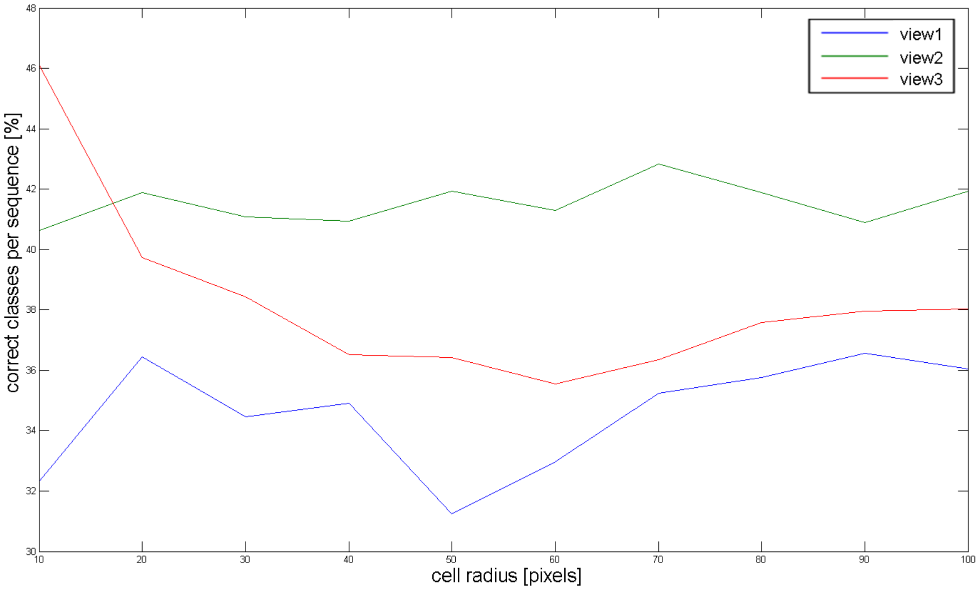


Figure 8.14: *Real data: Classification accuracy as function of the cell radius, presented here for each of the three views separately.*

Based on Figure 8.14 it can be stated that camera-1 performs best with cells of radius 20 pixels (37%), camera-2 with 70 pixels (43%), and camera-3 with 10 pixels (46%). In average camera-1 provides accuracy of 34.6%, camera-2 leads with 41.5%, and camera-3 gives out 38.3%.

If one would have to fix the cell resolution as same for all the views, the cell radius of 10 pixels would provide the best across-views accuracy (40%) of all the cell-resolutions by a very small margin. When selecting the best cell radius specifically per each view, the across-views accuracy is higher by 2% reaching average expectation of 42%. The best resolutions (20px,70px,10px) per camera are fixed for the following fusion experiments.

8.7.3 On Real Data - Fusion

In order to decide which fusion level provides a better accuracy, the majority fusion method is used to combine data from the three views at the two fusion levels of features (fusion-level index 1) and decisions (fusion-level index 2). See results per person, per behavior and per fusion level in Figure 8.15.

The fusion at decision-level provides an accuracy of 49%, which is 7% better than what was the expectation value for across-views fusion. In contrast, feature-level fusion performs close to the expected value giving out only 41%. Clearly, fusion at the decision-level is desired.

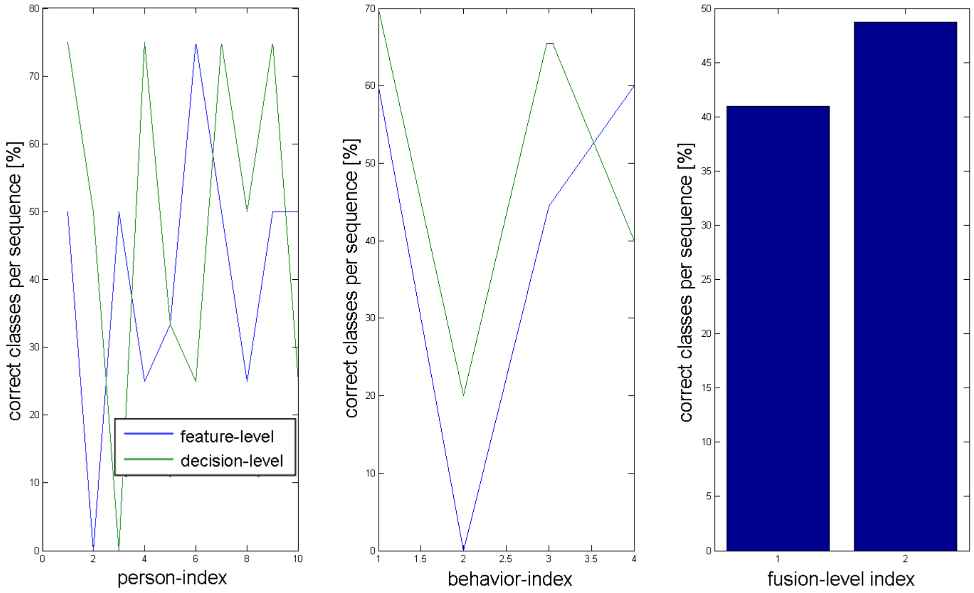


Figure 8.15: Real data: Classification accuracy with majority methods at both fusion levels, presented here left) per person, middle) per behavior, and right) per fusion level.

The results from the majority fusion method are compared to the two other fusion methods of view-selection. See Figures 8.16 and 8.17 for the fusion results of coverage-based and ideal view-selection methods.

The coverage-based view-selection increases the accuracy of both fusion levels. Feature-level fusion achieves 44% accuracy, and decision fusion still remains as the better fusion level giving out 51%. The oracle-based view-selection presents the theoretical upper limit of accuracy for selecting a view, with this real data at the decision-level providing accuracy of 75%.

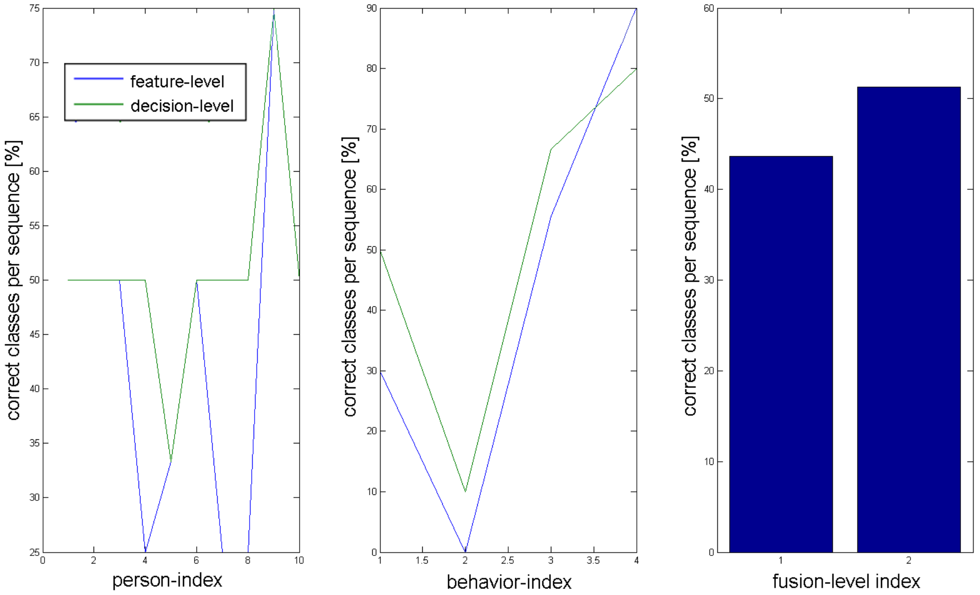


Figure 8.16: *Real data: Classification accuracy with coverage-based selection method at both fusion levels, presented here left) per person, middle) per behavior, and right) per fusion level.*

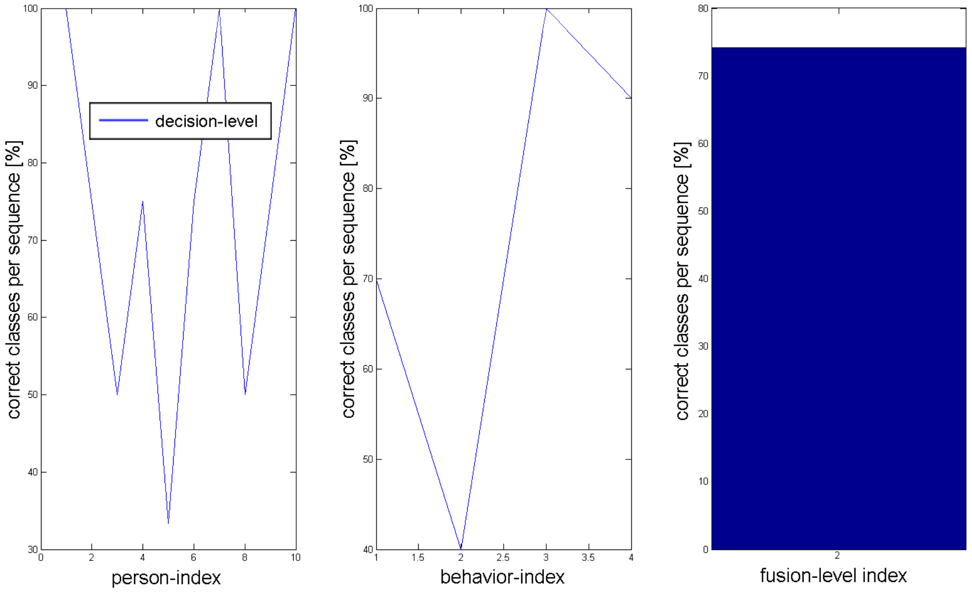


Figure 8.17: *Real data: Classification accuracy with ideal view-selection method at decision-level, presented here left) per person, middle) per behavior, and right) for decision-level.*

8.8 Discussion

A new method for modeling repetitive behavior by recording transitions between states was proposed. The method is based on features extracted from Action History Matrices (AHM). AHM consists of three types of information. First, the state depends on the nature of interesting repetitions. A state can be, e.g., an activity class (such as walking or interacting) or a location (such as transit area or counter). Second, the state observations will be captured in multiple resolutions such that spatial locations can be quantized into different sized cells and activity detection can be done over different length intervals. Third, the transitions will be accumulated into a transition matrix, optionally controlled with additional constraints on the transitions, such as requiring the person to stop in a location for the transition to be recorded.

On Proposed Methodology
In the studied application, the IDs of quantized locations (cells) were used to infer if a person was in a repetitive status. The cells were created at various resolution levels, providing coarse to fine details on transitions. A cell was dynamically created centered on the new location as a person moved through the environment; no cells existed before a person had entered the scene. All the transitions between cells were accumulated into a transition matrix; starting from the time a person enters the scene till the point where they exit the environment.

Once provided with an AHM-based transition matrix, the features derived from it were used to separate different classes of behavior. The six features were defined in such a manner that they describe the basic properties of a matrix; for example, what percentage of the transitions lies on the diagonal elements. These features were then exploited in training and running a behavior classifier based on NN, DT or naive Bayesian model.

Customer behavior in a shop was taken as an example application. Focus was in behavior that was repetitive because the customer encounters difficulties in finding a suitable item. Two datasets were created for experiments: a recorded and a synthetic dataset. Focus was in finding behavior that was repetitive, and in a case of positive decision further classifying which of the three basic repetitions was present: 1) the person staying within a location, 2) repeatedly returning to a location or 3) looping between few locations.

The dynamic multi-resolution cell based creation of a transition matrix by the AHM has potential and can be applied to behavior detection, in this case to classification of repetitive motion patterns. However, the proposed system does not yet achieve high accuracy values in classification with real data. The lack of accuracy in vision processing, the limitations of a single camera view, and the extreme variability between people create problems that need addressing.

On Fusion
Two different datasets, video recordings in a shop environment and motion path simulations, were used in the experiments. The AHM-based system achieves an accuracy of 97% with most suitable scale and naive Bayesian classifier on the simulated movement data. In addition, the performance of two fusion levels and three fusion methods were compared in the context of the AHM method with the multi-view recordings. The results show that decision-level fusion offers consistently better accuracy than

feature-level, and that the coverage-based view-selection fusion method outperforms marginally (51%) the majority method; An optimal upper limit for view-selection was found to be 75% with the recorded data.

Fusion Approach The aim of inference was to detect moments when the accumulated evidence implies that a person has started to repeat his or her movements. No fusion approach was fixed, as either consensus or sensitivity to alarming repetition could be considered as viable options.

The preferred fusion approach in practice might be sensitive for triggering alarms when considering a customer to be repetitive. Especially in the case of personnel responding to the alarm, the system could further rely on the judgment of the personnel on the situation. In this case, a sensitive fusion approach could be considered better [Design Rule 1]. If an automated guidance system would be used to support the customer, the application should be much more cautious in triggering alarms.

Vision Network The multi-camera data from the shoplab environment was gathered by a professional recording system. All the camera models and parameters were the same and stayed fixed during the experiments [Design Rule 2]. For example, if any of the settings in any of the cameras were adaptive, the camera could easily fail in image segmentation and the track of the person would be most likely lost. Depending on the fusion approach this complete lack of supporting data from one of the cameras, could badly affect the accuracy of detecting repetitions.

A dedicated ethernet-network was used in the data transmissions, with no external traffic or other impacts allowed within the recording network [Design Rule 3]. For example, if the stability of the local network would have been compromised, the data would not be timely delivered to the central fusion unit. Because all the tracking data is accumulated over time and the decision on repetition is made based on this accumulated set of data, the application is not as time sensitive as the ones discussed in the previous chapters. However, for correctly recording the transitions between locations and not jumping over a location-cell, the losses in consecutive frames have to be maintained small enough.

The experiments did not include the user supporting system that would possibly have to interact with the customer by guiding him with visuals. Therefore, no fusion mechanisms for handling rendering induced artifacts were needed. In a full implementation of a responding system, the effects of the created visuals to the vision processing should be counted for. Manual annotation of some interest regions within an image, or implementation of joint processing algorithms robust to color changes, could help in dealing with visual artifacts [Design Rule 4].

Fusion Architecture Within a confined specific environment such as a shop, there hardly are any resource limitations in capturing, transmitting and collecting multi-camera data. Some parts of a shop, such as entry/exit area, might have a specific phenomenon that can only be of interest in that specific area. In that case it would make sense to group together the cameras that observe such an area. However, in the proposed application the behavior of the user needs to be uniformly captured across the shop. Therefore, a fully centralized fusion architecture serves the application the best [Design Rule 6].

Fusion Level It can be noted that the raw matrix data is hardly combinable as the dimensions and elements differ between the cameras. Only by a calibrated vision network could the tracking be jointly done in 3D, which would make it possible to accumulate a joint transition matrix. With the recorded data, the fusion performed at decision-level always provided a better accuracy than fusion at feature-level [Design Rule 8]. This might be due to the great variability in feature-values between the views, with which the feature-level fusion is unable to cope. The variability is most likely due to three factors: view limitations, greatly different coverage-times on person per view, and the short duration of the real recordings.

Fusion Method The coverage-based view-selection fusion method outperforms the majority method with a small margin. Therefore, it appears that when individual sensors are not capable of good accuracy, a simple common consensus approach will not be able to increase the accuracy of classification. A more accurate method then should deploy a best-view criteria, based on which a single camera is used at beneficial times to infer the class [Design Rule 10]. Naturally, the type and suitability of the criteria become crucial for beneficial fusion [Design Rule 11]. An ideal upper limit for view-selection was found to be 75% with the recorded data.

In summary If the system can analyze on-line the accuracy of the person tracker for a specific view, this will help fusion in selecting the appropriate cell-resolution that gives increased accuracy in decision making. Some views provided better accuracy in average; therefore, a simple sensor selection might sometimes increase fusion accuracy. In contrast, the majority based fusion methods seem to struggle in achieving accuracy improvements when encountering cameras with generally low quality information. With the recorded multi-view data used in these experiments it is the decision-level fusion that consistently provides marginally better accuracy than fusion at the feature-level. So it appears that the high-variability conditions of real-data do favor fusion at a higher level of data abstraction, in this case with decision data.

