

ACTA

*Jari Hannuksela*

CAMERA BASED MOTION  
ESTIMATION AND  
RECOGNITION FOR HUMAN-  
COMPUTER INTERACTION

FACULTY OF TECHNOLOGY,  
DEPARTMENT OF ELECTRICAL AND INFORMATION ENGINEERING,  
UNIVERSITY OF OULU;  
INFOTECH OULU,  
UNIVERSITY OF OULU





ACTA UNIVERSITATIS OULUENSIS  
C Technica 313

*JARI HANNUKSELA*

**CAMERA BASED MOTION  
ESTIMATION AND RECOGNITION  
FOR HUMAN-COMPUTER  
INTERACTION**

Academic dissertation to be presented, with the assent of  
the Faculty of Technology of the University of Oulu, for  
public defence in Auditorium TS101, Linnanmaa, on  
December 19th, 2008, at 12 noon

OULUN YLIOPISTO, OULU 2008

Copyright © 2008  
Acta Univ. Oul. C 313, 2008

Supervised by  
Professor Janne Heikkilä

Reviewed by  
Doctor Ville Kyrki  
Professor Roope Raisamo

ISBN 978-951-42-8977-4 (Paperback)  
ISBN 978-951-42-8978-1 (PDF)  
<http://herkules.oulu.fi/isbn9789514289781/>  
ISSN 0355-3213 (Printed)  
ISSN 1796-2226 (Online)  
<http://herkules.oulu.fi/issn03553213/>

Cover design  
Raimo Ahonen

OULU UNIVERSITY PRESS  
OULU 2008

## **Hannuksela, Jari, Camera based motion estimation and recognition for human-computer interaction**

Faculty of Technology, Department of Electrical and Information Engineering, University of Oulu, P.O.Box 4500, FI-90014 University of Oulu, Finland; Infotech Oulu, University of Oulu, P.O.Box 4500, FI-90014 University of Oulu, Finland

*Acta Univ. Oul. C 313, 2008*

Oulu, Finland

### ***Abstract***

Communicating with mobile devices has become an unavoidable part of our daily life. Unfortunately, the current user interface designs are mostly taken directly from desktop computers. This has resulted in devices that are sometimes hard to use. Since more processing power and new sensing technologies are already available, there is a possibility to develop systems to communicate through different modalities. This thesis proposes some novel computer vision approaches, including head tracking, object motion analysis and device ego-motion estimation, to allow efficient interaction with mobile devices.

For head tracking, two new methods have been developed. The first method detects a face region and facial features by employing skin detection, morphology, and a geometrical face model. The second method, designed especially for mobile use, detects the face and eyes using local texture features. In both cases, Kalman filtering is applied to estimate the 3-D pose of the head. Experiments indicate that the methods introduced can be applied on platforms with limited computational resources.

A novel object tracking method is also presented. The idea is to combine Kalman filtering and EM-algorithms to track an object, such as a finger, using motion features. This technique is also applicable when some conventional methods such as colour segmentation and background subtraction cannot be used. In addition, a new feature based camera ego-motion estimation framework is proposed. The method introduced exploits gradient measures for feature selection and feature displacement uncertainty analysis. Experiments with a fixed point implementation testify to the effectiveness of the approach on a camera-equipped mobile phone.

The feasibility of the methods developed is demonstrated in three new mobile interface solutions. One of them estimates the ego-motion of the device with respect to the user's face and utilises that information for browsing large documents or bitmaps on small displays. The second solution is to use device or finger motion to recognize simple gestures. In addition to these applications, a novel interactive system to build document panorama images is presented.

The motion estimation and recognition techniques presented in this thesis have clear potential to become practical means for interacting with mobile devices. In fact, cameras in future mobile devices may, for the most of time, be used as sensors for self intuitive user interfaces rather than using them for digital photography.

*Keywords:* computer vision, facial feature extraction, head tracking, mobile device, motion estimation, user interface



## Preface

The research for this thesis was carried out in the Machine Vision Group of the Department of Electrical and Information Engineering at the University of Oulu, Finland during the years 2003-2008.

I would like to express my gratitude to Professor Janne Heikkilä for supervising the thesis. Without his invaluable ideas and guidance, this thesis would never have been completed. I am also grateful to Professor Matti Pietikäinen for allowing me to work in his research group and supervising me at an early stage of my studies. I would also like to thank Professor Olli Silvén for his expert advice.

I am grateful to Professor Roope Raisamo and Dr. Ville Kyrki for reviewing the dissertation manuscript. Their valuable comments greatly improved the quality of the thesis. I wish also to thank Gordon Roberts for the language revision.

I would like to thank my co-authors Pekka Sangi, Dr. Mark Barnard, Sami Huttunen, and Dr. Markus Turtinen for their valuable contributions. I would also like to thank Dr. Matti Niskanen for commenting on the manuscript. I would like to express my gratitude to Dr. David Doermann and Xu Liu for supervising and helping me during my stay at the University of Maryland. Many thanks to my colleagues in the Machine Vision Group for creating a most pleasant atmosphere. In particular, I wish to highlight the fruitful coffee break discussions on many interesting topics.

The financial support provided by the Infotech Oulu Graduate School, the Academy of Finland, the Finnish Funding Agency for Technology and Innovation (Tekes), the Nokia Foundation, the Tauno Tönning Foundation, and the Seppo Säynäjäkangas Science Foundation is gratefully acknowledged.

Finally, I would like to thank my father Torsti and mother Ritva for their love and care over the years. Most of all, I want to thank my wife Eija and our son Jere for their love and understanding during my studies.

Oulu, November 2008

Jari Hannuksela



## Abbreviations

2-D	Two-dimensional
3-D	Three-dimensional
API	Application programming interface
CPU	Central processing unit
DCT	Discrete cosine transform
DOF	Degrees of freedom
DOS	Disk operating system
EKF	Extended Kalman filter
EM	Expectation maximisation
FPS	Frames per second
GMM	Gaussian mixture model
GPS	Global positioning system
GPU	Graphical processing unit
GUI	Graphical user interface
HCI	Human-computer interaction
HMM	Hidden Markov model
HW	Hardware
I/O	Input and output
IR	Infrared
KF	Kalman filter
KLT	Kanade-Lucas-Tomasi
$k$ -NN	$k$ -nearest neighbour
LBP	Local binary pattern
LED	Light-emitting diode
LS	Least squares
MAP	Maximum a posteriori
ML	Maximum likelihood
NCC	Normalised colour coordinates
P3P	Perspective-three-point
PC	Personal computer
PDA	Personal digital assistant

RAM	Random-access memory
RANSAC	Random sample consensus
SAD	Sum of absolute differences
SIFT	Scale invariant feature transform
SKF	Switching Kalman filter
SLAM	Simultaneous localisation and mapping
SSD	Sum of squared differences
SW	Software
WIMP	Window, icon, menu, pointing device
WLS	Weighted least squares
ZSSD	Zero mean sum of squared differences
<b>B</b>	Structuring element
<i>BB</i>	Bounding box
<b>C</b>	Covariance matrix
<b>d</b>	Displacement vector
<i>d<sub>feature</sub></i>	Measured feature distance
<i>D<sub>feature</sub></i>	Reference feature distance
<i>E<sub>feature</sub></i>	Evaluation function for features
<i>f(x,y)</i>	Image intensity at a point $(x,y)$
<b>H</b>	Observation matrix
<b>P</b>	Feature position in world coordinates
<b>p</b>	Feature position in camera coordinates
<b>Q</b>	Process covariance matrix
<b>R</b>	Observation covariance matrix
<i>r,g,b</i>	Chromaticity coordinates in NCC colour space
<i>T<sub>L</sub></i>	Image intensity threshold
<i>T<sub>V</sub></i>	Valley image intensity threshold
<i>u,v</i>	Image coordinates
<i>V(x,y)</i>	Valley image intensity at a point $(x,y)$
<b>x</b>	State vector
<i>x,y,z,X,Y,Z</i>	Cartesian coordinates
<b>z</b>	Measurement vector
<b>θ</b>	Model parameter vector

## List of original articles

This dissertation is based on the following articles, which are referred to in the text by their Roman numerals (I–IX):