Future Work

There are several ways to improve the performance of the presented system. As actions of interest the use of movement history alone was studied, but different actions such as activity (walk, stand, interact etc.) can equally be used in defining the AHM. Resulting versatility of action history data should provide a self-strengthening basis for behavior inference.

Six features were proposed to describe the repetition characteristics of the AHM-matrix. It is probable that a larger and more application-specific set of features can further improve the inference accuracy. The accuracy of two basic data fusion mechanisms: selection and majority were discussed in this chapter. More powerful fusion mechanisms that exploit better criteria and scene context should further improve behavior inference.

Chapter 9

Office Ergonomics

Office environments provide an interesting opportunity for self-assessment. This chapter proposes a collaborative vision-system that leverages a personal webcam and cameras of the workplace to provide feedback relating to an office-worker's adherence to ergonomic guidelines, which can lead to increased well-being for the individual and better productivity in their work [1].

Section 9.1 starts this chapter by giving an introduction to the application of ergonomics and related research. The proposed system for both personal and ambient cameras, in addition to highlighting the fusion aspects in such a system, are defined in section 9.2. The multi-video experiment data is presented and important aspects related to the mobility feature are illustrated in section 9.3.

The experiments are divided into two parts: personal ergonomics and general mobility. Personal ergonomics relies solely on video analysis performed on the personal webcam. In contrast, the general mobility experiment exploits both personal and ambient cameras for deducing the worker's mobility state, thus giving rise to many fusion possibilities. These possibilities are studied for two fusion architectures: centralized and hierarchical, with various options for fusion level and fusion method. Results are shown for both experiments in section 9.4. The most important findings in relation to the vision fusion framework conclude the chapter in section 9.5.

[1] This chapter is (partly) based on:
Chen, Chih-Wei, Määttä, T., Wong, K., Aghajan, H. (2012). *A Framework for Providing Ergonomic Feedback Using Smart Cameras.* 6th ACM/IEEE Int. Conf. on Distributed Smart Cameras.

9.1 Related Work

The importance of proper ergonomics for the health and wellbeing of office workers is increasingly promoted by federal agencies such as OSHA (Occupational Safety and Health Administration) [152] and NIOSH (National Institute for Occupational Safety and Health). However, it is up to the individual workers to adhere to the proper ergonomics. Working long hours in front of a computer has become unavoidable for many people working in offices. However, the extended use of computers poses health risks including eye strain, and neck and shoulder pain. In response, ergonomic experts have developed guidelines that are designed to mitigate the risk of such workplace related injuries. In order to follow these guidelines, workers have to gain a measure of self-awareness of their bad habits, and be encouraged to correct them. The field of personal informatics has focused on providing tools to assist in this.

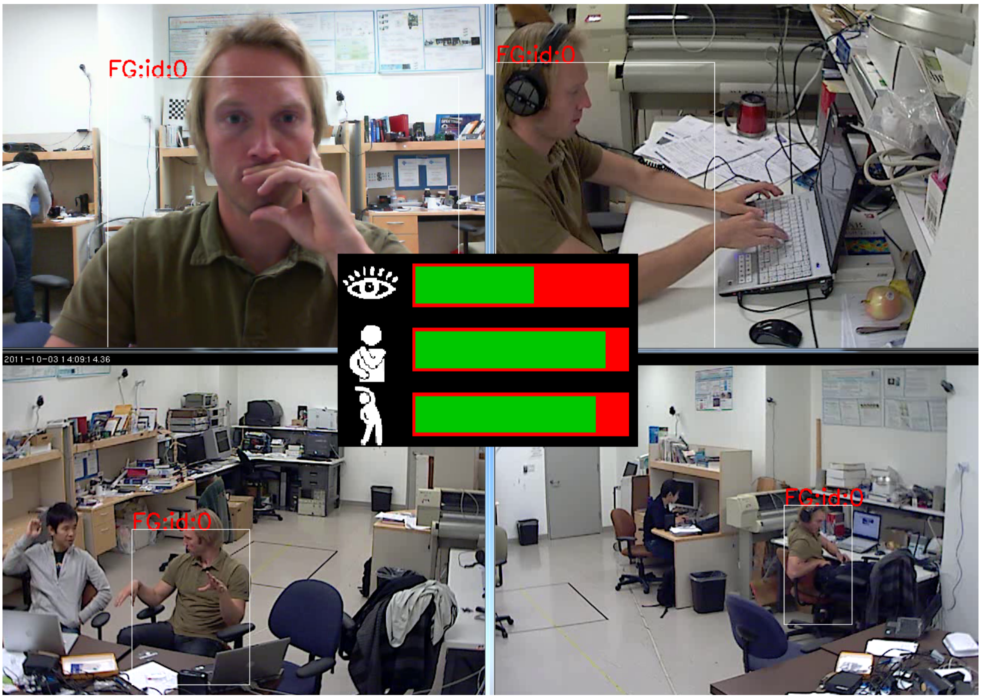


Figure 9.1: *Personalized Ergonomics: A multi-camera system monitors worker's daily activities around the office. Based on context-aware observations measures on attention, posture and mobility are computed. The personal data is accumulated over time based on which guidelines are personalized and feedback is given to the user.*

In this study a set of cameras are used to collect personal information relevant to workplace ergonomics. Given both close-by and ambient cameras, the estimate of the condition of the worker is both refined and given over a wider range of locations. By visualizing the processed information to the user we hope to increase the workers' awareness of their own condition related to the general guidelines. Figure 9.1 shows an example of a graphical feedback given by an application. Reminders for eye breaks, neck and shoulders exercises, and rest breaks could be sent to the user when the respective green bar is depleted. With contextual information the measurements can be correlated with different tasks, providing insights on changes in worker behavior. A small example on the effect of context is given in the experiments section.

There exist software programs that can remind the user at a fixed time interval to take a break. However, prompting reminders at bad timing might interrupt the user's work, causing the user to ignore the message. To make the reminders relevant, the system must take into account the user's behavior patterns and his personal schedule. For example, if the user usually takes a break every 45 minutes, then a reminder every 40 minutes might not be necessary. Similarly, if the user is scheduled to have a meeting in 5 minutes, he should not be bothered now. To this end a probabilistic model could be developed that learns the user's behavior pattern and adjusts accordingly the common guidelines. One such an approach to personalized ergonomic recommendations has been given in [153].

Gaze tracking has been used within many applications from analyzing the impact of advertisements for marketing studies, to developing innovative interfaces for HCI [154]. Most widely used methods are based on video-devices, because they are unobtrusive and cheap. Much work has been done to improve the performance, e.g., by using prior knowledge about the scene under a saliency framework [155], or by incorporating multiple cameras [156].

For eye blinking detection, Chau and Betke [157] proposed an approach in which eye location is detected from a temporal difference image when the user blinks, and templates for open eyes are created on-line. Local template matching tracks the eye location, and blinks are detected by thresholding the correlation score. A blink detector based on SIFT tracking and implemented on a GPU was proposed by Lalonde et al. [158].

Motion detection is commonly used to segment moving objects in video sequences. To this purpose techniques such as Mixture of Gaussian (MoG) which models each pixel as a Gaussian Mixture Model (GMM) have been commonly used [147]. These techniques have been extended, e.g., by including a feedback from the higher level modules (e.g., from person detector/tracker or illumination change detector) [159], or by including the removal of shadow areas [160]. These techniques are capable of good accuracy in ideal conditions, but are considered to be prone to segmentation errors when encountering environments with lighting changes, noise, and low contrast. Morphological operations are often used to overcome some of these problems to an extent. Graph cut based segmentation algorithms can perform better in such situations by grouping pixels together based on the neighbourhood similarities, but are computationally intensive [161].

Gathering comprehensive personal information has been made possible recently with the advent of ubiquitous sensors and computing power. A survey about how personal information is collected through ubiquitous sensors and reflected upon can be found in [162]. For example, the generation of a daily activity summary for triggering bad posture alarms was proposed by Jaimes in [163].

Detection of body posture and interactions with other people, are essential for improving wellbeing. In a 20-year study by Shirom et al. [164] a strong link was observed between higher level of peer social support and lowered risk of mortality. Chen and Aghajan [156] described methods for estimating the locations and head orientations of multiple users. Based on these two attributes, an interaction detector was trained to identify social events. The influence on behavior of these social events was studied by Chen et al. in [165]. The pose of the user together with simple activity monitoring was used by Chen at al. in [166] for automatically discovering regions of space and placements of furniture.

Ergonomics guidelines usually only provide high-level recommendations that are general for specific industry or task, but do not take into account personal preferences and habits. Therefore, warnings that strictly adhere to the guidelines might become annoying to the users, and could even jeopardize work efficiency and productivity. To address this problem, a multi-camera supported system that learns personal habits and preferences is proposed.

9.2 Vision System for Office Ergonomics

In the proposed system, there are two categories of smart cameras: the *personal webcam* and the *ambient cameras*. Additionally, an ambient camera that observes only the area of a persons desk is referred to as a *dedicated camera*.

9.2.1 System Architecture

The frontal personal camera above the user's computer screen extracts ergonomics related attributes. The ambient cameras monitor the entire office and record how multiple users utilize the office space. Data extracted by these smart cameras is sent to a central processing unit. The overview of the discussed system is illustrated in Figure 9.2.

The attributes are combined by a data fusion process and then used to learn the worker's profile. The model adapts to the visual observations, available user feedback and given contextual information. A user may provide feedback, e.g., by penalizing unwanted reminders or by providing the type of the task. Contextual information, e.g., on the common breaks and meetings can be gained by accessing worker's agenda. Personalized recommendations according to the modified user model can then be provided to the user.

9.2.2 Enabling Vision Techniques

The analysis of worker behavior relies on three different computer vision techniques: face tracking, silhouette segmentation, and motion history images.

Face Tracking The faceAPI from Seeing Machines [167] was used to track faces. FaceAPI is capable of tracking faces under a wide range of head rotations, which is critical for the proposed system since workers are not expected to look always directly into the monitor while performing various tasks at their desk. The faceAPI returns a 6 degrees of freedom estimate of the head's pose at moment t: the 3D head location and the yaw, pitch, and roll of head pose, which are expressed as a position vector x_t and a Euler angle vector θ_t respectively, $x_t = \left(X_t, Y_t, Z_t\right), \theta_t = \left(\alpha_t, \beta_t, \gamma_t\right)$.

Silhouette Segmentation With personal cameras, after performing a traditional GMM-based background subtraction and obtaining a foreground mask, the head

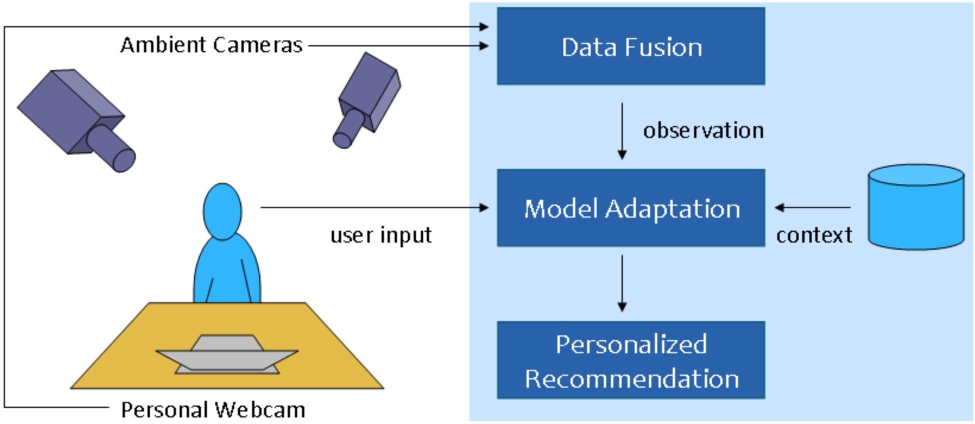


Figure 9.2: *System Diagram for providing personalized ergonomics.*

position given by faceAPI is used to refine the foreground mask. Morphological operations are applied to remove small holes in the foreground. The largest connected silhouette component that overlaps with detected face is defined as the worker's silhouette. With ambient cameras, similar refinement is performed based on HOG-template matching with SVM; see subsection 8.4.1 from the previous chapter for tracking details.

Motion History Images Motion history images (MHI) compute the difference between consecutive frames and mark the pixels with large difference values as the motion pixels $m_{i,j}$. In the experiments the motion frames were added together over a period of 8 seconds, 120 frames captured at 15fps. This formed an accumulated MHI-frame, that was used to compute MHI-based measures.

9.2.3 Measures for Ergonomics

The three techniques described before enable the computation of informative measures capable of determining worker's ergonomic situation.

User Attention For the proposed system it is satisfactory to extract the approximate gaze of the user by assuming the head orientation is aligned with gaze direction. This is a coarse approximation, but it allows to use the face tracking data to estimate what the user focuses on. Head direction is defined by the head position and orientation, which is considered then as the gaze direction. The estimated gaze vector is projected onto the plane spanned by the monitor for obtaining an attention heatmap.

Distance to Screen It is important to maintain a proper distance between a user and a computer screen to avoid eye strain [152]. Using the face tracking data, the distance between the user and the screen is extracted.

Head Motion Sitting in front of the computer can cause excess muscle tension in the neck, shoulder, and back. Stretching and short exercise can effectively relieve affected muscles and prevent strains from accumulating. Head motion can be derived from the face tracker data. In particular, the motion m_t of the user at time t is defined as the weighted sum of the tracked head displacement and rotation w.r.t. previous observation at $t-1$:

$$m_t = w_d \times |x_t - x_{t-1}| + w_r \times |\theta_t - \theta_{t-1}| \tag{9.1}$$

where w_d and w_r are the corresponding weights.

Work Periods and Breaks Taking regular breaks during sedentary work is another important activity recommended by ergonomic experts to promote health and well-being by reducing fatigue. The presence of a user in the view is used to determine if the user is working or on a break. The raw presence data provided by hits by faceAPI is processed by first finding gaps in the presence. The detected breaks that are shorter than a threshold are filtered out as insignificant absences. For the experiments a threshold of 10 seconds was used. From the starting and ending times of the work breaks, the system computes the durations and distributions of the work periods of the user.

Blinks The blink detector was built upon the face tracker, utilizing the tracked head and estimated eye locations. Given an observed video frame, the face tracker provides an estimate of the eye locations. Two local regions centered on these estimated locations are used to compute the accumulated pixel differences within a running time window. A decision on blinking is provided based on the blinking probability computed from the accumulated differences.

General Mobility For detecting the activity status of a person within the office, a rough measure of mobility is computed. This mobility feature is designed to be robust against changes in camera distances and angles. The feature is computed within a ROI, which is the rectangular region within an image that covers the visible part of the person. ROI updates it's position and size automatically based on the foreground pixels $f_{i,j}$ or pedestrian detection. The feature f_{mob} is defined as the ratio of motion pixels (MHI) to foreground silhouette pixels:

$$f_{mob} = \sum_{ROI} m_{i,j} / \sum_{ROI} f_{i,j} \tag{9.2}$$

Based on the mobility feature the system classifies a person's state as one of the three classes: *regular*, *mobile* and *in-transit*. The state is *regular* when person is engaged in focused working, thus only minor adjustments to hands, head and body are taking place. The state is considered *mobile* with larger rotations to body and displacements. *In-transit* includes cases when the person stands up or moves across the office. A simple thresholding was used to declare the state. The thresholds were defined by minimizing the overlap between the three classes. Threshold values of 0.075 and 0.7 were used.

9.2.4 Fusion of Camera Data

As multiple cameras provide their observations on the user's *general mobility*, great opportunities emerge for exploiting fusion in order to increase *certainty* and *visibility*. The different fusion levels and methods for office data are shown in Figure 9.3; see Figure 3.14 from chapter 3 for comparison.