- I Hannuksela J, Heikkilä J & Pietikäinen M (2004) A real-time facial feature based head tracker. Proc. 6th International Conference on Advanced Concepts for Intelligent Vision Systems. Brussels, Belgium: 267–272.
- II Hannuksela J, Sangi P, Turtinen M & Heikkilä J (2008) Face tracking for spatially aware mobile user interfaces. Proc. International Conference on Image and Signal Processing. Cherbourg-Octeville, Normandy, France. Lecture Notes in Computer Science 5099: 405–412.
- III Hannuksela J, Sangi P & Heikkilä J (2007) Vision-based motion estimation for interaction with mobile devices. Computer Vision and Image Understanding 108(1–2): 188–195.
- IV Sangi P, Hannuksela J & Heikkilä J (2007) global motion estimation using block matching with uncertainty analysis. Proc. 15th European Signal Processing Conference, Poznan, Poland: 1823–1827.
- V Hannuksela J, Huttunen S, Sangi P & Heikkilä J (2007) Motion-based finger tracking for user interaction with mobile devices. Proc. 4th European Conference on Visual Media Production. London, UK.
- VI Hannuksela J, Sangi P & Heikkilä J (2006) Motion-based handwriting recognition for mobile interaction. Proc. 18th International Conference on Pattern Recognition. Hong Kong, China 4: 397–400.
- VII Barnard M, Hannuksela J, Sangi P & Heikkilä J (2007) A vision based motion interface for mobile phones. Proc. 5th International Conference on Computer Vision Systems. Bielefeld, Germany: 1–10.
- VIII Hannuksela J, Barnard M, Sangi P & Heikkilä J (2008) Adaptive motion-based gesture recognition interface for mobile phones. Proc. 6th International Conference on Computer Vision Systems. Santorini, Greece. Lecture Notes in Computer Science 5008: 271–280.
- IX Hannuksela J, Sangi P, Heikkilä J, Liu X & Doermann D (2007) Document image mosaicing with mobile phones. Proc. 14th International Conference on Image Analysis and Processing, Modena, Italy: 575–580.

The writing of Papers I, II, VI and IX, was mainly carried out by the author of this dissertation, who was also responsible for the research made in these papers. The research of Paper III was carried out together with Mr. Sangi. The author was in charge of writing and implementing the experiments on the mobile phone. Paper IV was mainly written by Mr. Sangi. The author was closely involved in developing the techniques proposed and participated in the writing process. The writing and the experiments of Paper V were mostly performed by the author. The idea of the method used in this work came from Prof. Heikkilä. The research of Papers VII and VIII was done together with Dr. Barnard. With Paper VII, the author was involved in developing the system presented,

participated in the writing process, and was responsible for experimental set-up. In Paper VIII, the author was also in charge of the experiments and writing. Prof. Heikkilä participated in the finalisation of the paper and provided guidance and comments, as with the other papers.

# Contents

<b>Abstract</b>	
<b>Preface</b>	<b>5</b>
<b>Abbreviations</b>	<b>7</b>
<b>List of original articles</b>	<b>9</b>
<b>Contents</b>	<b>11</b>
<b>1 Introduction</b>	<b>15</b>
1.1 Computer vision based interfaces . . . . .	16
1.2 The contribution of the thesis . . . . .	18
1.3 Summary of original papers . . . . .	20
<b>2 Face and facial feature detection</b>	<b>23</b>
2.1 Previous approaches . . . . .	23
2.1.1 Feature based methods . . . . .	24
2.1.2 Image based methods . . . . .	25
2.2 Searching facial features via colour and morphology . . . . .	26
2.2.1 Detecting face regions using skin colour . . . . .	27
2.2.2 Searching facial features using morphology . . . . .	27
2.2.3 Face and facial feature analysis . . . . .	28
2.3 Detecting face and facial features using texture . . . . .	29
2.3.1 LBP features . . . . .	29
2.3.2 Classifier . . . . .	30
2.4 Discussion . . . . .	31
2.5 Summary . . . . .	33
<b>3 Head pose estimation and tracking</b>	<b>35</b>
3.1 Visual tracking . . . . .	35
3.2 Head pose estimation and gaze direction detection . . . . .	36
3.3 Related work . . . . .	37
3.3.1 2-D image based methods . . . . .	37
3.3.2 3-D model based methods . . . . .	38
3.4 Proposed new head tracking approaches . . . . .	41
3.4.1 Head tracking using facial features . . . . .	41
3.4.2 Head tracking using facial features and optical flow . . . . .	42

3.5	Discussion .....	44
3.6	Summary .....	46
<b>4</b>	<b>Motion estimation for mobile user interaction</b>	<b>47</b>
4.1	Related work on motion based interaction .....	47
4.1.1	Approaches using markers .....	48
4.1.2	Markerless approaches .....	48
4.2	Computing 2-D image motion .....	49
4.2.1	Gradient based methods .....	50
4.2.2	Feature based methods .....	51
4.2.3	Estimating motion using a sparse set of feature blocks .....	52
4.3	Camera ego-motion estimation .....	56
4.3.1	Estimating dominant global parametric motion .....	56
4.3.2	Modelling motion .....	57
4.3.3	Dominant global motion estimation .....	57
4.4	Multiple motion estimation .....	58
4.4.1	Data association .....	59
4.4.2	Object tracking using the Kalman-EM algorithm .....	60
4.5	Discussion .....	61
4.6	Summary .....	62
<b>5</b>	<b>Motion analysis in applications</b>	<b>63</b>
5.1	Controlling spatially aware user interfaces .....	63
5.1.1	Device motion as an input device .....	64
5.1.2	Device pose as an input device .....	65
5.2	Recognising device movements .....	66
5.2.1	Handwriting recognition .....	67
5.2.2	Sign recognition .....	68
5.2.3	Finger gesture recognition .....	68
5.3	Interactive system for document image mosaicing .....	70
5.4	Practical considerations .....	72
5.4.1	Usability .....	72
5.4.2	Platform support for the new techniques .....	73
5.5	Discussion .....	74
5.6	Summary .....	75
<b>6</b>	<b>Discussion</b>	<b>77</b>
<b>7</b>	<b>Conclusions</b>	<b>79</b>

<b>References</b>	<b>80</b>
<b>Original articles</b>	<b>88</b>



# 1 Introduction

"Man-computer symbiosis is an expected development in cooperative interaction between men and electronic computers. It will involve very close coupling between the human and the electronic members of the partnership. The main aims are 1) to let computers facilitate formulative thinking as they now facilitate the solution of formulated problems, and 2) to enable men and computers to cooperate in making decisions and controlling complex situations without inflexible dependence on predetermined programs."

The quotation above is from J. C. R. Licklider's seminal paper "Man-Computer Symbiosis" (Licklider 1960). Almost 50 years ago, Licklider opened a window to a future for communication between users and computers, the study of what we now call human-computer interaction (HCI).

The aim of HCI is to improve the interactions occurring at the user interface by making computers more usable for people. In 1960, when Licklider was writing, the users entered commands to a computer through switches and punched cards. Once the computer processed the input directed by the user, it responded via lights and line printers. Although computers were hard to use at that time, he argued: "In a few years, men will be able to communicate more effectively through a machine than face to face" and even outlined input and output (I/O) devices using multiple modalities including speech communication, writing surfaces and large wall displays (Licklider 1960). We now know that technology evolved much more slowly than he predicted.

Since the early days of computing, a few major styles of communication have dominated in HCI. In the early 1960s, command-line interfaces became commonplace, replacing the first rudimentary systems. In this straightforward model, the user types commands using an electronic keyboard and gets text output on the monitor. After this popular technique used in the UNIX and DOS operating systems, the graphical user interface (GUI) and its associated desktop metaphor, Windows, Icons, Menus, and Pointing (WIMP), emerged in the 1970s and 80s. Most personal computers (PCs) in use today provide an interface for their users that still employs this paradigm. WIMP-based GUIs allow productive and user-friendly direct control of the computer by means of a pointing device such as a mouse with a keyboard and a monitor (Myers 1998).

However, there are no longer just PCs used for word processing and spreadsheet ma-

nipulation at our desk. Computing is becoming something that is a significant part of our daily life and the interactions are often briefer, more episodic in their nature (Turk 2005). Today, a mobile phone is a common piece of equipment for almost everyone and the number of users is more than double the number of PCs in the world. Unfortunately, interaction designs of many current devices have been taken directly from desktop computers where a limited screen space and other resource limitations are not a problem. This has resulted in devices that are sometimes hard to use.

As more and more features and functions are crammed into hand-held devices, their limited keypads and small displays are becoming overloaded, potentially confusing the user who needs to learn to use each individual application. Based on the personal experiences of most people, increasing the number of buttons is not the best solution from the usability point of view. The full keyboard, mouse and higher resolution displays of PCs appear to give them clear benefits as computing platforms. However, the small size of mobile devices is an under-exploited asset, as are their multiple sensors. Properly combined these characteristics enable novel user interfaces that are ideal for hand-helds, but may not make sense with PCs.

## **1.1 Computer vision based interfaces**

Since more processing power, new sensing and display technologies are already available in mobile devices, there has been increased interest in building systems to communicate via different modalities such as speech, gesture, expression, touch, etc (Jaimes & Sebe 2007). In perceptual user interfaces, these independent modalities are combined to create more powerful interaction techniques. While these are unlikely to completely replace traditional interfaces, they will enrich and improve the user experience and task performance.

In particular, it has been shown that different sensors provide viable alternatives to conventional interaction in portable devices. For example, tilting interfaces can be implemented with gyroscopes (Rekimoto 1996) and accelerometers (Hinckley *et al.* 2000). Using both tilt and buttons, the device itself is used as input for navigating menus and maps. During the operation, only one hand is required for manipulation. Several devices employ a detachable stylus in which interaction is done by tapping the touch screen to activate buttons or menu choices. Interestingly, Apple's products make use of the same technology in a different way. In the iPhone, users are allowed to zoom in and out by performing multiple fingers gestures on the touch screen. In addition, a

proximity sensor shuts off the display in certain situations to save battery power, and an accelerometer senses the orientation of the phone and changes the screen accordingly.

On the other hand, many of the current devices have also two cameras built-in, one for capturing high resolution photography, and the other for lower resolution video telephony as shown in Fig. 1. Even the most recent devices, have not yet utilised these unique input capabilities enabled by cameras for purposes other than just photographing. With appropriate computer vision methods, information provided by images allow us to create new self intuitive user interface concepts for future challenges in the fast developing field of mobile computing.



**Fig 1. Typical mobile communication device with two cameras.**

Traditionally, vision has been utilised in perceptual interfaces to build systems that look at people and automatically sense and perceive the human users, including their location, identity, focus of attention, facial expression, posture, gestures and movements (Pentland 2000). A key advantage of using vision as an input modality is that the interaction is entirely passive and non-intrusive, as it does not require contact with the user or any special-purpose accessories (Turk 2004). However, vision is one of several possible sources of information that can be coupled with other sensors such as accelerometers enabling a more effective and efficient interaction.

A camera called EyeToy, a peripheral for Sony's PlayStation 2 game console has proven that computer vision technology has become feasible for consumer-grade applications. This device allows players to interact with games using simple motion estimation, colour detection and also sound, through its built-in microphone. Despite the significant progress made in technology over recent years, many challenges still remain. Vision based interfaces are typically limited in their interaction capabilities, often requiring customised hardware and they work only in more or less restricted environments (Freeman *et al.* 2000).

For each technology, of course, there are conditions in which it is favourable to use alternative techniques. For example, speech recognition fails in noisy environments and it is cumbersome to use touch screens with a pen or fingers on small screen devices. Turk (2004) lists questions that must be addressed when vision is applied for user interfaces. In the following, the most important of these issues are presented in the context of mobile interaction. Firstly, small changes in lighting or camera position cause most vision systems to fail. Especially in mobile use, techniques need to be robust and work on a wider variety of continually changing conditions. Secondly, the response time must be quick enough to be interactive. That is, we need faster, more precise and accurate algorithms and their implementations on both software and hardware. Finally, algorithms must adapt to the users, that is, they work for different people and work against unpredictable behaviour. In this thesis, the aim is to develop new computer vision based solutions to meet these requirements.

The comprehensive implementation of new interaction solutions calls for studies from many branches of the HCI. However, user interface design fields such as usability engineering or user experience design are not the focus of this thesis. Instead, this thesis addresses how vision can enable new, self intuitive user interface concepts and applications on mobile devices.

## **1.2 The contribution of the thesis**

The main contributions of this thesis are the development of computer vision approaches to allow efficient interaction with mobile devices. The proposed techniques include facial feature based head tracking, object motion analysis, and device ego-motion estimation. The key ideas of the methods presented rest on the utilization of the hand-held nature of the equipment. The new techniques developed are demonstrated successfully in three new mobile interface solutions.

For head tracking, two new methods are developed. The first method detects a face region and facial features employing skin detection, grey-scale morphology, and a geometrical face model. A Kalman filtering (KF) framework is then applied for tracking and for estimation of the 3-D pose of the head. The second method, designed especially for mobile use, detects the face and the eyes using local texture features and boosting. In this case, extended Kalman filtering (EKF) combines motion features extracted from the face region and eye positions to estimate the 3-D pose of the camera with respect to the face. Experiments with real image sequences, and the real-time performance achieved indicates that the proposed methods can be applied on platforms with limited computational resources.

A new object tracking approach for user interaction with hand-held devices is also presented. The idea is to combine Kalman filtering and expectation maximisation (EM) algorithms to track an object, such as an finger, using motion features. This technique is applicable also when some conventional techniques such as colour segmentation and background subtraction cannot be used. A new camera ego-motion estimation framework is also proposed. The method introduced exploits gradient measures for feature selection and feature displacement uncertainty analysis. In addition, many challenges and trade-offs are highlighted that one must face in the domain of low power mobile devices. Experiments with a fixed point implementation testify the effectiveness of the approach on a camera-equipped mobile phone.

The feasibility of the methods developed is demonstrated in three new mobile interface solutions. One of them estimates the ego-motion of the device with respect to the user's face, and utilises that information for browsing large documents or bitmaps on small displays. The second solution is to use device or finger motion to recognize simple gestures. To recognise device movements, a method using discrete cosine transform (DCT) to compute discriminating features from the motion trajectories and the use of  $k$ -nearest neighbour rule ( $k$ -NN) in classification is first proposed. In the other technique, the motion feature sequences are classified using Hidden Markov models (HMMs). Furthermore, a technique for user adaptation is introduced. In addition to these applications, a new interactive system to build document panorama images is presented. The system interactively guides the user to move the device over a large document page in such a manner that a high quality image can be assembled from individual frames.

### 1.3 Summary of original papers

This thesis consists of nine publications. Paper I presents a fast and efficient head tracking approach which automatically detects facial features and estimates the head pose using Kalman filtering. The method is applied successfully for the display control on a desktop computer.

Paper II introduces a new face tracking approach for the control of mobile user interfaces. The method employs local texture features to detect the face and the eyes of the user. An EKF combines local motion features extracted from the face region and the detected eye positions to estimate and track 3-D pose of the camera with respect to the user's face. In the experiments, the use of the camera position as input for spatially aware display is demonstrated.

Paper III proposes a new camera ego-motion estimation framework for mobile devices. In this paper, gradient measures are exploited to select a sparse set of features and analyse the uncertainty of the feature displacements measured. Global motion of the device is estimated after a voting based outlier removal process. Experiments with a fixed point implementation demonstrate the effectiveness of the technique on a mobile phone with a built-in camera.

In Paper IV, the motion estimation framework presented in Paper III is further developed. It describes the improved feature selection principle in order to provide more discriminative features. Moreover, it recommends the use of a ZSSD matching measure when lighting conditions change continually. The performance of the technique is evaluated in experiments with scene and document image sequences.

Paper V presents a new algorithm to track multiple motions. The technique combines the Kalman filter and the EM algorithms to estimate distinct motions using local motion features extracted from the scene. The method is applied to finger tracking on mobile interaction when the camera is also moving.

Paper VI describes a new interaction technique where a camera-enabled mobile device is used for writing characters and signs just by moving the device in the hand. The method computes DCT features from the motion trajectories obtained and then the  $k$ -NN rule is applied to classification. In experiments, good recognition rates were achieved demonstrating the feasibility of the method.

Paper VII introduces a new user interface solution for mobile phones where the user makes special signs through a series of hand movements. The method models and interprets the motion trajectories with HMMs. Recognition rates are further improved

when the classification results are filtered using a log-likelihood ratio and the velocity entropy to reject ambiguous sequences.

Paper VIII presents another new method for the control of mobile devices where the user operates the interface by simply moving a finger in front of a camera. These finger gestures produced are again modelled using HMMs. The recognition rate for a specific user is enhanced by utilising unsupervised adaptation.

Paper IX demonstrates a new user interface concept for document image scanning on mobile phones. The proposed system applies online camera motion estimation to the phone to assist the user in scanning the document. The full view of the document is then reconstructed automatically with the help of estimated device motion. Experiments on real document images captured and processed indicate the viability of the technique.



## 2 Face and facial feature detection

Automatic face detection is the first major step in many applications such as multimodal human-computer interaction (Jaimes & Sebe 2007), face recognition (Zhao *et al.* 2003) and facial expression analysis (Pantic & Rothkrantz 2000). It is a process of finding the position and size of one or more faces in an image and subsequently segmenting it from the rest of the image. For many applications, the current face detection systems are sufficiently mature to be used. Once the face is detected, also the facial features such as eyes, mouth etc. can be searched for to estimate, for example the face pose. Although it is very easy for humans to locate facial features, reliable computer vision based facial feature extraction in complex scenes and continually changing environmental conditions still remains a challenge. The human face and facial features may change its appearance due to facial expressions and other factors such as a beard, hairstyle, glasses etc. In addition to these internal factors, changes in the lighting, position and orientation of the face must be tolerated.