Fusion Architecture The manner in which data is gathered is defined by the fusion architecture; see section 4.1 for more details. *Centralized* and *hierarchical* architectures are studied in this chapter. A centralized architecture combines directly all the data in a central unit, whereas a hierarchical architecture first performs fusion for subgroups of cameras and sends the results to a central unit.

Fusion Level The type of data to be combined is defined by the fusion level; see section 3.2 for more details. There are three types of combinable data within the proposed uncalibrated vision network. *Features* can be combined into a single feature-value. Both *class probabilities* and *class labels* can similarily be combined into a single estimate. *Hybrid* fusion is considered as a fourth level, in which all the resulting decisions from each of the previously mentioned levels are combined into a hybrid decision.

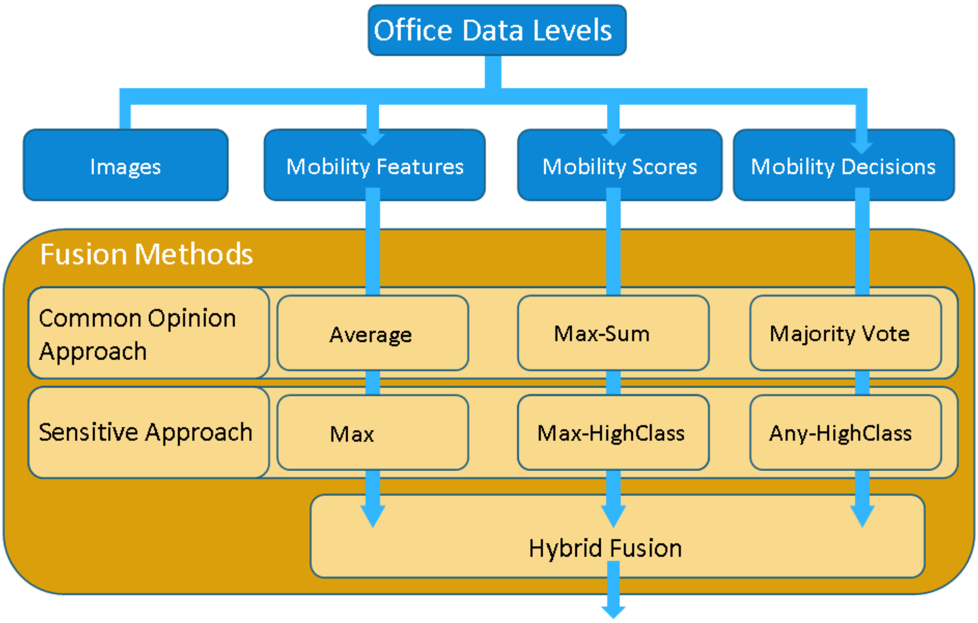


Figure 9.3: *The three fusion levels of feature, score, and decision for stating general mobility status. Fusion is performed with either common opinion or sensitive approach based fusion methods.*

Fusion Method The algorithm used to combine the data is defined by the fusion method; see section 3.3 for more details. Two approaches to fusion methods are studied: the first approach relies on the *common opinion* between the cameras, the second is *sensitive* to detection of the more mobile classes.

In the common opinion approach, the mobility features are combined by taking the average. Class probabilities/scores are combined by choosing the class with the most of probability mass. Class labels/decisions are merged by selecting the class with largest amount of votes/labels. In the sensitive approach, the features are combined by taking the maximum. The class of highest mobility, that had the highest score in any of the score candidates of that time, is the combined decision. Similarly, the class of highest mobility that has one or multiple votes is the combined decision.

9.3 Dataset used in the Experiments

Two researchers recorded their daily activities in a lab using their webcam and the IP-cameras according to the layout shown in Figure 9.4.

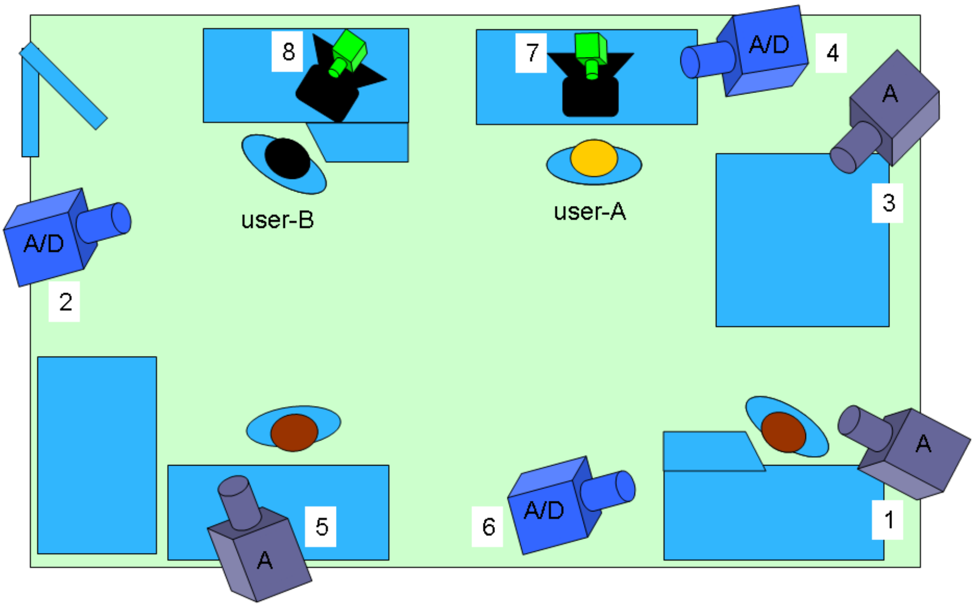


Figure 9.4: *Illustration of the recordings setup, with ambient (indices 1-6), dedicated (2,4,6) and personal cameras (7,8) labeled accordingly.*

Recordings started in mid-September 2011 and lasted for a month. Video was captured every weekday between 2pm and 4pm, using both the laptop webcams and six IP-cameras. Webcams were recorded on the respective laptops, whereas ambient videos were gathered over cabeled IP-network on another laptop. In overall, 40 hours of video from each camera was recorded at 640 × 480 resolution at 15 fps.

The annotation of this video data gathered on general mobility within the office is shown in Figure 9.5 for a single week. Correspondingly, the mobility feature values

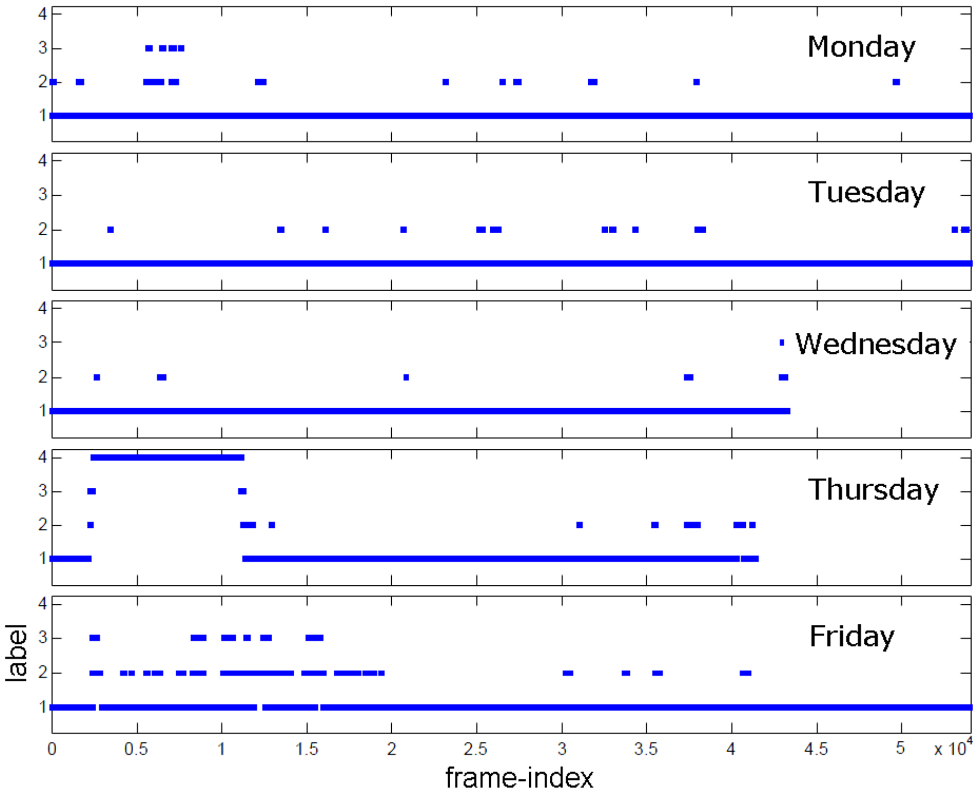


Figure 9.5: *Data labels starting from top for the sequences of Oct 3rd-7th for each frame. Label value* 1 *as* regular, 2 *as* mobile, 3 *as* in-transit, *and* 4 *as* away. *A frame of a sequence does not belong to multiple classes; overlaps are due to limitations in visualization.*

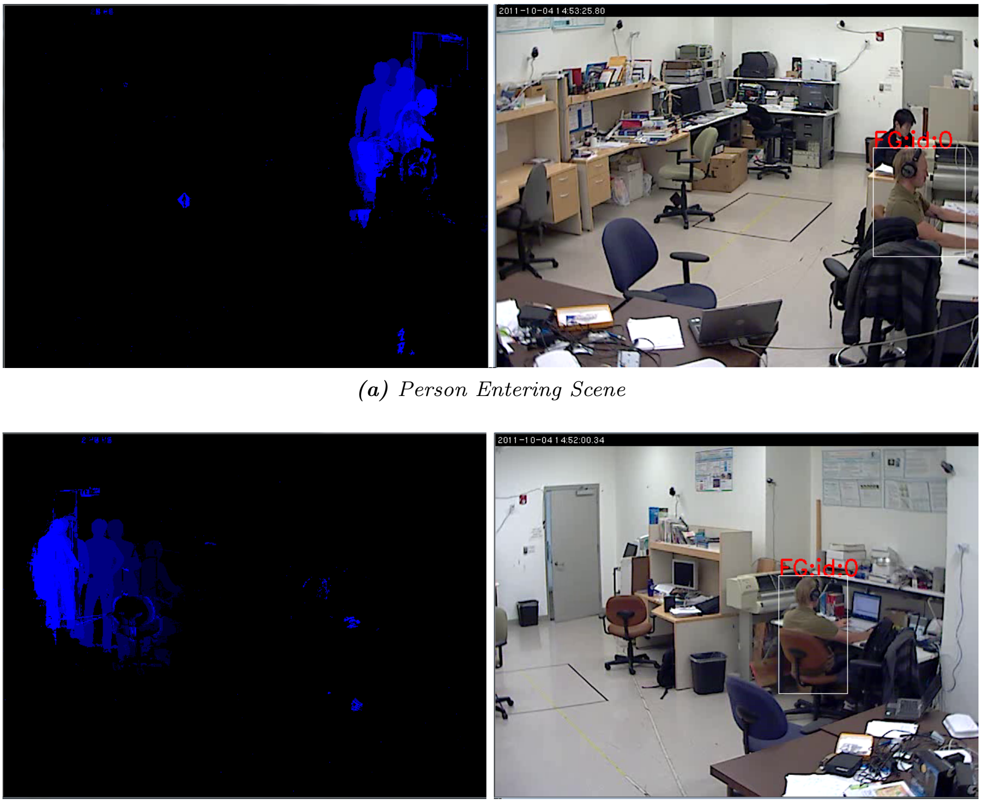
for one of these experiment days is presented per camera in Figure 9.7. The feature-values presented are the original values, but shifted in time to correspond to the reference timing given by the personal camera, camera-7.

9.3.1 Single-Camera Processing Challenges

MHI reacts to *motion* observed over time. Therefore there is a trail of motion, even after a person has sat down or has left the office. This trail will introduce remnant motion pixels for the duration of the motion buffer. Figure 9.6 exemplifies the cases in which a ghost trail is visible. The brighest blue signifies the most recent motion.

Image segmentation by background subtraction has two major challenges: initialization and update of the background model. All objects in the scene will be considered as part of the background, unless an initialization step ensures no users are considered as part of the background. A short sequence of 100-frames, with a clean background for the same day as the test sequence, was used to create a person-free background model in the experiments. Additionally the timing and the area-of update should be controlled. Otherwise, any changes to scene illumination or camera-gain deteriorate image segmentation. The lighting was fixed during all sequences, and in case of movements of the webcam a new initialization sequence from the time of new settings was provided. See Figure 9.8 for an illustration of these two problems.

Any passers-by will also be detected as part of the foreground as they too are new to the background model. By only considering 8-connected foreground pixels that partly overlap with the detected face/hog-template as belonging to the person's silhouette, some robustness to background movements is achieved. An example on removal of background noise by limiting the foreground to the 8-connected silhouette pixels is shown in two left-most images of Figure 9.9.



(a) Person Entering Scene

(b) Person Leaving Scene

Figure 9.6: *Examples of left) MHI-images used for detection of general motion, right) the regular video frame.*