In this chapter, we introduce face and facial feature detection techniques, which can be utilised to implement an automatic head tracking system for HCI. First, Section 2.1 presents a review of the previous work on face and facial feature detection. Then, Sections 2.2 and 2.3 propose new methods for performing a facial feature extraction task on devices with limited computational resources. Finally, Section 2.4 discusses the methods introduced in this chapter.

### 2.1 Previous approaches

A number of face detection methods have been proposed in the literature over the last few decades (Hjelmås & Low 2001, Yang *et al.* 2002). Yang *et al.* (2002) have a fine-grained categorisation in which face detection methods are divided into knowledge-based, feature invariant, template matching, and appearance-based methods. However, a problem with this grouping is that some methods can be classified into more than one category (Yang *et al.* 2002). Another common and simple way of classifying methods has been proposed by Hjelmås & Low (2001). According to them, face and facial feature detection methods are divided into two categories. The first one, called the feature based approach, extracts some important facial features as a first step, after which the

face is detected by utilising information about the feature locations and the knowledge of their relationship. In the second category, called the image based approach, the face is detected as a whole unit and the facial features are then localised by considering the facial geometry.

### **2.1.1 Feature based methods**

Feature based methods typically employ some segmentation method as a pre-processing step to detect faces and facial features. An advantage of the segmentation process is that the search area for the face is smaller, and therefore computational requirements are significantly decreased.

Several approaches first use skin colour analysis to extract face-like regions. A typical solution is to verify the shape of the detected regions with an ellipse or to use heuristic knowledge of the human head (Sobottka & Pitas 1998, Yang & Ahuja 1998). Although processing of colour is faster than any other features, it is still a challenging task to segment the skin colour in continuously changing lighting conditions. Motion is another cue which can be used to segment faces from the images. For example, frame differencing (Graf *et al.* 1996) and optical flow (Lee *et al.* 1996) have been used for this purpose. Bala *et al.* (1997) also applied motion information to extract the position of the eyes. In general, colour and motion information are not sufficient to detect faces, and therefore the analysis usually continues with a low-level feature extraction step.

Most of the low-level approaches at first locate features coarsely by searching areas of low intensity within possible face regions. These methods include basic computer vision techniques such as edge detection (Yow & Cipolla 1997), morphology (Wong *et al.* 2003) and projection analysis (Brunelli & Poggio 1993). The morphological operators and projections are especially suitable for real-time applications due to their simple and fast implementation. A drawback of these methods is that the grey-scale data information is easily influenced by a change of illumination conditions and noise. For example, projection curves are therefore not always very smooth, which makes them difficult to analyse.

Another approach to extracting low-level features is to take advantage of properties such as the symmetry of the face. For example, Reifeld & Yeshurun (1992) developed a generalized symmetry interest operator to detect the eyes and the mouth. Loy & Zelinsky (2003) presented a fast radial symmetry transform to detect facial features.

Yet another way to extract facial features is to use Gabor filters to locate the corners of the eyes and the iris (Herpers *et al.* 1996, Sirohey & Rosenfeld 2001).

Pre-processing methods produce many candidates for facial features and some knowledge of the face is needed. For example, a simple geometrical face model based on anthropometry of the human face can be generated to detect facial features of interest. Hjelmås & Low (2001) divide the methods performing this kind of feature analysis into feature searching (Jeng *et al.* 1998) and constellation analysis techniques (Yow & Cipolla 1997).

If more accuracy is needed, the facial feature positions are extracted using more advanced models depicting the actual physical appearance of features. These methods, including active shape models such as active contours, called snakes (Kass *et al.* 1988), deformable templates (Yuille *et al.* 1992), and point distribution models (Cootes & Taylor 1992), are applied directly to the facial image or coarsely detected regions in the facial image. The latter is a more common approach. However, the use of advanced methods typically means higher computational complexity than extracting only the locations of the features.

### **2.1.2 Image based methods**

Image based methods treat face detection as a two class pattern recognition problem. These approaches search for faces by classifying all possible sub-images of a given image as a face or non-face pattern. Due to the diversity of the faces, the classifier is trained on a large number of samples of face and non-face images to discriminate between these two classes.

Sung & Poggio (1998) presented an example based learning approach for locating vertical and frontal faces. They model the distribution of face and non-face patterns using Gaussian clusters and the applied Multilayer Perceptron for detection. A similar approach was adopted and extended by Rowley *et al.* (1998).

Schneiderman & Kanade (2000) also model the probability distribution of the face class, but they employ a naive Bayes classifier to estimate the joint probability of the local appearances and positions of the sub-images of the face at multiple resolutions. Later, Schneiderman (2004) described an algorithm that searches for the structure of a Bayesian network based classifier. The method was used to automatically train detectors of frontal faces, eyes, and the iris of the human eye. The performance of a Bayesian

network was reported to be superior to semi-naive Bayes and other state-of-the-art algorithms.

Viola & Jones (2001) introduced a rapid object detection scheme based on a boosted cascade of simple Haar-like features. They apply AdaBoost (Freund & Schapire 1995) to find the most discriminative features for distinguishing the face and facial patterns from the background. The AdaBoost algorithm is a discriminative learning method that has been widely used in different object detection tasks. The idea is to combine several relatively weak classifiers into a strong cascade of classifiers. Overall, the accuracy and speed of the Viola & Jones (2001) method has made it the most popular and well-known method for face detection.

Later, Lienhart & Maydt (2002) extended the original feature set proposed by Viola & Jones (2001), introducing a set of rotated Haar-like features. These rotated rectangular areas can also be summed up quite efficiently; however, now two passes over the image are required instead of one for the original features. Zhang *et al.* (2002) presented the real-time multiview face detection system following the work of Schneiderman & Kanade (2000) and that of Viola & Jones (2001). They also used an extended set of Haar-like features and introduced a new learning algorithm called FloatBoost. Later, also Jones & Viola (2003) extended their work to multi-view face detection. (Huang *et al.* 2005) developed a boosting algorithm called Vector Boosting. They claimed that the method achieves significant improvements in both speed and accuracy compared to previously published methods.

Another technique for multiview face and eye detection was introduced by Wang & Ji (2007). In their approach, the idea is to use a statistical learning method to extract discriminant features. The experiments performed indicate improved performance over existing methods. However, computing discriminant features is more time-consuming than, for example, Haar-features. Yet another approach to multi-view face detection was proposed by Osadchy *et al.* (2007). Their method employs a convolutional network to simultaneously detect faces and estimate the face pose. They reported comparable results to the previous multi-view detectors.

## **2.2 Searching facial features via colour and morphology**

This section proposes a feature based method, originally presented in Paper I, to search for facial features. Facial features are extracted in three consecutive steps. First, the face regions are found by skin colour analysis and the shapes of the regions found are

verified. In the second step, the facial feature candidates are searched for within those regions. Finally, the candidates are evaluated to find facial features.

### **2.2.1 Detecting face regions using skin colour**

We detect possible face regions in an input image utilising the skin detection method by Martinkauppi *et al.* (2003). They have found Normalised Colour Coordinates (NCC) combined with the skin locus model most appropriate for skin detection under varying illumination. In NCC colour space, we only use two chromaticities  $r$  and  $b$  for detection in a histogram where the skin colour occupies a small cluster. The cluster is found by training with facial images in different illuminations. If  $r$  and  $b$  of a pixel fall into the area of the skin locus, the pixel belongs to skin. In practise detection is implemented using a look-up table. The result is enhanced using a morphological binary closing, and then the shape of the regions are verified by performing connected component analysis.

### **2.2.2 Searching facial features using morphology**

Once a possible face is detected, the features are searched for within the facial region. Based on the observation that facial features such as eyes and mouth are usually darker than their surroundings, we use a morphological valley detector called black top-hat (Gonzalez & Woods 2002). At first, a closing operation is performed and then the original image is subtracted from the result. The valley image  $V$  is defined as

$$V(x,y) = f(x,y) * \mathbf{B} - f(x,y), \quad (1)$$

where  $f(x,y)$  is the image intensity at a point  $(x,y)$  and  $\mathbf{B}$  is a structuring element with the size of 3 by 3 pixels.