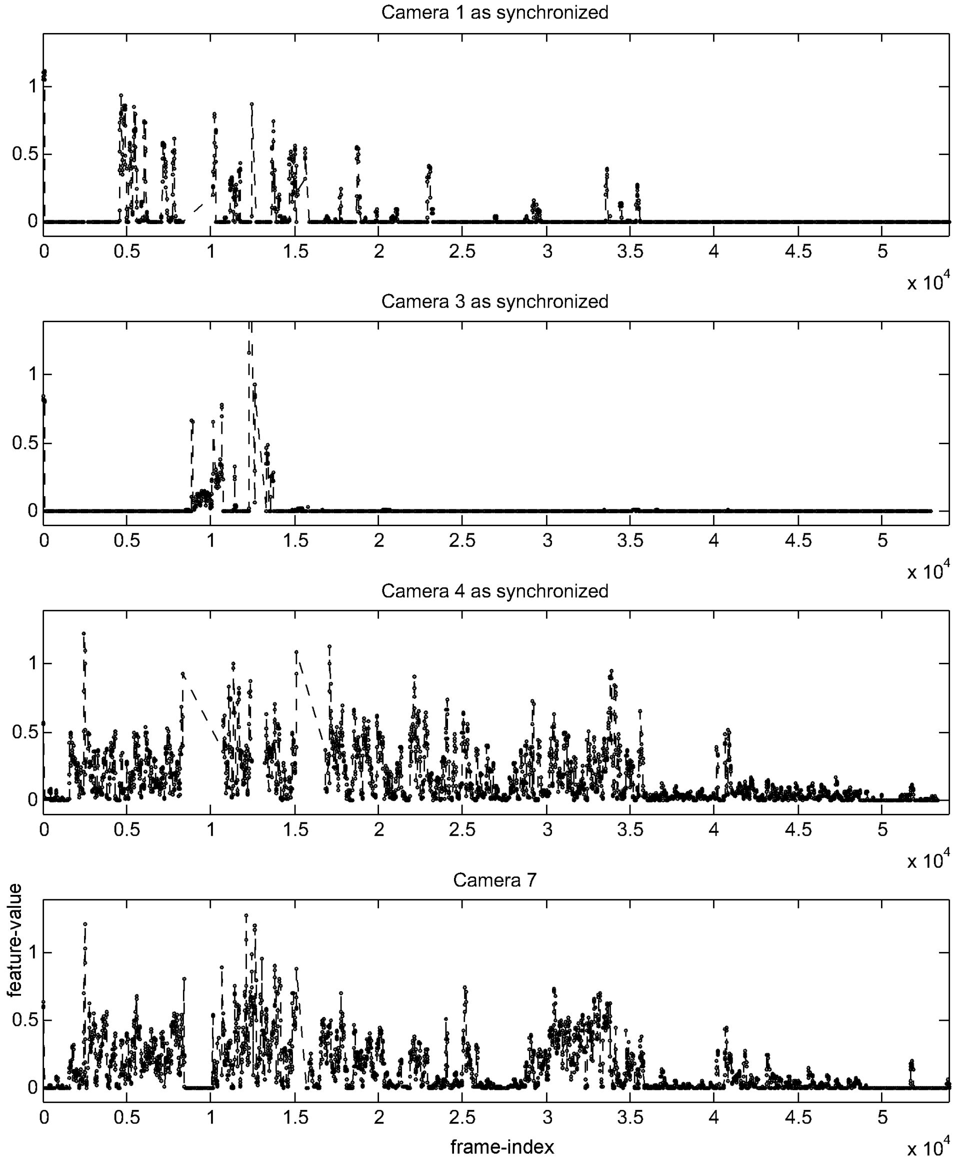
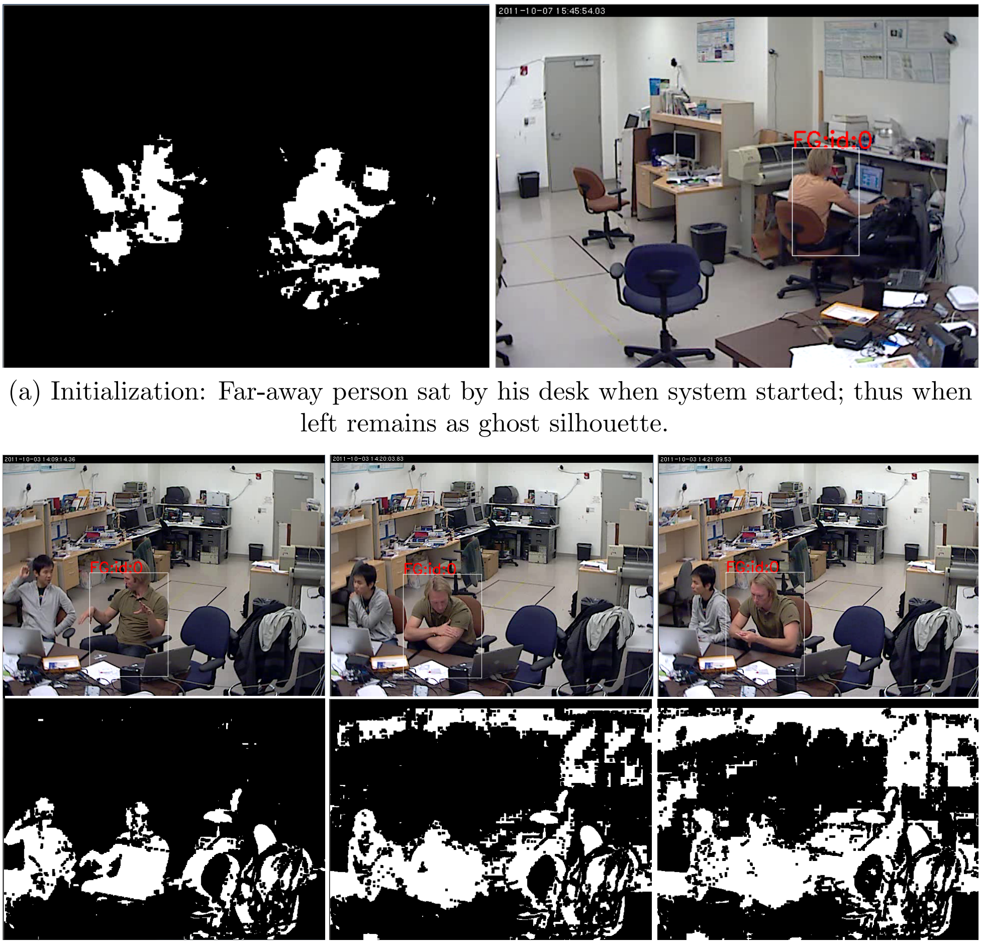


Figure 9.7: *The values of the mobility feature for each camera observing User-A on Oct 7th. For this and the other days the feature values are further normalized to [0, 1] interval by dividing with the maximum value.*



(a) Initialization: Far-away person sat by his desk when system started; thus when left remains as ghost silhouette.

(b) Update: As sequence develops from left to right more noisy FG-pixels appear due to gradual changes in gain/illumination.

Figure 9.8: *Two common image segmentation issues.*

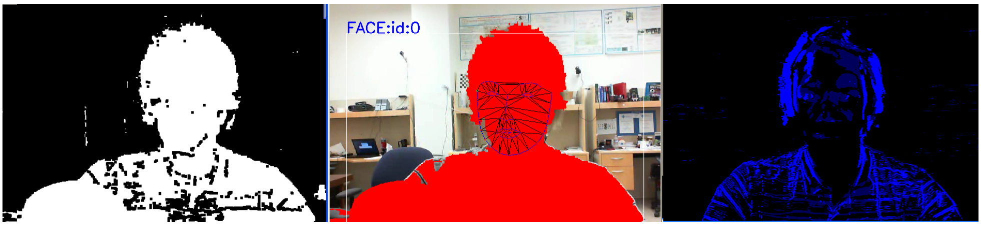


Figure 9.9: *Examples of left) image segmentation, middle) face-tracking aided foreground segmentation, and right) MHI for a personal camera.*

Three issues caused significant *synchronization* problems for the recorded video-streams. First, cameras dropped frames while capturing. Second, some data packets, and hence some frames got lost in transmission. Third, some frames were dropped during the recordings by the laptop as it fails to write all the frames from all six IP-video streams. Figure 9.10 illustrates how much three of the IP-cameras fell behind of the personal webcam (camera-7). Because Oct 5th had significant fluctuations, and thus complicated to compensate shift by linear approximation, it was omitted from general mobility experiments of that week.

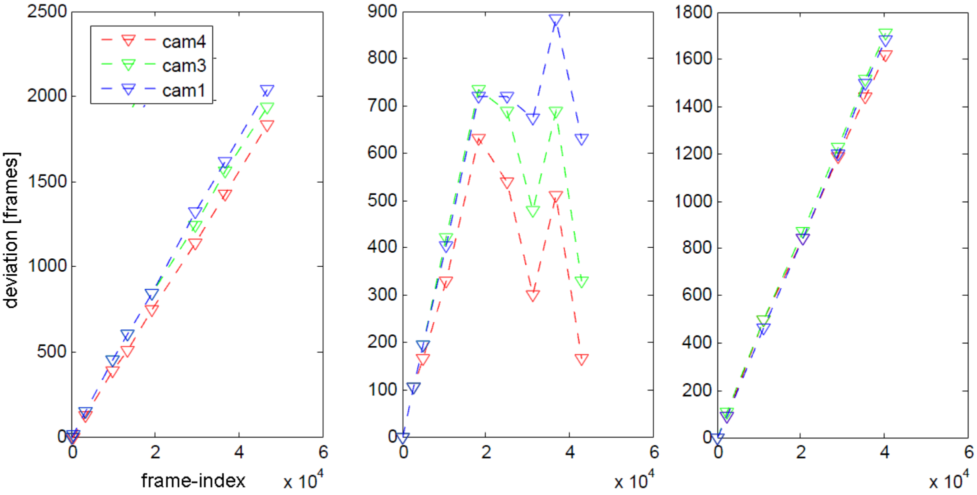
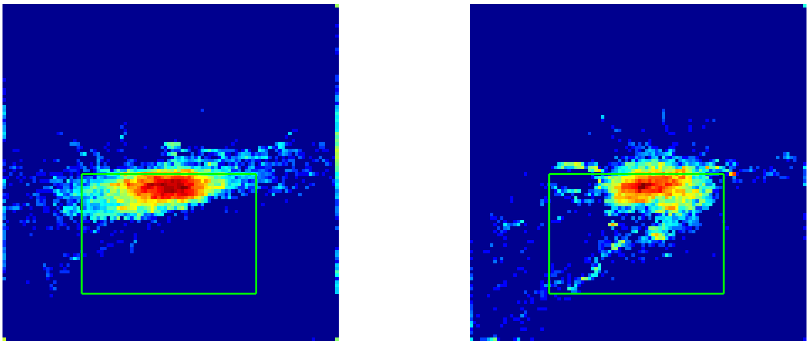


Figure 9.10: *Example deviations presented in number of frames of IP-cameras w.r.t. personal webcam (cam-7), starting from left for all the frames of the sequences of Oct 4th, 5th and 6th.*

9.4 Experiments

Analysis of ergonomics at the desk was performed by examining the working styles of two individuals during two of the recorded days based only on their personal webcams. In addition, general mobility of user-A within office was studied with his webcam and three IP-cameras over four days.

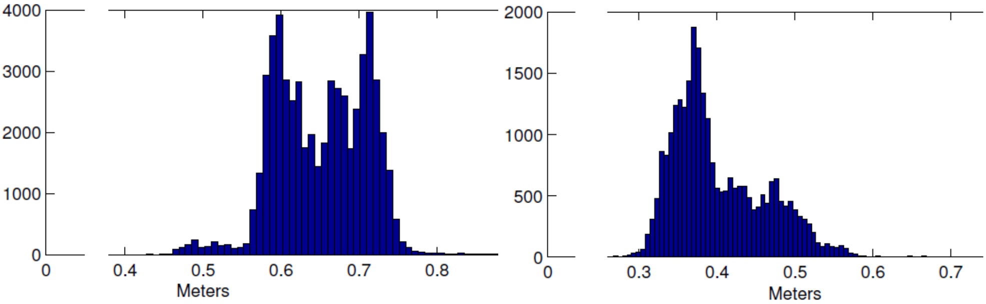
Attention HeatMaps Heat maps indicating areas of focus were obtained from the user gaze analysis, see Figure 9.11. The majority of the time was spent looking directly at the computer screen. The elongated pattern in the horizontal direction is due to the user panning their head side to side. Both users frequently look away from their computer screens, thus decreasing eye fatique. User B's heat map also indicates frequent patterns of looking down and toward the side, which correlates to the user reading a document placed flat on his desk. User B should place his documents closer to the same plane with the screen [152]. Additionally, both heatmaps indicate a proper monitor placement, as the centroid of the heat map is approximately at eye level.



(a) user A (b) user B

Figure 9.11: Intensity of gaze attention ranging from no attention (blue) to most of attention (red), with green boxes indicating approximate screen locations: ***(a)*** *mainly focused on objects at eye level,* ***(b)*** *focused on both screen and objects on his desk.*

Screen Distance The histograms of the screen viewing distance for the two test users for the times their face was successfully tracked are shown in Fig 9.12. Two distinct viewing patterns are observed. User A's viewing pattern is highly bi-modal and the viewing distances are greater compared to User B's pattern, which features a tail towards longer distances. These would imply that User A is probably farsighted and seems to work mostly in two distinct sitting postures: the ergonomically preferred and leaning in postures. User A was asked, and he did confirm that he most likely is farsighted, but not yet diagnosed.



(a) user A (b) user B

Figure 9.12: Histograms of screen distances: ***(a)*** *features a distinctive bi-modal distribution,* ***(b)*** *the distance distribution features a tail, possibly indicative of work not involving computer.*

Head Mobility, Blinks and Duration To infer the state of head mobility, the displacement weight w_d in Eqn. 9.1 was set to 1 and rotation weight w_r to 0.3, measuring x in centimeters and θ in degrees. The weight for displacement was set bigger, because 1) the tracking of head location is more robust than that of head rotation, and 2) the changes in head location are considered more important in order to relieve tension. A person was labeled to have high *mobility* at time t if motion $m_t > T_m$, with T_m as 10.

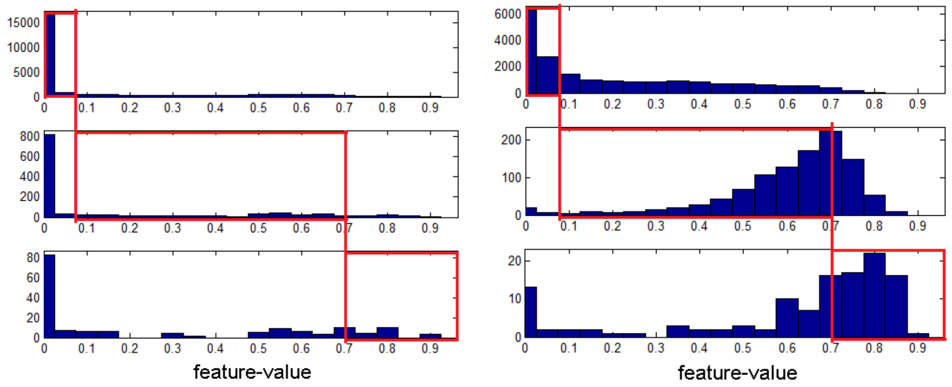
		Break Interval	Presence	Mobility	Blinks
A	Mon Avg	55 m. 40 sec.	82.1%	6.2%	8.4/m.
B	Mon Avg	33 m. 42 sec.	78.6 %	10.2%	14.3/m.
	Fri Avg	28 m. 34 sec.	61.4 %	12.9%	26.5/m.
	Reading	16 m. 15 sec.	-	14.4%	23.6/m.
	Coding	42 m. 27 sec.	-	4.8%	17.7/m.

Table 9.1: Example statistics of user A/B activity on different days.

Table 9.1 shows example statistics of two users highlighting differences between the user's habits from blink rate to average work period. Using contextual data for User B, it can be seen that computer usage patterns change with the type of job being performed. For example, when reading a paper document, User B switches between the paper and the computer frequently, and therefore has higher blink rate and head motion. Whereas programming causes focused attention, reduced head motion and increased time between breaks. The more frequent eye blink rate on Friday might indicate eye fatigue after a week of work.

General mobility The mobility feature f_{mob} for the far-away ambient cameras and the close-by dedicated and personal cameras is shown Figure 9.13. Near-zero feature-values are evident with each camera and class. Main reason for this are the synchronization problems, as non-mobile moments have been included in the mobile classes. The mobile and in-transit classes with ambient cameras largely overlap, largely due to viewpoint limitation in computing the mobility feature.

Both ambient and close-by cameras were used jointly to infer the mobility state of the person. Each decision is compared to the annotated label and only matching decisions count as correct. All the figures show correct classification-ratios for each class, for all samples (overall), and averaged across the classes (normalized), per day. Same legend and axis-notation apply to Figures 9.14-9.16. The colored lines in the figures highlight the classification ratios that are discussed in the text in percentages.



(a) Ambient Cameras 1&3 (b) Close-By Cameras

Figure 9.13: *The histograms of mobility feature f_{mob} for each mobility class, starting from top: regular, mobile and in-transit classes. The red boxes indicate the feature-ranges which are used for each of the mobility classes.*

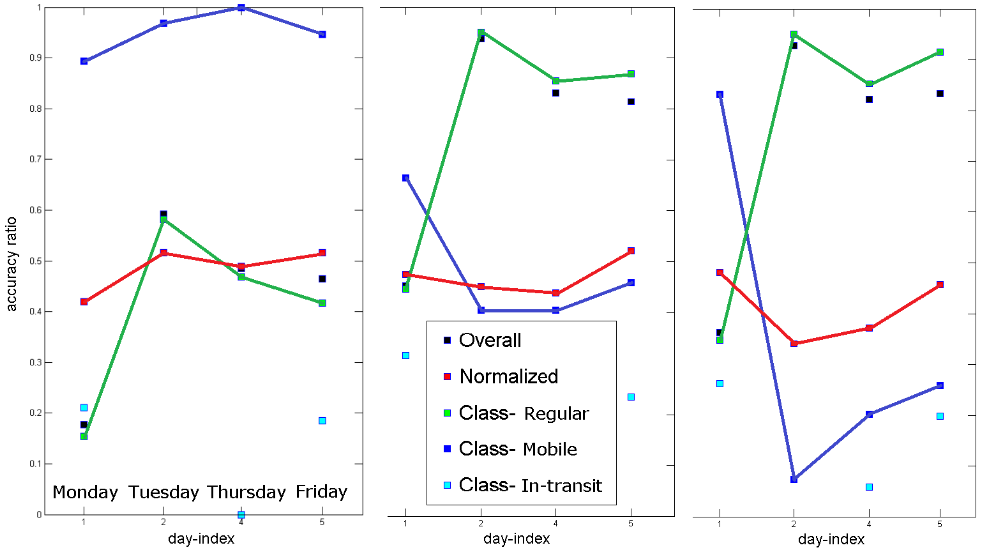
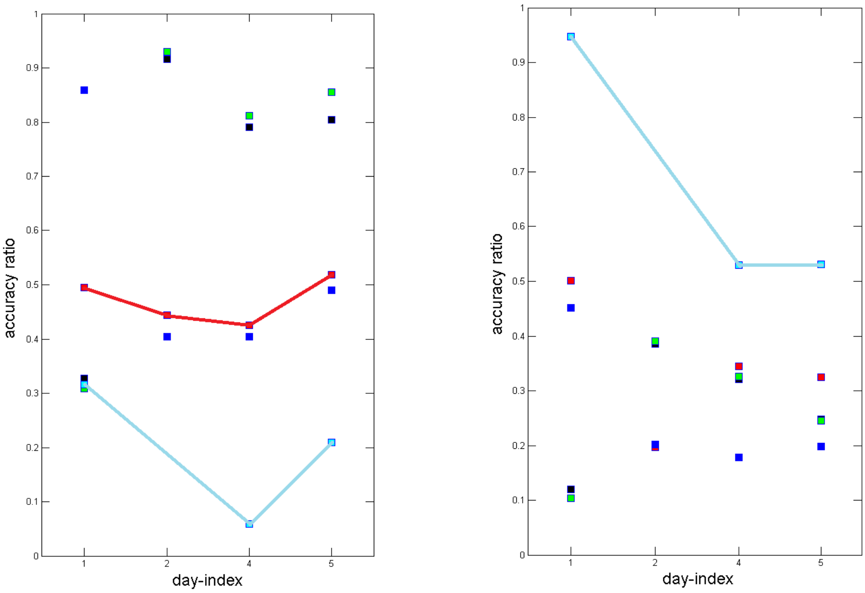


Figure 9.14: *Classification Accuracy: Centralized Fusion with Common Opinion methods for the sequences of Oct 3rd, 4th, 6th, and 7th, starting from left with Fusion Levels: feature, score, and decision.*

The classification accuracy, ranging [0.0, 1.0] with 1.0 representing 100% correct classification of all frames of particular day, of centralized common opinion fusion at each of the basic fusion levels is shown in Figure 9.14. Fusion of features gives better results for *mobile* (dark blue curve) than for *regular* (green curve) class, contrary to fusion of scores and decisions. As normalized (red curve) feature fusion is slightly better (49%) than score (47%) or decision fusion (41%).

The centralized hybrid level results of both fusion approaches are given in Figure 9.15. With the sensitive approach, *in-transit* (light blue curve) class is detected much better, but the detection of the other two classes suffers.

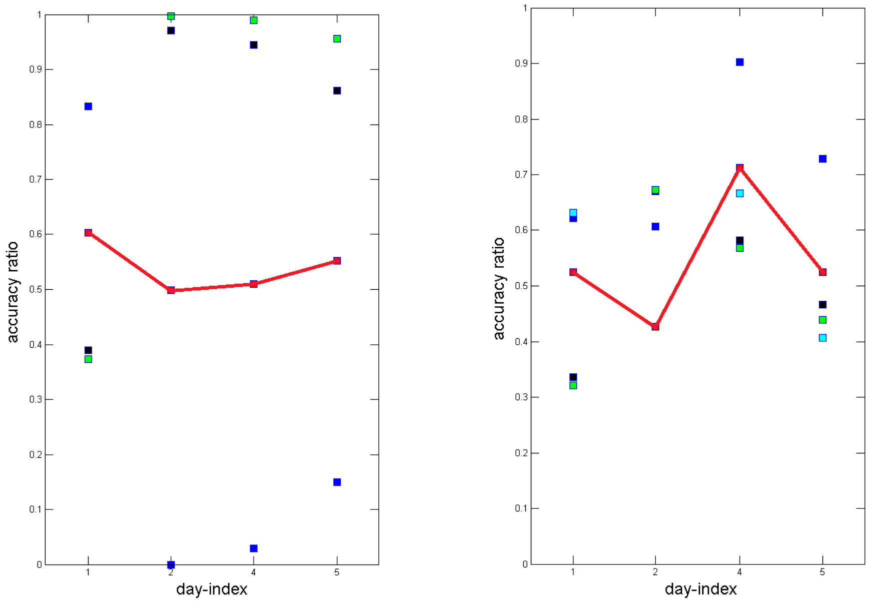
Correspondingly, the hybrid results of both ambient and close-by cameras separate with common opinion approach are shown in Figure 9.16. Ambient cameras can reach a normalized accuracy (red curve) of 54% and close-by 55%, which on average is better than centralized with 47%.



(a) Common Opinion

(b) Sensitive

Figure 9.15: *Classification Accuracy: Centralized Hybrid of both Method approaches for the sequences of Oct 3rd, 4th, 6th, and 7th.*



(a) Ambient Cameras 1&3

(b) Close-By Cameras

Figure 9.16: *Classification Accuracy: Hierachical Hybrids of the Common Opinion method for the sequences of Oct 3rd, 4th, 6th, and 7th.*

9.5 Discussion

Video-based analysis can provide much beneficial information about working habits. With a webcam descriptive measures on ergonomics by the desk can be extracted. By adding more cameras the range of analysis can be increased, but often the quality and comparability of data suffers as problems emerge, e.g., in data synchronization and image processing.

On Application

The proposed system was designed with capabilities to accommodate learning the user's behavior pattern, and thus providing ergonomic reminders based on the inferred user activities. Further extensions to the proposed system include additional well-being measures, possibly given by sensors of other modalities with increased level of context information. In future work, having tools to analyze working behavior enables the use of personalized recommendation systems that can provide the suggestions specifically to the particular worker according to his agenda, status and activities.

On Fusion

Both close-by and far-away cameras were used to compute a measure of worker's general mobility as he or she moved and interacted within the entire office environment. These observations were affected in the experiments by video synchronization issues and image processing challenges. Fusion of data can help in coping with these problems.

Fusion Approach The aim of inference was to detect the mobility status of a worker (*regular, mobile, and in-transit*) according to the current multi-camera evidence on movement mobility. In general, a common opinion between the cameras would suffice as there is, e.g., no particular reason to be more sensitive to detecting the more mobile classes from the ergonomics point-of-view. However, in these experiments it was noticed from the feature data that due to the image processing and synchronization problems a more sensitive approach would be necessary in order to detect moments of bigger mobility. Therefore, both majority and sensitive approaches were studied with the fusion methods [Design Rule 1].

Vision Network Two kinds of cameras were used in recording the experiment data. The personal webcams were connected by USB to the worker's laptop, which recorded the video. The dedicated and ambient IP-cameras were connected by Ethernet-cables to the local network, from which another laptop recorded all the six video-streams simultaneously. The resolution, frame-rate and video-codecs were defined as same for all the video sources. All automatic camera adjustments were disabled and the same manual settings were put in [Design Rule 2].

The webcams did not record timestamps, which were included in the video-frames with the IP-cameras. A clap of hands in the start of the sequence by the worker was used to initially synchronize the webcams to the IP-cameras. Some key moments were used later in the sequence to re-initialize the synchronization. Because different media (USB vs Ethernet) were used in the transmission, and laptops with different resources and computational loads were used to record the streams, the synchronization constantly fluctuated. This fluctuation introduced a lot of false values for the mobility feature as, e.g., a webcam would be observing a standing up motion while an IP-camera is few frames behind thus still observing the person sitting. All these problems hindered the timely delivery of corresponding frames [Design Rule 3].

As the user's personal screen would be used to provide the ergonomic feedback and analysis results, the amount of visual artifacts that would be picked up by the cameras could be considered very small. Therefore, no special fusion precautions would be necessary by the observing vision network [Design Rule 4].

Fusion Architecture A common approach to fusion has been to gather data directly to a single processing unit that is responsible for performing the fusion. In addition to such a fully centralized fusion architecture a hierarchical fusion architecture was studied by examining separately the accuracy of distant ambient cameras and close-by cameras. The hierarchical architecture was found to provide beneficial conditions when encountering data that behaves similarly only within a group of sensors. For example, the mobility feature data of user-A from the dedicated and personal camera was highly correlating. This fact could be exploited by grouping these two views under the same node, and through fusion achieving better classification [Design Rule 6].

Fusion Level Another major aspect of a fusion process is the type of data to be combined. All three levels of feature, score and decision were studied under the presented architectures and fusion methods. In the experiments conducted, no clear accuracy differences between the levels were found. The hybrid level that exhaustively combines the results from all the three levels did show a more stable trend in accuracy across the days than any individual fusion level [Design Rule 9].

Fusion Method Two different approaches to fusion methods were studied across all the fusion levels and architectures. Fusion method defines the operations that are applied to the set of estimates in order to get a single joint estimate. Based on the experiments it can be concluded that the decisions made by the system can be heavily influenced by the fusion method alone. For example, a preference to detect classes of higher importance can be made possible by the simple choice of a particular fusion method approach, regardless of the fusion level and the structure it is processed in [Design Rule 11].

In summary Considering a simple aspect of camera distance w.r.t. the observed person, it appears to be safe to group cameras with similar distances, as they will most likely encounter similar limitations due to diminished visibility to the person. This can be considered even more likely when considering cameras with the same height and tilt. By grouping cameras with similar measurements, more stability in data is achieved and better results from fusion can be expected. It would appear that by applying fusion methods that are sensitive to hard-to-detect activities, their detection can be enhanced, regardless of the level of data and the manner in which the data is gathered.

Chapter 10

General Discussion

Vision networks have great potential in monitoring people. There are many challenges in vision processing and pattern recognition. Coping with these challenges is much supported by the multi-view data. The richness of data in having the observations from many viewpoints does not guarantee a better decision making. It is the fusion approach that makes the difference by increasing the certainty in the decisions made, and by making properties visible that otherwise would not have been noticed.

Section 10.1 starts the chapter by briefly addressing the potential and future directions of vision network technology. The past, current and future application opportunities are discussed in section 10.2. The important role of multi-view fusion in and the applicability of the proposed vision fusion framework for vision systems is briefly reviewed in section 10.3. Section 10.4 provides remarks on future issues and directions concluding the chapter and this thesis.

10.1 Vision Networks - Potential

The two most important building blocks of vision networks are algorithms for computer vision and machine learning. Vision processing delivers the observations of a person by detecting, tracking and matching. Machine learning is the brain that takes the observations and turns the raw input data into knowledge about actions, activities and behavior of a person.

Computer vision is full of challenges, because both people and the environments they inhabit are dynamic. Both of them can take different forms and appear different under changing conditions. The research conducted for this thesis exploited so called passive vision solutions, that is, the cameras only captured the light emitted by the scene. They did not transmit any additional light sources or patterns onto the scene. In all the studies conducted, the key to achieving robust system accuracy was to use hybrid approaches. Hybrid approaches exploit complementary algorithms in which the aim is to counteract the drawbacks or limitations of an algorithm by the benefits of another algorithm and vice versa.

Still, two common major challenges of initialization and update remain in vision processing. First, nearly always there are assumptions of the scene layout and of the status of people that have to be made when a vision system starts to operate. These assumptions need to be clearly stated, and either followed by the operators, or detected by the system and acted upon accordingly. Second, the update of models and parameters of vision processes is crucial for maintaining stable and beneficial results. An update at a bad time can easily lead to a spiral of bad parameter changes, which again can lead to even worse system updates and eventually complete system failure.

Inference of patterns of activities or behavior is performed with an increasing number of probabilistic models. These models are commonly re-structured and specified to better suit the end application of the vision system. The model is usually selected based on previous experiences and recommendations by others. Much literature does cover comparisons between the popular choices, but still the preference of a model seems to remain in the hands of the practitioner. No universally superior models exist currently that would outperform others in all applications, and it is very unlikely to happen anytime soon. It is very common to tailor a widely accepted model that has been researched in similar problems to a specific purpose. Perhaps the most improvements in near future will be done due to increased storage space and computing power, as these enable the use of larger feature-spaces and databases of gathered data. Having access to much larger data pools, and having resources to process them simultaneously are most probably the key factors in future improvements of inference.

10.2 Emerging Applications - Opportunities

Vision networks have been used in a variety of applications. The first applications were introduced mainly for military surveillance from which surveillance-related applications have spread to wide public spaces such as airports and other public transport areas. In these applications, common tasks have been the detection of irregular situations, uncommon behaviors and suspicious properties that are left behind. Shops as well have been covered with cameras for the purpose of monitoring the personnel and the clientele w.r.t. thefts and violence.

A lot of interest has emerged for using vision systems to provide not only security, but also to offer user-support and promote well-being and social connectedness. In this thesis studies were presented for four such application contexts. First, an application for eliciting social connectedness through a dynamic silhouette representation between remote sites was proposed in chapter 6. Second application presented in chapter 7 expanded from one room to covering an entire home with continuous capture and inference of user activity, which was considered as an additional property that could be shared over a remote visual link. Third, an application for shops was presented in chapter 8 based on repetitive behavior analysis that would automatically provide support for a shopper who cannot find the item(s) they are looking for. The fourth application in chapter 9 concerned self-observation for office-workers who might not be following ergonomic guidelines and thus most likely encountering issues with injuries and difficulties in working efficiently.

Many more application opportunities for personal spaces have emerged. Remote elderly care and monitoring is one such field that can have major impact on general well-being and save much resources in health-care. Automotive industry has also started to exploit cameras not only looking outwards of the car, such as in parking and lane changing assistance, but also in observing what is happening inside the car. For example, analysis of driver attention has been one of the new impressive safety solutions. Additionally, many new personalized services for homes such as automatic lighting controls are expected to benefit from the scene understanding only possible by vision systems.

10.3 Fusion in Vision Networks

In a vision network, the key factor behind solid accuracy are mechanisms of sensor fusion. These mechanisms can be designed in four steps. First, the *fusion approach* defines the specific purpose of the fusion process. Second, the *fusion architecture* defines the manner in which the multi-view data is gathered together. Third, the *fusion level* defines the type of data that is gathered and combined. Fourth, the *fusion method* defines how the gathered data is combined into a single estimate.

10.3.1 Benefit from Multi-View Fusion

Through the experiments in this thesis many options for fusion were studied, and the benefits from fusion were highlighted in relation to the proposed applications.

The first benefit of multi-view fusion is in creating more complete observations than is possible by a single camera. An example of this was given in chapter 6. The 3D shape and 3D location of a person in the room were estimated based on multiple silhouettes from a calibrated vision network. The fragility of a calibration-based fusion approach was demonstrated as a 13 cm deviation in calibration for a space of size 200x200x200 cm resulted in a zero-volume shape. In general, fusion methods for occupancy testing that exploited soft values and prior camera information could overcome some of the errors in silhouette segmentation and calibration, but at the same time falsely declaring some space occupied.

The second benefit of fusion is in increasing visibility to a person. This benefit could be understood as a specialization of the first benefit. An example of this was given in chapter 6, in which some experiments were conducted for understanding the effect of camera configuration to silhouette-based 3D shape estimation by fusion of occupancy tests. For providing the tightest shape estimate and thus the best visibility to a person, it was found that the camera configuration should surround the person from all sides, have no opposite cameras, and have the cameras placed by maximizing the vertical displacement between adjacent cameras.

The third major benefit of fusion is in increasing the range of making observations. In chapter 7, the recognition of six activities was performed across multiple rooms of a house environment. Fusion methods that were based on selection of a single view could already perform well. In the ideal selection case of always finding the most appropriate view to use, classification accuracies up to 93 and 99% were achieved in the studied cases. Fusion methods that were based on combination of views outperformed the practical selection-based fusion methods commonly by a margin of 10%.