A pixel at  $(x,y)$  belongs to a feature candidate if the pixel's grey-level is dark enough, and the response from valley detection is high enough, that is, if the following equations are true

$$f(x,y) < T_L \quad \text{and} \quad V(x,y) > T_V, \quad (2)$$

where  $T_L$  and  $T_V$  are pre-defined thresholds. The detection result can be noisy and there exist some isolated pixels. Therefore we improve the result by utilizing morphological binary closing with a mask size of 3 by 3 pixels. Then, connected component analysis is performed and centres of mass of the objects are used as feature coordinates.

### 2.2.3 Face and facial feature analysis

After the possible facial features are detected, we evaluate feature constellations using a geometrical face model including eyes, eyebrows, nostrils and mouth. This kind of model was originally proposed by Jeng *et al.* (1998). The line passing through the centres of the eyes is called the base line. The  $d_{mouth}$ ,  $d_{nostril}$  and  $d_{eyebrow}$  indicate the distances from other features to the base line.

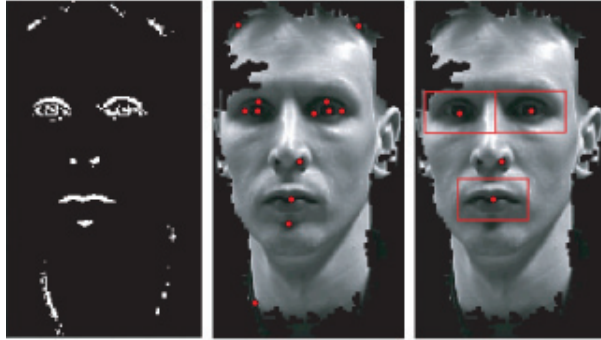
Two features form a possible eye pair if they are located in the upper half, and they meet some general geometrical conditions. Based on the distance between eyes  $D$ , other reference distances  $D_{nostril} = 0.7D$ ,  $D_{eyebrow} = 0.35D$  and  $D_{mouth} = 1.1D$  are experimentally derived from measurements with several real faces. For each candidate eye pair, the other facial features are searched for. The possible mouth for an eye pair is searched from a region that is located within the distance of  $1.1D$  from the base line. If that region contains a feature, it is treated as a mouth for this possible eye pair. In order to rank many candidates, a special evaluation function is proposed for each facial feature as follows

$$E_{feature} = \exp(-10(\frac{d_{feature} - D_{feature}}{D})^2), \quad (3)$$

where  $feature = \{mouth, nostril, eyebrow\}$ . The evaluation function for the eye pair is

$$E_{eyepair} = \exp(-10(\frac{D - 0.4BB_{width}}{D})^2), \quad (4)$$

where  $BB_{width}$  is the width of the face bounding box. The total evaluation value is a weighted sum of the values for each facial feature. The weights for the eye pair, mouth, nostrils and eyebrows are 0.4, 0.3, 0.1 and 0.05, respectively. The weight is based on the importance of a given feature for face detection. The eyes are considered the most important features. The constellation which has the largest evaluation value, is assumed to be a face. Fig. 2 shows an example of facial feature extraction where results for valley detection are on the left, candidates in the middle and the best constellation on the right.



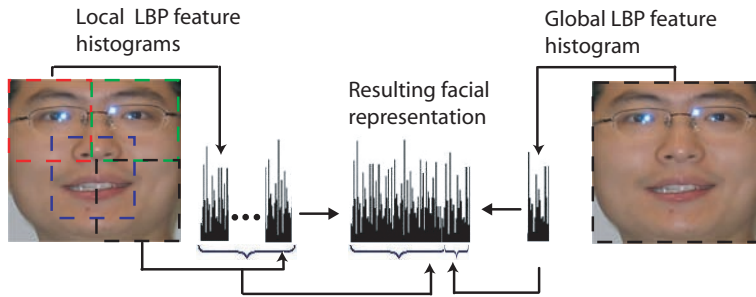
**Fig 2. Facial feature extraction: the result of valley detection is on the left, feature candidates are in the middle and the best constellation found is on the right.**

## **2.3 Detecting face and facial features using texture**

In this section, we introduce a new image-based face and facial feature detection method. This technique has originally been described in Paper II. Our approach uses efficient grey-scale invariant texture features called the local binary pattern (LBP) methodology (Ojala *et al.* 2002) and AdaBoost learning (Freund & Schapire 1995).

### **2.3.1 LBP features**

The LBP features can detect local texture primitives such as spots, edges or corners from the images (Ojala *et al.* 2002). They have been found to be very discriminative in facial image analysis (Hadid *et al.* 2004). Similar to the approach of Hadid *et al.* (2004), we use the facial representation where the LBP feature histograms are separately computed over the sparse set of local image regions and the whole face area. These are then concatenated for creating the final face descriptor. Fig. 3 illustrates the idea of LBP based facial representation. The original 20 by 20 pixels face image is divided maximally into nine overlapping regions of 10 by 10 pixels. The basic 4-neighbours LBP operator with a 16-bin histogram is used resulting in a total of a 144-bin histogram of local features. In addition to that, the global 16-bin histogram is concatenated to the local features in order to create the full 160-bin face histogram.



**Fig 3. Facial representation with LBP texture features: a face image is with set of local and global LBP histograms. Revised from [II]. © 2008 Springer.**

### **2.3.2 Classifier**

In our case, weak features used in classification are the LBP histogram bin values calculated over certain image regions. These positions and corresponding LBP values are learned off-line with the set of labelled face samples. Inspired by the well-known Viola & Jones (2001) approach, we built a cascaded classifier structure to speed up the detection. On the early cascade levels, only a few histogram positions were considered to rapidly reject the majority of classified image regions. The face like image regions were classified with more features in order to make robust detection. We applied the trained cascade to an image pyramid using a sliding window approach to classify image regions in different scales. As an result, the rectangular coordinates of each detected face were obtained.

Once a possible face of the user is detected, the eyes are searched for within the facial region. The approach for detecting the eyes is similar to the face detection, except the size of the template image is 16 by 20 pixels to more accurately cover the eye region. Face orientation and size were used to restrict the search window and scales in the eye detection phase. In this sense, the actual detection is carried out with reduced computational costs.

## 2.4 Discussion

A lot of research has been done in the field of facial image processing over a long time period. Researchers have shown good results on frontal face detection and detection algorithms have recently achieved a quality level acceptable to consumers for a number of applications. For example, many digital cameras are now included with a mode, where the camera detects faces in a scene and then automatically focuses. However, mobile devices still have several limitations especially compared to dedicated digital cameras. These devices have limited computational power and memory available and they must run several applications in parallel which decreases available resources for the used algorithms. Furthermore, considering current methods there are still needs to improve robustness, speed, and accuracy, especially in dark lighting conditions. Even then, only few researchers have investigated the problem of face detection on mobile phones (Venkataramani *et al.* 2005) so far.

This thesis presents efforts in bringing real-time face and facial feature detection to mobile devices to be used in human-computer interaction. In Paper I, the goal was to extract facial features, which can be used to track and estimate the face pose. Based on extensive literature survey, we present a feature based method that combines the advantages of the different algorithms presented earlier in the literature. First, we implemented a fast skin detection scheme based on the skin locus method (Martinkauppi *et al.* 2003) to find face like regions from the image. This was found to be very suitable for real-time use. Then, we employed morphological valley detection for extracting facial features roughly from the facial area. The original idea of the approach was presented by Wong *et al.* (2003). In our work, we fine-tuned valley image computation and improved the post processing of the obtained result image. It was also discovered that the proposed valley detection is highly usable in tracking facial features. Finally, the geometrical face model used is based on the work presented by Jeng *et al.* (1998). The original model does not match very well for Caucasian people. Therefore, we have modified their face model parameters and additionally included two nostrils instead of the nose. We also modified the search procedure to be more appropriate for our purposes. We noticed that using the proposed combination of the improved methods it is possible to automatically and reliably detect facial features from images with lower computational requirements than the methods presented earlier. Although the developed method was tested on a desktop computer, we are confident that it can be implemented on the mobile phone platform and the method clearly fulfils the real-time

demands. The main drawback of the proposed method lies in the use of skin detection because a change in illumination conditions can deteriorate detection results. Another shortcoming is that a near frontal face orientation is needed.

Currently, image based methods are more popular than feature based methods since they can achieve good detection accuracy while still retaining adequate computational efficiency. Most often used face detection methods are based on the approach using Haar features and Adaboost learning (Viola & Jones 2001). In paper II, we present an image based method for face and eye detection, employing LBP features instead of using Haar-like features. The idea of using LBP features is not new, and they are shown to perform well in face detection and recognition earlier (Hadid *et al.* 2004). In contrast to Haar-like features, LBP features have also the advantage of low computational cost and they have more discriminative power. In our work, we extended the use of LBP features also to eye detection and made an efficient implementation on mobile devices. The maximum frame rate of 30 fps was achieved using a Nokia N95 mobile phone. The proposed image based method is more robust to illumination changes than the proposed feature based method. Also, the method can handle more variations in the face orientation. The shortcoming is that the method does not provide so accurate feature coordinates and the computational needs are slightly higher.