In chapter 8, the detection of repetitive behavior was conducted in a small shop covered by three partially overlapping camera views. Decision-level fusion offered consistently better accuracy than feature-level, when dealing with sensors of low individual classification accuracy ($\leq 50\%$). In chapter 9, a vision network was capturing a worker in an office with four camera views partially overlapping. Based on multi-camera observations this system tried to infer the general mobility class out of the three options: regular, mobile, and in-transit. A hierarchical fusion architecture was studied and was found to provide beneficial conditions for inference, when encountering feature-data that shares similar distributions only between a group of sensors.

The fourth and most commonly discussed benefit of fusion is increasing the certainty of a particular property. In chapter 6 the certainty of an occupied part of 3D space was jointly decided upon by multiple cameras. The more cameras were added, the tighter the shape estimate became, as the certainty in the occupancy of a part of space disappeared. In chapter 8, the detection of repetition was improved even with poorly performing cameras by combining their decisions through a view selection based fusion method, achieving an accuracy above 51% for a four-class classification. Based on the experiments conducted in chapter 7 it appeared that fusion at the level of features was more attractive within well controlled conditions. In contrast, decision-level fusion appeared to offer better accuracy, if the people shared fewer characteristics and cameras were placed at various distances. It should be noted, that the margin between feature-level and decision-level fusion was only in the magnitude of a few percent.

The fifth benefit of fusion is in increasing the visibility of a particular property. In chapter 9 two different approaches to fusion methods across all the fusion levels were studied. It was noticed that the decisions on general mobility class that are made by the system, can be heavily influenced by the fusion method approach. For example, a certain class can be preferred by the fusion method, if having any evidence supporting the class among the multiple cameras. This sensitivity in fusion can be achieved almost regardless of the fusion level and architecture. However, a hierarchical fusion architecture might in some cases make it complicated to implement a locally sensitive but globally robust-to-noise fusion.

10.3.2 Applicability of Fusion Framework

The framework that was proposed in chapter 5 sets out to systemize the approach for building vision networks that will greatly benefit from fusion. By including the aspects of fusion already into the design process and not as an afterthought to superior single camera processing, the vision system is able to exploit the full potential of multi-view data fusion. The main points of the proposed Vision Fusion Framework are:

1. showcases the common vision network architecture of three levels.
2. explains the major aspects of each vision module and how they affect fusion opportunities.
3. highlights the important interface between vision and fusion enabled by data alignment and association.
4. defines the fusion process as a four-step approach.
5. explains the major aspects of the fusion approach and how they affect fusion opportunities.
6. showcases the entire framework in feedback-connection to the environment and the user.
7. guides the design process with fusion rules that are connected to the specifics tiers of fusion and vision network.

10.4 Conclusion

The research in this thesis has demonstrated and discussed only a very small part of the ambient intelligence solutions, the benefits they hold, and the impacts they can have in the future. It is most probable that as more and more applications reach maturity and provide thus irreplaceable value to their user, the technology of vision networks will be in high regard.

All the applications studied in this thesis have the noble goal of enabling services to either protect or support the well-being of people. There has been an increasing amount of discussion on the concerns about the privacy issues of vision networks as more and more aspects of regular life are being monitored. For example, public authorities like police are using video cameras paired with software capable of reading the license plates of vehicles and querying information from both public and private databases. Similarly, an increase in the amount of CCTV-cameras has been noticeable within big cities in countries such as China and England.

Privacy concerns are one of the major challenges for the deployment of vision networks in ambient intelligence applications. The cameras installed outdoors that raise privacy concerns, are now been moved indoors and especially to places traditionally viewed as private spaces. Based on the discussions the author has had during this research, there seems to be three factors that can help in this shift. First, the vision systems should be transparent. That is, what is been processed, and what is sent to which place, should be visible to the user. Second, the user should have full control over what is been sent and stored, and who can connect to his private network. Third, the value of the service should outweigh the workload and other related negative effects. If these aspects would be met, it is likely that ambient intelligence solutions would have much more demand in the future.

A major new trend in imaging technologies is active sensing. Technologies such as structured-light and time-of-flight are been developed to mature level with many impressive applications, such as Microsoft Kinect, already catching the interest of the public. These technologies are able to capture much more detailed aspects in much more robust manner than has ever been possible with passive cameras. This should lead to computation of more detailed features with more stable behavior over time. With such a good basis of features, the accuracy of inference should similarly rise and offer systems that can be deployed in much more error-sensitive scenarios such as automatic alarming in elderly homecare.

In conclusion, the key mechanism in powerful vision networks is intelligent fusion methodology for handling multi-view data. Only by a systematic approach to the many aspects of fusion can practitioners directly share a common terminology and collaborate more efficiently for further developing fusion related to sensing technologies. The proposed vision fusion framework provides an attempt to streamline the design process and to find a set of common fusion rules within a common structure for practitioners to discuss, refine and follow.

Part III

Appendices

Bibliography

[1] Deutsch D: **Quantum computation**. *Physics World* 1992.

[2] Adleman LM: **Molecular computation of solutions to combinatorial problems**. *Science* 1994, **266**(11):1021–1024.

[3] Mell P, Grance T: **The NIST Definition of Cloud Computing**. Tech. rep. 2009.

[4] Hutt DL, Snell KJ, Bélanger PA: **Alexander Graham Bell's Photophone**. *Opt. Photon. News* 1993, **4**(6):20–25.

[5] Gokturk SB, Yalcin H, Bamji C: **A Time-Of-Flight Depth Sensor - System Description, Issues and Solutions**. In *Proceedings of the Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)*, *Volume 3* 2004.

[6] Silberman N, Fergus R: **Indoor scene segmentation using a structured light sensor**. In *IEEE International Conference on Computer Vision Workshops (ICCV Workshops)* 2011:601 –608.

[7] Mitchell HB: *Multi-Sensor Data Fusion: An Introduction.* Springer Publishing Company, Incorporated, 1st edition 2007.

[8] Tenney R, Sandell N, of Technology Laboratory for Information MI, Systems D: *Detection with Distributed Sensors.* LIDS-P, Laboratory for Information and Decision Systems, Massachusetts Institute of Technology 1980.

[9] Chair Z, Varshney P: **Optimal Data Fusion in Multiple Sensor Detection Systems**. *IEEE Transactions on Aerospace and Electronic Systems* 1986, **AES-22**:98–101.

[10] Aziz A, Tummala M, Cristi R: **Optimal data fusion strategies using multiple-sensor detection systems**. In *Conference Record of the Thirty-First Asilomar Conference on Signals, Systems amp; Computers*, *Volume 1* 1997:941 –945.

[11] Jain A, Nandakumar K, Ross A: **Score normalization in multimodal biometric systems**. *Pattern Recognition* 2005, **38**(12):2270 – 2285.

[12] Veeramachaneni K, Osadciw L, Ross A, Srinivas N: **Decision-level fusion strategies for correlated biometric classifiers**. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* 2008:1–6.

[13] Shoaib M, Dragon R, Ostermann J: **Context-aware visual analysis of elderly activity in a cluttered home environment**. *EURASIP J. Adv. Sig. Proc.* 2011.

[14] Diaz Alonso J, Ros Vidal E, Rotter A, Muhlenberg M: **Lane-Change Decision Aid System Based on Motion-Driven Vehicle Tracking**. *Vehicular Technology, IEEE Transactions on* 2008, **57**(5):2736 –2746.

[15] Smith P, Shah M, da Vitoria Lobo N: **Determining driver visual attention with one camera**. *Intelligent Transportation Systems, IEEE Transactions on* 2003, **4**(4):205 – 218.

[16] Harwig R, Aarts E: **Ambient Intelligence: invisible electronics emerging**. In *Proceedings of the IEEE International Interconnect Technology Conference* 2002:3–5.

[17] van der Poel C, Pessolano F, Roovers R, Widdershoven F, de Walle G, Aarts E, Christie P: **On ambient intelligence, needful things and process technologies**. In *Proceeding of the 30th European Solid-State Circuits Conference (ESSCIRC)* 2004:3–10.

[18] Aarts E: **Ambient intelligence: a multimedia perspective**. *IEEE Multimedia Journal* 2004, **11**:12–19.

[19] Marchesotti L, Piva S, Regazzoni C: **Structured context-analysis techniques in biologically inspired ambient-intelligence systems**. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans* 2005, **35**:106 – 120.

[20] Calbi A, Dore A, Marcenaro L, Regazzoni CS: **Multimodal cognitive system for immersive user interaction**. In *Proceedings of the First International Conference on Immersive Telecommunications*, Brussels, Belgium: ICST 2007:3:1–3:6.

[21] Tabar AM, Keshavarz A, Aghajan H: **Smart home care network using sensor fusion and distributed vision-based reasoning**. In *Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks* 2006:145–154.

[22] Zhang Z, Scanlon A, Yin W, Yu L, Venetianer PL: **Video Surveillance using a Multi-Camera Tracking and Fusion System**. In *Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications (M2SFA2)*, Marseille, France: Andrea Cavallaro and Hamid Aghajan 2008.

[23] Han J, Bhanu B: **Fusion of color and infrared video for moving human detection**. *Pattern Recogn.* 2007, **40**(6):1771–1784.

[24] Zhai Y, Tian YL, Feris R, Brown L, Bobbitt R, Hampapur A, Pankanti S, Fan Q, Yanagawa A, Velipasalar S: **Composite Event Detection in Multi-Camera and Multi-Sensor Surveillance Networks**. In *Multi-Camera Networks: Concepts and Applications*, Elsevier 2009.

[25] Gatica-Perez D, Lathoud G, Odobez JM, McCowan I: **Audiovisual Probabilistic Tracking of Multiple Speakers in Meetings**. *Trans. Audio, Speech and Lang. Proc.* 2007, **15**(2):601–616.

[26] Talantzis F, Pnevmatikakis A, Constantinides AG: **Audio-visual active speaker tracking in cluttered indoors environments**. *Trans. Sys. Man Cyber. Part B* 2009, **39**:7–15.

[27] Gandhi T, Trivedi MM: **Vehicle mounted wide FOV stereo for traffic and pedestrian detection**. In *International Conference on Image Processing* 2005:121–124.

[28] Fang Y, Yamada K, Ninomiya Y, Horn B, Masaki I: **Comparison between infrared-image-based and visible-image-based approaches for pedestrian detection**. In *Proceedings of IEEE Intelligent Vehicles Symposium* 2003:505–510.

[29] Kämpchen N, Dietmayer K: **Fusion of Laserscanner and Video for Advanced Driver Assistance Systems**. In *Proceedings of 11th World Congress on Intelligent Transportation Systems (ITS)* 2004.

[30] Ghahroudi MR, Sabzevari R: **Multisensor Data Fusion Strategies for Advanced Driver Assistance Systems**. In *Sensor and Data Fusion.* Edited by Milisavljevic N, InTech 2009.

[31] Svoboda T, Martinec D, Pajdla T: **A Convenient Multi-Camera Self-Calibration for Virtual Environments**. *PRESENCE: Teleoperators and Virtual Environments* 2005, **14**(4):407–422.

[32] Durrant-Whyte HF: **Sensor models and multisensor integration**. *International Journal of Robotics Research* 1988, **7**(6):97–113.

[33] Brunelli R: *Template Matching Techniques in Computer Vision: Theory and Practice.* Wiley Publishing 2009.

[34] Andriluka M, Roth S, Schiele B: **People-tracking-by-detection and people-detection-by-tracking**. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* 2008.

[35] Dasarathy BV: **Sensor fusion potential exploitation-innovative architectures and illustrative applications** 1997, **85**:24–38.

[36] Elmenreich W: **Sensor Fusion in Time-Triggered Systems**. *PhD thesis*, Technische Universität Wien, Institut für Technische Informatik, Treitlstr. 3/3/182-1, 1040 Vienna, Austria 2002.

[37] White FE: **Data Fusion Lexicon, Joint Directors of Laboratories**. Tech. rep., Naval Ocean Systems Center 1987.

[38] Durrant-Whyte HF: *Integration, Coordination and Control of Multi-Sensor Robot Systems*. Norwell, MA, USA: Kluwer Academic Publishers 1987.

[39] Wald L: **A European proposal for terms of reference in data fusion** 1998.

[40] Boström H, Andler SF, Brohede M, Johansson R, Karlsson A, van Laere J, Niklasson L, Nilsson M, Persson A, Ziemke T: **On the Definition of Information Fusion as a Field of Research**. Tech. Rep. HS- IKI -TR-07-006, University of Skövde, School of Humanities and Informatics 2007.

[41] Laurentini A: **The Visual Hull Concept for Silhouette-Based Image Understanding**. *IEEE Trans. Pattern Anal. Mach. Intell.* 1994, **16**:150–162.

[42] Matusik W, Buehler C, Raskar R, Gortler SJ, McMillan L: **Image-based visual hulls**. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques (SIGGRAPH)* 2000:369–374.

[43] Torresan H, Turgeon B, Ibarra-castanedo C, Hebert P, Maldague X: **Advanced Surveillance Systems: Combining Video and Thermal Imagery for Pedestrian Detection**. In *In Proc. of SPIE, Thermosense XXVI, Volume 5405* 2004:506–515.

[44] Surya G, Subbarao M: **Depth from Defocus by Changing Camera Aperture: A Spatial Domain Approach** 1993.

[45] Xu N, Ahuja N: **On the Use of Depth-From-Focus in 3D Object Modelling from Multiple Views**.

[46] Zhou C, Lin S, Nayar SK: **Coded Aperture Pairs for Depth from Defocus**. In *IEEE International Conference on Computer Vision (ICCV)* 2009.

[47] Petschnigg G, Szeliski R, Agrawala M, Cohen M, Hoppe H, Toyama K: **Digital photography with flash and no-flash image pairs**. In *ACM SIGGRAPH 2004 Papers*, New York, NY, USA: ACM 2004:664–672.

[48] Henry P, Krainin M, Herbst E, Ren X, Fox D: **RGB-D Mapping: Using Depth Cameras for Dense 3D Modeling of Indoor Environments** 2011.

[49] Wu AH C: **Opportunistic feature fusion-based segmentation for human gesture analysis in vision networks**. In *Proc. of IEEE SPS-DARTS* 2007.

[50] Yu D, Frincke D: **Alert confidence fusion in intrusion detection systems with extended Dempster-Shafer theory**. In *Proceedings of the 43rd annual Southeast regional conference, Volume 2.*

[51] Garcia-Salicetti S, Mellakh MA, Allano L, Dorizzi B: **Multimodal Biometric Score Fusion: The Mean Rule vs. Support Vector Classifiers**.

[52] Qian M, Aguilar M, Zachery KN, Privitera C, Klein S, Carney T, Nolte LW: **Decision-level fusion of EEG and pupil features for single-trial visual detection analysis.** *IEEE transactions on bio-medical engineering* 2009, **56**(7):1929–1937.

[53] Mansoorizadeh M, Charkari NM: **Hybrid feature and decision level fusion of face and speech information for bimodal emotion recognition**. In *14th International CSI Computer Conference (CSICC)* 2009:652–657.

[54] Sanderson C, Paliwal K: **Information Fusion and Person Verification Using Speech and Face Information**. In *Technical Report IDIAP* 2002.

[55] Sanderson C, Paliwal KK: **Identity Verification Using Speech And Face Information**. In *Digital Signal Processing* 2004:449–480.

[56] Arrow K: *Social Choice & Individual Values.* Monograph (Yale University), Yale University Press 1963.

[57] Kim K, Roush F: *Introduction to mathematical consensus theory.* Lecture notes in pure and applied mathematics, M. Dekker 1980.

[58] Utete SW, Barshan B, Ayrulu B: **Voting as Validation in Robot Programming**. *I. J. Robotic Res.* 1999, **18**(4):401–413.

[59] Dawes RM: **The robust beauty of improper linear models in decision making.** *American psychologist* 1979, **34**(7):571.

[60] Radova V, Psutka J: **An Approach to Speaker Identification Using Multiple Classifiers**. In *Proceedings of the 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* 1997.

[61] Klein L: **A Boolean algebra approach to multiple sensor voting fusion**. *IEEE Transactions on Aerospace and Electronic Systems* 1993, **29**:317 –327.

[62] Kacelenga R, Erickson D, Palmer D: **Voting fusion adaptation for landmine detection**. *Aerospace and Electronic Systems Magazine, IEEE* 2003, **18**(8):13 – 19.

[63] Giraud C, Jouvencel B: **Sensor selection: a geometrical approach**. In *IEEE/RSJ International Conference on Intelligent Robots and Systems 95. 'Human Robot Interaction and Cooperative Robots', Volume 2* 1995:555–560.

[64] Viet NH, Munthe-Kaas E, Plagemann T: **Energy-balanced sensor selection for social context detection**. In *IEEE International Conference on Pervasive Computing and Communications Workshops* 2012:32–37.

[65] Armaghani F, Gondal I, Kamruzzaman J: **Dynamic Sensor Selection for Target Tracking in Wireless Sensor Networks**. In *Vehicular Technology Conference (VTC Fall), 2011 IEEE* 2011:1 –6.

[66] Bajovic D, Sinopoli B, Xavier J: **Sensor Selection for Event Detection in Wireless Sensor Networks**. *Signal Processing, IEEE Transactions on* 2011, **59**(10):4938 –4953.

[67] Prokaj J, Medioni G: **Accurate efficient mosaicking for Wide Area Aerial Surveillance**. In *Proceedings of the 2012 IEEE Workshop on the Applications of Computer Vision (WACV)* 2012:273–280.

[68] Bellman R: *Dynamic Programming.* Dover Books on Mathematics, Dover 2003.

[69] Wu C, Khalili AH, Aghajan H: **Multiview activity recognition in smart homes with spatio-temporal features**. In *Proceedings of the Fourth ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC)* 2010:142–149.

[70] Kittler J, Hatef M, Duin RPW, Matas J: **On combining classifiers**. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 1998, **20**:226–239.

[71] Stiefelhagen R, Yang J, Waibel A: **Estimating focus of attention based on gaze and sound**. In *Proceedings of the 2001 workshop on Perceptive user interfaces (PUI)* 2001:1–9.

[72] Wu H, Siegel M, Stiefelhagen R, Yang J: **Sensor fusion using Dempster-Shafer theory [for context-aware HCI]**. In *Proceedings of the 19th IEEE Instrumentation and Measurement Technology Conference (IMTC), Volume 1* 2002:7–12.

[73] Funk N: **A Study of the Kalman Filter applied to Visual Tracking** 2003.

[74] Gan Q, Harris C: **Comparison of two measurement fusion methods for Kalman-filter-based multisensor data fusion**. *Aerospace and Electronic Systems, IEEE Transactions on* 2001, **37**:273 –279.

[75] Puranik S, Jannett T: **A comparison of the tracking performance of some distributed multisensor data fusion algorithms based on Kalman filter methods**. In *Proceedings of the 35th Southeastern Symposium on System Theory* 2003:455–459.

[76] Kasper R, Schmidt S: **Sensor-data-fusion for an autonomous vehicle using a Kalman-filter**. In *6th International Symposium on Intelligent Systems and Informatics (SISY)* 2008:1 –5.

[77] Hall D, McMullen S: *Mathematical Techniques in Multisensor Data Fusion.* Artech House Information Warfare Library, Artech House 2004.

[78] Bayes M, Price M: **An Essay towards Solving a Problem in the Doctrine of Chances. By the Late Rev. Mr. Bayes, communicated by Mr. Price, in a letter to John Canton, M. A. and F. R. S.** *Philosophical Transactions (1683-1775)* 1963.

[79] Lima P: **A Bayesian Approach to Sensor Fusion in Autonomous Sensor and Robot Networks**. *IEEE Instrumentation Measurement Magazine* 2007, **10**.

[80] Smaili C, El Najjar M, Francois: **Multi-sensor Fusion Method Using Bayesian Network for Precise Multi-vehicle Localization**. In *11th International IEEE Conference on Intelligent Transportation Systems (ITSC)* 2008:906 –911.

[81] Guan L, Franco JS, Pollefeys M: **3D Occlusion Inference from Silhouette Cues**. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 2007:1–8.

[82] Dempster AP: **Upper and lower probabilities induced by a multivalued mapping**. *Annals of Mathematical Statistics* 1967, **38**:325–339.

[83] Shafer G: *A mathematical theory of evidence.* Princeton university press 1976.

[84] Pong P, Challa S: **Empirical analysis of generalised uncertainty measures with dempster shafer fusion**. In *10th International Conference on Information Fusion* 2007:1 –9.

[85] Mezai L, Hachouf F, Bengherabi M: **Score fusion of face and voice using Dempster-Shafer theory for person authentication**. In *11th International Conference on Intelligent Systems Design and Applications (ISDA)* 2011:894 –899.

[86] Stearns SD: **Optimum Detection Using Multiple Sensors**. In *Proc. of Carnahan Conf. on Security Technology* 1983.

[87] Xiong N, Svensson P: **Multi-sensor management for information fusion: issues and approaches**. *Information Fusion* 2002, **3**(2):163 – 186.

[88] Sinha A, Chen H, Danu D, Kirubarajan T, Farooq M: **Estimation and Decision Fusion: A Survey**. In *IEEE International Conference on Engineering of Intelligent Systems* 2006:1 –6.

[89] Steinberg AN, Bowman CL, White FE: **Revisions to the JDL data fusion model**. In *IEEE International Conference on Engineering of Intelligent Systems*, *Volume 3719*, SPIE 1999:430–441.

[90] Shulsky A, Schmitt G: *Silent Warfare: Understanding the World of Intelligence.* Intelligence & national security library, Brassey's 2002.

[91] Boyd JA: **A Discourse on Winning and Losing** 1987.

[92] Markin M, Harris C, Bernhardt M, Austin J, Bedworth M, Greenway P, Johnston R, Little A, Lowe D: **Technology foresight on data fusion and data processing** 1997.

[93] Bedworth M, O'Brien J: **The Omnibus model: a new model of data fusion?** *IEEE Aerospace and Electronic Systems Magazine* 2000, **15**:30–36.

[94] Aghajan H, Wu C, Kleihorst R: **Distributed Vision Networks for Human Pose Analysis**. In *Signal Processing Techniques for Knowledge Extraction and Information Fusion.* Edited by Mandic D, Golz M, Kuh A, Obradovic D, Tanaka T, Springer US 2008:181–200.

[95] Karakaya M: **Collaborative Solutions to Visual Sensor Networks, PhD Dissertation**. *PhD thesis*, University of Tennessee 2011.

[96] Durrant-Whyte H, Stevens M: **Data fusion in decentralised sensing networks**. In *Proceedings of the International Conference on Information Fusion* 2001.

[97] Hartley RI, Zisserman A: *Multiple View Geometry in Computer Vision.* Cambridge University Press, ISBN: 0521540518, second edition 2004.

[98] Zhong ZG, Yi JQ, Zhao DB: **A Robust Two Feature Points Based Depth Estimation Method**. *Acta Automatica Sinica* 2005, **31**(5):693–698.

[99] Dasarathy BV: **Elucidative fusion systems - an exposition**. *Information Fusion* 2000, **1**:5–15.

[100] Ruyter de B, Huijnen C, Markopoulos P, IJsselsteijn W: *Creating Social Presence through Peripheral Awareness*, Book Industry Services (Bis) 2007 :69–71.

[101] Apperley M, McLeod L, Masoodian M, Paine L, Phillips M, Rogers B, Thomson K: **Use of Video Shadow for Small Group Interaction Awareness on a Large Interactive Display Surface**. In *Proc. Fourth Australasian User Interface Conference(AUIC)* 2003.

[102] Miwa Y, Ishibiki C: **Shadow communication: system for embodied interaction with remote partners**. In *Proceedings of the 2004 ACM conference on Computer supported cooperative work (CSCW)* 2004:467–476.

[103] Yasuda S, Hashimoto S, Koizumi M, Okude N: **Teleshadow: feeling presence in private spaces**. In *ACM SIGGRAPH sketches* 2007.

[104] Grivas K: **Digital Selves: Devices for intimate communications between homes**. *Pers. Ubiquit. Comput.* 2006, **10**:66–76.

[105] Hoover A, Olsen B: **A real-time occupancy map from multiple video streams**. In *Proceedings of IEEE International Conference on Robotics and Automation, Volume 3* 1999:2261–2266.

[106] Yenilmez L, Temeltas H: **Real time multi-sensor fusion and navigation for mobile robots**. In *9th Mediterranean Electrotechnical Conference (MELECON), Volume 1* 1998:221–225.

[107] Elfes A: **Using occupancy grids for mobile robot perception and navigation**. *Computer* 1989, **22**(6):46 –57.

[108] Berger JO: *Statistical Decision Theory and Bayesian Analysis.* New York: Springer-Verlag 1985.

[109] Lee D, , Nakamura Y: **Motion capturing from monocular vision by statistical inference based on motion database: Vector field approach**. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* 2007, :617–623.

[110] Plänkers R, Fua P: **Tracking and modeling people in video sequences**. *Comput. Vis. Image Underst.* 2001, **81**(3):285–302.

[111] Han H, Wang Z, Liu J, Li Z, Li B, Han Z: **Adaptive background modeling with shadow suppression**. In *Proc. of Intelligent Transportation Systems* 2003:720–724.

[112] Cheung KM, Kanade T, Bouguet JY, Holler M: **A real time system for robust 3D voxel reconstruction of human motions**. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Volume 2* 2000:714 – 720.

[113] Graphics-Optics-Vision Group MPII: **Kung-Fu Girl (2005). A synthetic test sequence for multi-view reconstruction and rendering research.**[http://www.mpiinf.mpg.de/departments/irg3/kungfu/].

[114] Al-Khaffaf H, Talib A, Salam R: **Removing salt-and-pepper noise from binary images of engineering drawings**. In *19th International Conference on Pattern Recognition (ICPR08).* 2008:1–4.

[115] Farhadi A, Forsyth D, White R: **Transfer Learning in Sign language**. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 2007.

[116] Farhadi A, Tabrizi MK: **Learning to Recognize Activities from the Wrong View Point**. In *ECCV '08: Proceedings of the 10th European Conference on Computer Vision*, Berlin, Heidelberg: Springer-Verlag 2008:154–166.

[117] Cherla S, Kulkarni K, Kale A, Ramasubramanian V: **Towards fast, view-invariant human action recognition**. *Computer Vision and Pattern Recognition Workshop* 2008, **0**:1–8.

[118] Collins R, Gross R, Shi J: **Silhouette-based Human Identification from Body Shape and Gait**. In *Int. Conf. on Face and Gesture* 2002:351–356.

[119] Barnich O: **Motion detection and human recognition in video sequences**. *PhD thesis*, University of Liège, Belgium 2010.

[120] Rabiner LR: **A tutorial on hidden markov models and selected applications in speech recognition**. In *Proceedings of the IEEE* 1989:257–286.

[121] Mccallum A, Freitag D, Pereira F: **Maximum Entropy Markov Models for Information Extraction and Segmentation**. Morgan Kaufmann 2000:591–598.

[122] Lafferty J, McCallum A, Pereira F: **Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data** 2001.

[123] S Singh SV, Ragheb H: **MuHAVi: A Multicamera Human Action Video Dataset for the Evaluation of Action Recognition Methods**. In *2nd Workshop on Activity monitoring by multi-camera surveillance systems (AMMCSS)* August 29, 2010.

[124] Lu Y, Cohen I, Zhou XS, Tian Q: **Feature selection using principal feature analysis**. In *MULTIMEDIA '07: Proceedings of the 15th international conference on Multimedia* 2007:301–304.

[125] **The Free Dictionary - Behavior** 2010.

[126] Watson JB: **Psychology as the Behaviorist Views it**. *Psychological Review* 1913, **20**:158–177.

[127] Mazur JE: **Mathematical models and the experimental analysis of behavior.** *Journal of the experimental analysis of behavior* 2006, **85**(2):275–291.

[128] Wickens CD: *The effects of control dynamics on performance*, Wiley-Interscience 1986 chap. 39.

[129] Moray N: *Monitoring behavior and supervisory control*, Wiley-Interscience 1986 chap. 40.

[130] Helbing D: **A mathematical model for the behavior of pedestrians**. *Behavioral Science* 1991, **36**(4):298–310.

[131] Mangel M, Clark CW: *Dynamic Modeling in Behavioral Ecology.* Princeton University Press 1989.

[132] Pentland A, Lin A: **Modeling and Prediction of Human Behavior**. *Neural Computation* 1995, **11**:229–242.

[133] Barber D, Cemgil A: **Graphical Models for Time-Series**. *IEEE Signal Processing Magazine* 2010.

[134] Wray RE, Laird JE: **Variability in Human Behavior Modeling for Military Simulations**. In *Behavior Representation in Modeling & Simulation Conference (BRIMS)* 2003.

[135] Chen CW, Aztiria A, Aghajan H: **Learning human behaviour patterns in work environments**. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* 2011:47 –52.

[136] Benabbas Y, Ihaddadene N, Djeraba C: **Motion pattern extraction and event detection for automatic visual surveillance**. *J. Image Video Process.* 2011, :7:1–7:15.

[137] Stauffer C, Grimson W: **Learning patterns of activity using real-time tracking**. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2000, **22**(8):747–757.

[138] Keezer WS, Fenic AP, Nelson BL: **Representation of user transaction processing behavior with a state transition matrix**. In *Proceedings of the 24th conference on Winter simulation (WSC)* 1992:1223–1231.

[139] Muhammad S, Wachowicz M, de Carvalho L: **Evaluation of wavelet transform algorithms for multi-resolution image fusion**. In *Proceedings of the 5th International Conference on Information Fusion, Volume 2* 2002:1573–1580.

[140] Naidu V: **Multi-resolution image fusion by FFT**. In *International Conference on Image Information Processing (ICIIP)* 2011:1 –6.

[141] Palubinskas G, Reinartz P: **Multi-resolution, multi-sensor image fusion: general fusion framework**. In *Joint Urban Remote Sensing Event (JURSE)* 2011:313 –316.

[142] Wu C, Khalili AH, Aghajan H: **Multiview activity recognition in smart homes with spatio-temporal features**. In *Proceedings of the Fourth ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC)* 2010:142–149.

[143] Määttä T, Härmä A, Aghajan H: **On efficient use of multi-view data for activity recognition**. In *Proceedings of the Fourth ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC)* 2010:158–165.

[144] Louis W, Plataniotis KN: **Co-Occurrence of Local Binary Patterns Features for Frontal Face Detection in Surveillance Applications**. *EURASIP J. Image and Video Processing* 2011.

[145] Gualdi G, Prati A, Cucchiara R: **Contextual information and covariance descriptors for people surveillance: an application for safety of construction workers**. *J. Image Video Process.***2011**:9:1–9:16.

[146] Bouttefroy PLM, Bouzerdoum A, Phung SL, Beghdadi A: **Integrating the projective transform with particle filtering for visual tracking**. *J. Image Video Process.***2011**:6:1–6:11.

[147] Stauffer C, Grimson W: **Adaptive background mixture models for real-time tracking**. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, *Volume 2* 1999:637–663.

[148] Dalal N, Triggs B: **Histograms of oriented gradients for human detection**. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, *Volume 1* 2005:886–893.

[149] Haralick RM, Shanmugam K, Dinstein I: **Textural Features for Image Classification** 1973, (6):610–621.

[150] Miyamoto E, Merryman T: **Fast Calculation of Haralick Texture Features**.

[151] Bishop C: *Neural networks for pattern recognition.* Oxford University Press, USA 1995.

[152] OSHA: **Laboratory Safety Ergonomics for the Prevention of Musculoskeletal Disorders in Laboratories**. *http://www.osha.gov/Publications/laboratory/OSHAfactsheet-laboratory-safety-ergonomics.pdf* 2012.

[153] Chen CW, Maatta T, Bing-Yung Wong K, Aghajan H: **A Framework for Providing Ergonomic Feedback Using Smart Cameras**. In *Proceedings of the 6th ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC)* 2012.