It is difficult to make a fair comparison between different existing methods for facial feature extraction in the literature. The properties like image sizes, illumination, image quality, feature size, vary from one study to another. It is also hard to compare the computational load of algorithms due to different computer platforms and implementation issues. The main benefit of the methods developed is speed, and in order to fairly compare different algorithms they must be implemented on the same platform. Furthermore, our main goal in the experiments was to make a working system to mobile user interaction purposes. Therefore, we have not focused on fine-tuning face and facial feature detection algorithms. In the future work, the image and feature based methods could be combined. For example, the face and eyes can be first roughly located using an image based approach and then a feature based method can be applied to obtain more accurate feature coordinates. If the facial features should be tracked, the extraction is more difficult due to continuous changes of appearance. In this case, deformable templates can be useful because they also recover the shape of the feature.

## 2.5 Summary

In this chapter, we reviewed approaches for face and facial feature detection. The approaches were divided into feature based and image based methods. In this thesis, we have proposed two new solutions: one feature based method and one image based method. The new feature based method utilises skin detection, morphology and a geometrical face model to find facial features from the image. In the proposed image based method, LBP texture features with AdaBoost learning are used to find the face and facial features.

The advantage of the feature based method is fast and simple implementation suitable for devices with limited resources. However, especially the skin detection is very susceptible to lighting changes and not directly applicable in a continuously changing mobile environment. Furthermore, it is difficult to obtain a geometrical model for the face, which applies to all users due to variability in facial feature distances between individuals. On the other hand, the image based method does not need a geometrical model and it can be also implemented with low computational cost and good detection performance under varying illumination conditions. However, it needs an extensive set of training data and it is difficult to obtain accurate coordinates for facial features such as eyes, while feature based approaches make it possible to extract more accurate positions and representations such as contours for facial features. Therefore, we can easily say that both approaches have pros and cons when considering practical mobile applications.



## 3 Head pose estimation and tracking

This chapter discusses the development and implementation of a real-time head tracker. Head tracking and gaze direction detection provides an important input modality for perceptual user interfaces. The estimated position and orientation (pose) of the head allows recognition of simple gestures such as nodding and head shaking. The detected human gaze direction also determines which part of the screen the user is looking at. Continuously locating the pose of the user's face can be used to control spatially aware user interfaces and may be also necessary or at least helpful for recognising the user's facial expressions.

In this thesis, head pose estimation is performed by first locating the face and features from the face region as described in Chapter 2, and then solving the pose using feature correspondences between successive images. However, the measured 2-D coordinates of the facial features are usually distorted due to various error sources in the image formation process. These errors may produce large pose estimation uncertainties. Visual tracking of the facial features over a longer period of time with a proper estimation technique can improve the pose estimation result significantly. First, in Sections 3.1 and 3.2 the problems of visual tracking and head pose estimation are addressed. Section 3.3 reviews the related work on head tracking. Then, new methods to track and estimate the head pose are presented in Sections 3.4.1 and 3.4.2, respectively. Finally, Section 3.5 discusses the methods introduced in this chapter.

### 3.1 Visual tracking

The basic idea behind visual tracking is to determine the measurable state of a target object from a sequence of images (Toyama & Hager 1999). The complexity of the tracking problem depends on the application. For example, the position of a face in image coordinates can be sufficient and in other cases, movement of different parts of the face need to be tracked. Visual tracking can be understood as a process of repeated estimation and search steps. It should be noted that the estimation is usually performed in the state-space, not in the image space. Typically, the state variables are related to the structural model of the object.

Human motion, including face or head motion, can be rigid, articulated, or de-

formable (Black *et al.* 1997). Rigid motion means that there is no relative motion between different parts of the object with respect to each other. In the case of articulated motion, the structure of the object can be characterised as comprising rigid components connected by simple constraints such as hinges, slides etc. In deformable motion the relative position of any points of the object can change, which makes object tracking a much more difficult problem. In this thesis, we treat the head as a rigid object. In the next section, we focus on head pose estimation and tracking problems.

### **3.2 Head pose estimation and gaze direction detection**

The process of finding the position and orientation (pose) of an object is known as the pose estimation problem. The user's gaze direction can be obtained from the estimated pose and it is determined by two factors. The orientation of the head specifies the overall direction of the gaze and the orientation of the eyes determines the exact gaze direction. Obviously, the latter is limited by the head orientation. Here, we focus on estimating the orientation of the head, which is considered to be precise enough for our purposes.

3-D pose estimation from 2-D images can be solved in many ways. The classical approach is stereo vision, where two cameras with two different views are used to capture 2-D images of the given 3-D object. Another approach is to use only one stationary camera and capture a sequence of images of moving objects. Equivalently the camera can be moving and the object is stationary. In those cases, both pose and the 3-D structure of the object can be estimated. On the other hand, the 3-D pose of the object can be derived from a single image if some constraints on the object geometry are set. All of these methods need to establish correspondence pairs between the object and the images. In this study, we use a model based approach in which the correspondences are computed between facial features extracted and a model of the human head.

Visual tracking of the head throughout an image sequence will improve the pose estimation accuracy if the information from the previous frames is utilised. The tracking task can be formulated as a linear or non-linear estimation problem depending on the camera model used. This formulation allows the use of different state estimation techniques, such as Kalman filtering (KF) (Kalman 1960), extended Kalman filtering (EKF) (Mendel 1995) and Particle filtering (Isard & Blake 1996).

Kalman filtering is probably the most common algorithm for implementing the tracker, although Particle filtering and some others have been shown to provide certain advantages especially in the presence of significant background clutter. For example,

particle filters allow for non-Gaussian probability functions. Despite their methodological improvements, many of the solutions such as particle filtering are computationally expensive if compared to Kalman filtering. Other popular methods for tracking include mean shift algorithm (Comaniciu *et al.* 2000) and optical flow based methods such as the Kanade-Lucas-Tomasi (KLT) tracker proposed by Tomasi & Kanade (1991).

### **3.3 Related work**

In this section, some representative work in the area of visual tracking of the human head and simultaneous estimation of the head pose is presented. According to Black *et al.* (1997) tracking can be a 3-D model based or a 2-D image based. The 3-D model-based trackers assume a 3-D model of the object and estimate the pose based on feature correspondences. The 2-D image or appearance-based methods use intensity or colour information on the images to track the object parts. Mapping 2-D images of the object and corresponding 3-D poses is based on learning from appearance examples. Usually the 2-D image based tracking performance is decreased by the articulation and deformation.

#### **3.3.1 2-D image based methods**

Methods using 2-D image information typically utilise visual cues from the entire head and can be region based, colour based or shape based. In practise, methods can usually combine many cues to accomplish tracking task.

Region based trackers typically use information such as colour to segment the object of interest from the image. There are two groups of region-based methods: view-based and parametric. The view-based methods find the best match for a region in a search area with a reference template. The parametric methods assume a parametric model of changes in the image and compute an optimal fitting of the model to the pixel data in a region.

Many solutions exploit colour information to accomplish the tracking task. One very popular solution is the continuously adaptive mean shift (camshift) algorithm presented by Bradski (1998). It is based on the mean shift algorithm (Comaniciu *et al.* 2000), which is a robust non-parametric iterative technique for finding the mode of probability distributions. Camshift detects the mode in the probability distribution image by applying mean shift while dynamically adjusting the parameters of the target

distribution. In a single image, the process is iterated until convergence. A method can be applied to successive frames to track a face. The search region can be restricted around the last known location of the face, resulting in remarkable computational savings.

Also, Birchfield (1999) proposed a tracker that simultaneously exploits the elliptical contour fitted with the help of colour information. This method can handle occlusions and out-of-plane rotations, but requires manual initialisation.

The face tracking system presented by Yang *et al.* (2006) consists of two interactive modules. First, the detection module uses a method similar to Viola & Jones (2001) for detecting faces. Then, tracking is performed by a dominant colour feature selection method based on mean shift analysis.

### **3.3.2 3-D model based methods**

In the following discussion, model based approaches are further divided into methods, those that compute optical flow and those that use facial features to estimate the head pose.

#### **Methods using optical flow**

Black & Yacoob (1995) developed a regularized optical flow method that uses an eight parameter 2-D planar model. In their work, patches are attached to the facial features and movements of different facial parts are followed. However, the use of the 2-D planar model limits accurate tracking to medium head motions. Large head motion causes problems.