[154] Hansen D, Ji Q: **In the Eye of the Beholder: A Survey of Models for Eyes and Gaze**. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2010, **32**:478 –500.

[155] Valenti R, Sebe N, Gevers T: **What Are You Looking at? - Improving Visual Gaze Estimation by Saliency**. *International Journal of Computer Vision* 2012, **98**(3):324–334.

[156] Chen CW, Aghajan H: **Multiview Social Behavior Analysis in Work Environments**. In *ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC)*, Ghent, Belgium 2011.

[157] Chau M, Betke M: **Real time eye tracking and blink detection with USB cameras**. Tech. rep., Boston University 2005.

[158] Lalonde M, Byrns D, Gagnon L, Teasdale N, Laurendeau D: **Real-time eye blink detection with GPU-based SIFT tracking**. In *Proceedings of the Fourth Canadian Conference on Computer and Robot Vision*, Washington, DC, USA: IEEE Computer Society 2007:481–487.

[159] Harville M: **A Framework for High-Level Feedback to Adaptive, Per-Pixel, Mixture-of-Gaussian Background Models**. In *Computer Vision - ECCV 2002, Volume 2352 of* Lecture Notes in Computer Science, Springer Berlin Heidelberg 2002:543–560.

[160] Wang H, Suter D: **A re-evaluation of mixture of Gaussian background modeling [video signal processing applications]**. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Volume 2* 2005:1017–1020.

[161] Chen D, Denman S, Fookes C, Sridharan S: **Accurate Silhouettes for Surveillance - Improved Motion Segmentation Using Graph Cuts**. In *International Conference on Digital Image Computing: Techniques and Applications (DICTA)* 2010:369–374.

[162] Li I, Dey A, Forlizzi J: **A stage-based model of personal informatics systems**. In *Proceedings of the 28th international conference on Human factors in computing systems (HCI)* 2010:557–566.

[163] Jaimes A: **Sit straight (and tell me what I did today): a human posture alarm and activity summarization system**. In *Proceedings of the 2nd ACM workshop on Continuous archival and retrieval of personal experiences (CARPE)* 2005:23–34.

[164] Shirom A, Toker S, Alkaly Y, Jacobson O, Balicer R: **Work-Based Predictors of Mortality: A 20-Year Follow-Up of Healthy Employees**. *Health Psychology* 2011, **30**(3):268–275.

[165] Chen CW, Aztiria A, Ben Allouch S, Aghajan H: **Understanding the influence of social interactions on individual's behavior pattern in a work environment**. In *Proceedings of the 2nd International Conference on Human Behavior Unterstanding*, Springer-Verlag 2011:146–157.

[166] Wu C, Aghajan H: **User-Centric Environment Discovery With Camera Networks in Smart Homes**. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* 2010.

[167] **Seeing Machines - faceAPI**. *http://www.seeingmachines.com/product/faceapi/*.

Glossary

This glossary presents briefly the most important general and fusion specific terms related to the research in this thesis, and concludes by listing the most often used abbreviations.

List of Terms

General Terms

Vision Network	(VN) is a system that infers person's appearance, activity or behavior based on observations given by multiple cameras.
Computer Vision	(CV) consists of algorithms that extract information on objects from images.
Feature Extraction	is a process which computes measures from an object usable for discrimination between the object-classes.
Pattern Recognition	(PR) consists of algorithms that detect specific patterns of, e.g., shape and trajectory.
Machine Learning	(ML) is a process in which a system learns or is trained for the common patterns of a phenomenon.
Intrinsic Calibration	a process which detects the extent an image gets distorted by a camera.
Extrinsic Calibration	a process which discovers how cameras are situated w.r.t. each other and the scene.
Data Alignment	a process in which observation data is aligned both temporally and spatially.
Data Association	a process which finds correspondences for observations both within and between views.
Ambient Intelligence	(AmI) is a vision for the use of small, integrated, and hidden set of ubiquitous sensors within ordinary environments for supporting people in their daily lifes.

Fusion-Specific Terms

Sensor Fusion	a process of combining observations from multiple sensors into a single estimate.
Catastrophic Fusion	a situation in which the average accuracy of a multi-sensor system is worse than that of a single sensor.
Fusion Approach	defines the purpose of fusion either for consistency or responsiveness.
Fusion Architecture	defines the manner in which multi-camera observations are gathered.
Fusion Level	defines the type of observations (image, feature, score or decision) to be combined.
Fusion Method	defines the algorithms that are used to combine the observations.
Vision Fusion Framework	(VFF) is a generic structure proposed in this thesis that details and combines VN and sensor fusion for guiding the design of vision systems by introducing a set of design rules within a common structure.
Design Rule	gives an instructional statement considering fusion based on theoretical and/or experimental results.

List of Abbreviations

ADL	activities of daily-life
LAN	local area network
ML	machine learning
DT	decision tree
NN	artificial neural network
SVM	support vector machines
PGM	probabilistic graphical models
CRF	conditional random field
CPU	central processing unit
PTZ	pan/tilt/zoom
RFID	radio-frequency identification
ROI	region-of-interest
VOI	volume-of-interest
POI	person-of-interest
FOV	field-of-view
RES	resolution in pixels
FPS	frame-rate
VNA	vision network architecture
IBVH	image-based visual hull
AMR	arithmetic mean rule
LRT	likelihood ratio
LDA	linear discriminant analysis
MMSE	minimum-mean-square-error
WLS	weighted Linear Sum
IP	internet protocol
RTP	real-time transport protocol
NTP	network time protocol

FA	fusion architecture
FCFA	fully centralized FA
FDFA	fully distributed FA
HFA	hierarchical FA
EFS	elucidative fusion systems
GT	ground truth
FAR	false acceptance rate
FRR	false rejection rate
PFA	principal feature analysis
AHM	action history matrix
GMM	gaussian mixture model
HOG	histogram of gradients
BG	image background
BG	image foreground
MHI	motion history image

Summary

Sensor Fusion in Smart Camera Networks for Ambient Intelligence

Support and security to people and energy-efficiency of buildings are some of the many benefits of smart environments of the future, which use embedded networks of different sensors to understand what individuals are doing daily, and based on the patterns found they adaptively provide personalized support to these activities (Ambient Intelligence).

Many of the smart environments incorporate unobtrusive vision networks as one of the sensing technologies. Wearable and planted sensors can provide more robust performance, but they can heavily restrict the freedom and movements of a user. Vision networks (VN) have multiple cameras providing their observations of a scene, which can consist of vast areas or multiple rooms. Based on these observations VN can infer the appearance, location, activity or behavior of a person.

In building technology that is capable of observing people, many particular challenges are faced. People are deformable and adaptive. This makes us difficult to detect, follow and describe. The same person can wear different clothes, and the dimensions of the body vary between individuals. The same activities can be performed by different gestures and pace. All these differences are affected by context such as the time of day.

The richness of data created by having the observations from many viewpoints, does not guarantee improved accuracy for a VN. It is the intelligent data fusion mechanisms that make the difference by increasing the certainty in the decisions made, and by making properties visible that otherwise would have not been noticed.

Having the technology to build vision networks capable of human behavior interpretation the question becomes, how to systematically across various applications design vision systems in a modular manner while integrating intelligent fusion approaches to the system design? What are the common approaches and their consequences to fusion potential and exploitation?

This thesis aims to provide the necessary theory within systematical tools for tackling the above problems in two parts. The first part (Chapters 1-4) provides the necessary theoretical background on VNs and sensor fusion theory specific to VNs. The second part (Chapters 5-10) presents the proposed vision fusion framework with a set of design rules and covers a multitude of fusion experiments in different application domains.

Many types of vision networks based on coverage, placement, calibration, and collaboration (Chapter 2) have been successfully used with other complementary sensing technologies in various traditional applications within surveillance and security and in emerging applications for the more private spaces in homes and offices. In order to provide a clean and intuitive division of the many aspects of sensor fusion, three different aspects of fusion within VNs should be defined (Chapters 3-4). The fusion architecture, the fusion level, and the fusion method are the building blocks of a fusion approach. Based on the structured approach to VNs and fusion, a vision fusion framework (VFF) is proposed (Chapter 5). The aim of the framework is to provide a systematic way for designing vision networks with great emphasis on the support, creation and exploitation of fusion potential.

Four different VN-applications are presented and experimented with (Chapters 6-9). By chapter 8, a functional and robust algorithm for automatic single-person tracking was developed based on a hybrid approach of image segmentation and template matching. A background subtraction based silhouette segmentation provides accurately the areas of the new-to-the-scene objects within fairly static indoor settings. Template matching based on HOG-SVM pedestrian detection is used to initialize the track of a person and to update changes to the track and size of the person; all these verified by the silhouette candidates. Only extreme changes in illumination or close-to-person-size moving objects, such as doors, can create problems for the hybrid algorithm.

The best type of a feature in describing a property of a person should be invariant to the scale and orientation of the person. The design of a scale-invariant feature can be done by integrating a compensation mechanism for person distance, e.g., based on the size of the silhouette. In contrast, designing a orientation-invariant feature gets more difficult the more detail the feature is trying to represent. Sensor fusion becomes critical in neglecting false observations of vision processing, while still giving enough importance, e.g., for the hard-to-see movements of arms.

Before any other aspect of fusion is selected for a problem, the approach of fusion has to be defined (Chapter 9). The goal of fusion can be considered two-fold. First, fusion is to build certainty in the final estimate based on the observations from multiple views. Second, sometimes it is even more important to not miss specific types of actions, which might be only detected by a single camera. Depending on the choice on the fusion approach, the architecture, the level, and the method of fusion are defined.

Only if resources are severily limited, or compensation for camera noise or failure can not be implemented, should one revert to another fusion architecture than the fully centralized fusion architecture (FCFA) (Chapter 9). FCFA gathers all the sensory data to a centralized unit. This offers maximum flexibility in exploiting any of the fusion options, and the best conditions for dealing with noise.

No conclusive evidence was found in favor of a specific fusion-level across applications. However, in silhouette-based activity recognition (Chapter 7), fusion at the feature-level outperformed fusion at decision-level, when considering cameras with similar relative locations/orientations and individuals with similar appearance/pace. In contrast, with the recorded multi-view data that has more variability, the decision-level fusion was better. Similarly, in tracking-based repetition detection (Chapter 8) fusion at the decision-level performed better, with the more challenging recorded dataset. In overall, when applying similar fusion methods, it would appear that decision-level fusion offers better robustness. Additionally, by using multiple fusion-levels by combining equally the results they provide (Chapters 6, 7 and 9), a more stable accuracy is achievable.

Some tendencies were discovered of the most common fusion methods. Already by adaptive view selection based on pre-defined criteria, can better accuracy be reached than when relying on a single camera (Chapter 7). However this accuracy is heavily influenced by the suitability of the criteria used to select the view. This makes view selection slightly volatile.

A safer fusion method is that of sensor voting that uses, e.g., all the decisions available through majority voting to come to a consensus (Chapters 6, 7 and 9). Majority voting has been widely used, and it seems to be a safe option for achieving good accuracy results, and therefore represents a good benchmark for other fusion methods. However, it should be noted that when dealing with a group of views that cannot individually provide reasonable accuracies, this common consensus approach is not able to match the accuracy that can be provided by a simple sensor selection (Chapter 8).

Among the methods studied in this thesis, the best accuracy a fusion method can provide is given by relative influence, which determines the importance of cameras w.r.t. each other (Chapter 6). The relative influence is implemented by assigning a weight to each sensor. By computing these weights based on prior information, e.g., on the camera layout w.r.t. the user, is better accuracy achieved. By adapting the weights on-line, can further robustness to noise conditions be gained. However, on-line adaption of weights is even more complicated to robustly implement than that of view-selection criteria (Chapters 7 and 8). Therefore, it would be advised to use stable prior information driven weights when applicable than those computed only by adaptive online-techniques.

To conclude, the key mechanism in powerful vision networks is intelligent fusion of the multi-view data. It is hoped that by a systematic approach to the four main aspects of fusion, can practitioners directly share a common terminology and collaborate more efficiently in developing fusion related to sensing technologies.

Acknowledgments

It is time for the best part, a chance to thank all the people that I have had the pleasure of meeting and working with during my PhD years.

My first experiences go back to the exciting PhD-corner office at Philips Research Laboratories HTC-36 in Eindhoven. These guys made the jump into PhD research feel like a chance to do something exciting, and to have fun while delving into the research. A nostalgic thank to Nicolas, Tom, Alberto and Tobias.

I was introduced to a great gathering of like-minded individuals, who had a common goal of creating more exposure to our community of PhDs and postDocs within Philips (PPC). I met many wonderful PhDs and made a few good friends. A big thank you for all those fun weekly meetings and coffee-breaks: Marjolein, Marian, Marleen, Juergen, Aaron, Tim, Angel, Marian, Janneke, Nele, Alberto, Greg, Jos, Emile and the active members. I hope PPC is there to stay.

I am very grateful for the travel and collaboration with other research groups I was able to experience here in the Netherlands, and in California. I am grateful to Boris de Ruyter for organizing the Stanford cooperation, and to Martha Russell for making me always feel welcomed and helping me in connecting to interesting people. The exposure to other interests and ways of thinking, have been an eye-opening experience, which I hope will stay with me forever. I would also like to thank the dissertation committee consisting of Peter de With, Paul Havinga, Pieter Jonker, and Wilfried Philips for the valuable time they invested in evaluating this dissertation and for providing me with constructive feedback that improved this dissertation.

Along the PhD years, I was fortunate to find good friends within my colleagues. The research and travel in California would have not been the same without you: Tao, Chen, Louis, Amir, Nima, Stephanie, Samar, and JJ. Joris and Maurits, thank you both for the fun and educational times in California and Eindhoven. To all my friends at Philips, thank you for making every working day more fun than I dared to imagine, thank you Nemanja, Sven, Vincent, Enrique, Christoph, Illapha and Emile. I am also grateful to meeting all the interns and others PhDs during my time at Philips Research, as this experience would not have been the same without you.

Aki, I cannot imagine these years without your professional and thoughtful support. It truly was a pleasure to have you as my daily supervisor, thank you for listening and for all the help you gave. My gratitude goes beyond our professional collaboration. I am also very grateful to the reassuring supervision given by Henk at TU/e, and to the trend-aware and generous host who had me visiting his laboratory every year, thank you Hamid.

My family has always been there when I have needed their help. Even when working abroad, and sometimes even beyond the Atlantic, they have kept in touch, asking how I am doing and occasionally sending packages with all kinds of Finnish goodies. Just to give me a taste of home, reminding me of the tranquility I am able to enjoy, whenever I have had the time to travel back to Sotkamo. Kiitos koko perheelle.

Alina, you and your help and support have been the rock I've been able to rely on. I am forever grateful.

Tommi Määttä
Eindhoven, May 2013

Curriculum Vitae

Tommi Määttä was born on April 12th, 1982, in Sotkamo, Finland. He attended Sotkamo Sports High School and graduated in 2001. In 2002 he entered the Information Technology degree programme at Tampere University of Technology (TUT) in Finland. He majored in signal processing, specializing in video and image processing, and minored in hypermedia. MSc Degree Thesis was written as a result of a 10 month internship at Philips Research Laboratories Eindhoven, Netherlands starting in June 2007. This work was supervised by Irek Defee (TUT), Aki Härmä (Philips Research) and Hamid Aghajan (Stanford). He graduated as MSc with distinction (cum laude) from TUT August of 2008.

He enrolled as a PhD candidate to the department of Electrical Engineering of Eindhoven University of Technology (TU/e) in the Netherlands right after graduating from TUT. His PhD-project was organized as collaboration between two universities, TU/e and Stanford University in Palo Alto, California USA, and Philips Research DSP/VIP-groups in Eindhoven. In the spirit of collaboration, one year of the four was spent as a visiting scholar in Stanford at the WSNL/AirLAB research group. This PhD project was carried out under the supervision of Henk Corporaal (TU/e), Aki Härmä (Philips Research) and Hamid Aghajan (Stanford). The results of this PhD project are presented in this dissertation.