Inspired by their work, Basu *et al.* (1996) presented a system to track heads with a large amount of head motion. Instead of using a planar model, they used a 3-D ellipsoidal model of the head and coupled it with general optical flow computation. The algorithm starts by computing optical flow for the image and then it estimates the 3-D motion of the rigid model. The estimated motion parameters are used to modify the location and rotation of the model for the next frame. The motion estimation for each frame depends on the accuracy of the estimation in the previous frame and therefore the errors tend to propagate and grow. The system was not real-time due to slow computation of the optical flow.

DeCarlo & Metaxas (1996) introduced a system using a polygonal head model that

is manually positioned on the user's face. Optical flow is extracted at some feature points and it is regularised by the model movements. Finally, the measurements are stabilised utilising a Kalman filter. A drawback is that using optical flow leads to error accumulation. Therefore, additional face edge information is utilised to prevent divergence.

Cascia *et al.* (2000) presented an approach using a texture mapped 3-D rigid surface model for the head and formulated tracking as an image registration problem. The head is modelled as a cylinder and dynamic texture is used for tracking. The lack of fixed features in the face region again leads to error accumulation although confidence maps are used to minimise the problem.

Similar to their work, also Xiao *et al.* (2003) utilised a cylindrical model and presented a method to recover full motion of the head under perspective projection. They built a real-time head tracker, which was successfully used as part of a facial expression analysis system.

## **Methods using facial features**

Prior research using facial features includes the early work of Azarbayejani *et al.* (1993). They utilised an extended Kalman filter (EKF) to recursively estimate a head structure and motion from image sequences of rigid motion. In their system, distinct features corresponding to the corners of the eyes and nostrils are tracked and projected on an ellipsoidal head model. The use of the system is applicable if the same points are visible in most of the frames.

Another way of improving the performance of tracking, is to combine object contour matching and optical flow based methods (Brox *et al.* 2006). Their system uses contour matching, if the object silhouette contains enough information to estimate the pose and the object motion is small between successive frames. In addition to correspondences from the silhouette, they add matches from the optical flow computed and this helps to deal with larger movements.

Jebara & Pentland (1997) presented a similar feature tracking system for detecting, modelling and tracking human faces. They used an extended Kalman filter (EKF) to recover 3-D structure, pose, motion and focal length. The system automatically detects faces using skin colour and facial features, including eyes, nose and mouth based on a symmetry transform and image intensity gradient. In the tracking stage, features are measured using 2-D image correlation. However, the face and facial feature detection is

not performed in real time and if the feature is occluded due to a large amount of head movement the tracking fails.

Later, Ström *et al.* (1999) developed a head tracker based on the work of Jebara & Pentland (1997). The main change was in the feature points used in the tracking. The head tracker system tries to find the best features to track for each face instead of using predefined features such as the eyes, nose and the mouth. The input image Hessians are used to rank features. The system is expected to be more robust against large head movements. For example, if the eye is occluded a point from the ear is selected for tracking.

Ballard & Stockman (1995) proposed a system to estimate gaze direction. They assumed that distances between facial features are not changed with facial rotation and translation. The eyes and nose of the user are tracked using parameterised templates. Using these three points, the pose and gaze is computed with the perspective-three-point (P3P) algorithm. A disadvantage is that the distance between the camera plane and features has to be measured manually in the initialization phase. Also the gaze direction is not computed in real-time.

Yang *et al.* (1998) proposed a region based technique for face and facial feature tracking in real-time. They used a statistical skin colour model for facial region detection and face tracking. Then, the facial features such as eyes, nostrils and lips are extracted and tracked, and the head pose is estimated using a perspective camera model. Although, region based tracking can be robust, accurate results are difficult to achieve.

Vacchetti *et al.* (2004) formulated the tracking problem as one of local bundle adjustment in such a way that it can be solved very quickly. They exploit correspondences between incoming frames and a limited number of keyframes using a corner detector to extract features from the object area. Their approach requires a 3-D model of the target, which can be represented by a 3-D mesh.

Building a generic 3-D head model is a difficult task. To overcome this problem, Dornaika & Davoine (2006) used a deformable 3-D wireframe face model. They developed a particle filter based framework for tracking the pose of the face. In experiments, they obtained accurate tracking even in the presence of significant facial expression variations, occlusions, and illumination changes. Later, Dornaika & Orozco (2007) extended and improved their tracker in order to obtain more accurate and stable head pose parameters.

## 3.4 Proposed new head tracking approaches

In this section, we propose two alternative solutions to track and estimate the pose of the head. The first approach uses only facial features including the eyes and mouth, while the second approach combines facial features and optical flow extracted from the face region. In both cases, the Kalman filtering framework is utilised to implement the tracking system.

Kalman filtering (Kalman 1960) is a widely applied technique for pose estimation and tracking in computer vision. The main features of the Kalman filter are modelling the random process under consideration using a state-space model and recursive processing of the noisy measurement data. A filter is optimal if the dynamic model is linear, the measurement model is linear, and the noise processes involved are Gaussian distributed. Furthermore, the recursive nature of the algorithm makes it convenient to use in real-time systems where the data can be integrated into the state estimate and there is no need to store previous measurements. It also allows the prediction future measurements using the current state estimate. In many computer vision problems, however, the measurement model is non-linear, and thus Kalman filtering cannot be used. To overcome this problem one solution is the extended Kalman filter (EKF) that linearises the measurement model around the current state estimate. The drawback of the linearisation is that the EKF is no longer an optimal estimator.

The Kalman filter algorithm estimates and tracks the pose recursively, repeating two stages: prediction and correction. At the first stage, the pose and locations of the features at the next time instant are predicted based on the previous pose estimate and the dynamical model. In the correction stage the predicted pose is adjusted by using the measurements. Next, the proposed approaches, originally presented in Papers I and II, are briefly introduced in Sections 3.4.1 and 3.4.2, respectively.

### 3.4.1 Head tracking using facial features

In the first method, we use Kalman filtering to estimate the 3-D pose of a moving head and to track facial features. In order to track the head, we propose a simple rigid head model including the mouth ( $\mathbf{P}_1$ ) and the eyes ( $\mathbf{P}_2$  and  $\mathbf{P}_3$ ). The centre of the head coordinate system is set to the centre of gravity of these three points. We model the head pose and motion using first order dynamics, which allows prediction and filtering

of the pose from a sequence of images. The head is treated as a rigid object that moves with constant velocity, and the motion is subject to random perturbations.

We use a perspective camera model to relate the object coordinates  $\mathbf{P}_i$  to image coordinates  $(u_i, v_i)$  of three facial features. The 3-D pose of the head is computed with these points using the perspective-three-point (P3P) algorithm (Fischler & Bolles 1981). However, the solution is not unique and there exist up to four possible solutions for the pose. In order to evaluate all possible poses, we propose a solution which solves the system in closed-form (Linnainmaa *et al.* 1988). We choose the pose where the z-axis of the head coordinate system intersects nearest to the centre of the display. Also in tracking, our idea is to evaluate all the possible solutions, and choose the one that is nearest to the previous pose to be the measurement.

In Paper I, we evaluated the system's ability to track the facial features with several real image sequences containing large head motions and pose changes. The tracking was successful with most of the sequences. During testing we met major problems only with the case where the sequence contains a person with glasses. Certain types of spectacle rims and bad illumination can cause the system to fail in tracking. We also evaluated the theoretical pose estimation accuracy with simulated data, due to the lack of ground truth for the head pose. The data of 100 generated frames contains head poses corresponding to a real situation where the user is trying to control the cursor on the display. Very good preliminary pose estimation results were achieved, although it should be remembered that the accuracy values obtained are more or less theoretical, which apply under certain constraints. It was also observed that the error is biggest in the depth direction. The obvious reason for this is that the depth information is more uncertain than lateral information due to the perspective projection. In Paper I, an application for user interface control is also presented. This is reviewed in Chapter 5.

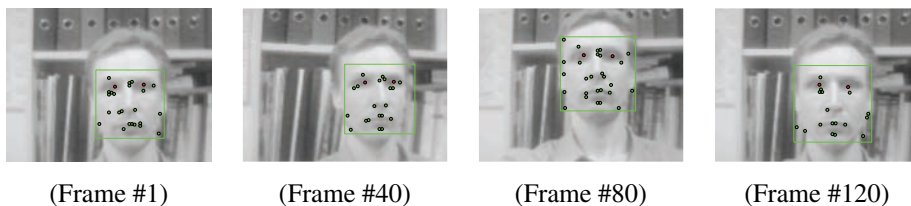
### **3.4.2 Head tracking using facial features and optical flow**

In the second method, we apply extended Kalman filtering (EKF) to estimate the camera position with respect to the head, and to track the eyes as well as the motion features. The motion features are obtained via local motion analysis, and they encode information about displacements of a sparse set of image blocks between two frames. The local motion analysis technique is described in Chapter 4. The face is modelled as a rigid plane which is not an accurate description, but considering our application it provides reasonable pose estimates to control the user interface.

Initially the face plane is assumed to be parallel to the image plane and its distance to the camera is some predefined constant. The face model includes the eye positions ( $\mathbf{P}_1$  and  $\mathbf{P}_2$ ) and the motion feature positions ( $\mathbf{P}_i, i = 3, 4, \dots, N + 2$ ).  $\mathbf{P}_1$  and  $\mathbf{P}_2$  are the backprojected face plane coordinates of the eyes detected in the first image.  $\mathbf{P}_i, i = 3, 4, \dots, N + 2$  are the backprojected face plane coordinates of the motion feature positions  $\mathbf{p}_i$ . The model points for motion features are updated for each incoming frame, but the eye positions are fixed after initialization. Also, in this case the motion is modelled using a first order dynamic model.

The measurement model is needed to relate the 3-D pose parameters to the 2-D image observations. We use a perspective camera model to transform the object coordinates  $\mathbf{P}_i$  to image coordinates  $(u_i, v_i)$ . This non-linear observation model is linearised using its partial derivatives with respect to the state variables that include the 3-D pose parameters of the camera with respect to the head and their velocities.

The actual measurements  $\mathbf{z}_i, i = 1, \dots, 2$  for the eyes are extracted in a region around the predicted locations. The size for the region is adjusted dynamically by projecting the prediction uncertainty to the image coordinates. The eye detection is performed as described in Section 2.3 and  $(u, v)$ - coordinates are obtained as a result. If there is too much deviation from the prediction, then the predicted position is chosen instead of the actual measurement. In the case of motion features, the measurements are  $\mathbf{z}_i = \mathbf{p}_i + \mathbf{d}_i, i = 3, \dots, N + 2$ , where  $\mathbf{p}_i$  is a motion feature position in the previous frame and  $\mathbf{d}_i$  is the detected displacement in the current frame. The measurement noise for the eyes is assumed to be Gaussian with zero-mean and a variance of 9 pixels resulting in the error covariance matrices  $\mathbf{C}_1 = \mathbf{C}_2 = 9\mathbf{I}$ . Covariance matrices  $\mathbf{C}_i, i = 3, \dots, N + 2$  are derived for motion features using the method described in Section 4.2.3.



**Fig 4. Facial feature tracking example. Revised from [II]. © 2008 Springer.**

In Paper II, we assessed the feature tracking subjectively because the ground truth for the pose was not available. Fig. 4 shows an example result of the extracted facial feature locations during tracking. The tracking was successfully completed without interruptions in all of the sequences. There were sometimes large errors in the eye measurements or rarely the eye was not detected at all. In these situations the tracker used the predicted measurements and tracking did not fail. Although the lighting changed continuously throughout the sequences, the system still worked properly. The advantage of the method is that real-time performance on a resource limited mobile device can be easily achieved. In Chapter 5, we also describe how the method was successfully used for controlling the spatially aware user interface with a camera-equipped mobile phone.

### **3.5 Discussion**

In this thesis, real-time performance has been one of the main goals. As discussed above, the well-known Kalman filtering framework offers tools for estimating the pose efficiently. In our solution, we propose a tracking system using only three facial features: the eyes and the mouth. The method of extracting the features was introduced in Chapter 2. The new idea in Paper I was to use these three points to solve the pose with the P3P algorithm and use the resulting pose as a measurement in Kalman filtering. The advantage of this is that we can use linear measurements instead of non-linear measurements. If we had used a perspective camera model to map measurements to the state variables, linearisation of the measurement model would have been needed. The drawback is that using only three points in pose estimation makes the system susceptible to noise. If there are errors in the facial feature extraction, this certainly decreases the pose estimation accuracy; also user interface control suffers dramatically. Therefore, special attention should be paid to the development of facial feature extraction methods.

In contrast to other methods in the literature that use facial features or optical flow only, we apply both ways to improve the performance of the new system introduced in Paper II. First, face and eye detection is performed using the methods described in Chapter 2. Then, we apply a new motion estimation framework developed in Paper III to extract motion features from the face region. Extracting facial features makes it possible to obtain more accurate absolute positional information, whereas utilising optical flow increases the reliability of the system. However, the extracted features are not tracked in the current solution. Including tracking and filtering will increase the accuracy the

pose estimates. This will be considered in our future work. Compared to the method presented in Paper I, we achieve more reliable performance. The shortcoming is that the computational cost is slightly increased.

The systems introduced in this chapter were basically tested with real image sequences in realistic usage scenarios. Although very good preliminary results were achieved, it should be remembered that the results obtained only apply under some constraints. Further work is still needed, for example, to make algorithms work in greatly varying environmental conditions. At this point of development, the conducted testing was enough to prove the correctness of the approaches. Especially the low computational cost of the method makes it usable for real-time applications. Compared to the state-of-the-art methods such as (Cascia *et al.* 2000, Xiao *et al.* 2003, Dornaika & Orozco 2007) our methods are not so robust to out-of-plane rotations. However, considering the intended applications, it is not necessarily required.

In future, the facial features to be extracted and tracked could be the corners of the eyes and the mouth instead of the centre points. The corners are more permanent than the centres of these features due to possible deformations during tracking. The natural deformations of the features will cause large deviations in the image measurements and pose estimation. Also, the amount of features is increased from three to six, which allows more robust pose estimation and tracking. The tip of the nose could also be a rigid feature to be extracted, but that is a very difficult task due to the low contrast of skin. Another important issue is to choose an appropriate model for the head in the tracking phase. In this thesis, very simple models are used for the face. In the future, more accurate 3-D models could be applied that make it possible to achieve better head pose estimates. The head model could be, for example, a 3-D wire frame model. A shortcoming with more advanced models is that they typically increase the computational load.

An alternative for future work is to use Kalman filtering to recover also the structure of the tracked head (Jebara *et al.* 1999). In this case, features do not necessarily have to be facial features like eyes or mouth, but only stable features are needed such as the motion features utilised in Paper II. The advantage is that there is no need for some particular feature such as the eye to be visible all the time in the tracking phase. However, tracking the same features over the sequence improves the estimation accuracy and help to avoid drifting. One interesting approach to be considered for this problem is the use of Simultaneous Localisation and Mapping (SLAM) to track the device motion while

simultaneously creating a 3-D map of the head. For example, the recently introduced MonoSLAM (Davison *et al.* 2007) could be appropriate for real-time applications.

### **3.6 Summary**

In this chapter, we first discussed visual tracking and head pose estimation in general and reviewed some representative work on head tracking. Then, we proposed two different approaches to build a head tracker. First, we introduced a system that utilises Kalman filtering to track and estimate the head pose by extracting the eyes and the mouth from the face region. Although very good preliminary results were achieved, using only three points makes the system susceptible to noise. In Paper I, it was concluded that more stable features are needed, which do not necessarily have to be predefined facial features.

Therefore, in the second solution, besides the eyes also motion features are extracted from the face region. The use of a large number of motion features allows us to acquire more evidence about face movement than using only three facial points. A drawback is that motion features provide only relative information. Therefore, the eye positions are added to the tracker in order to avoid drifting problems. The results can be further improved by adding more fixed points to the face model, providing enhanced positional information. Another way of improving estimation accuracy and avoiding the drifting problem can be achieved by tracking 2-D image features through several successive images.

We conclude that the solutions proposed in this chapter can be powerful enablers for controlling spatially aware user interfaces. This is demonstrated in the applications presented in Chapter 5.

## **4 Motion estimation for mobile user interaction**

Computer vision based motion analysis offers new intuitive ways of communicating with hand-held electronic devices. The information used as an input for the user interface is, for example, an absolute camera 3-D pose with respect to a given target, or a relative camera 2-D movement measured between two successive images. The target of the known geometry in pose estimation can be either markers placed in the scene or some object such as the user's face or finger. The detected movement can be then used for 2-D navigation or 3-D manipulation applications on the device.

This chapter presents techniques for extracting motion from image sequences for mobile user interface purposes. The introduction begins with a review of related work on the use of the motion as an input in camera based interaction with mobile devices. Section 4.2 introduces approaches to measuring 2-D image motion from successive frames and proposes a new method using a sparse set of features. Then, two approaches which require 2-D motion estimation as a preprocessing step are described. Firstly, Section 4.3 presents a method for estimating the ego-motion of the camera while the user operates the mobile device through a series of hand movements. Secondly, Section 4.4 describes a technique for tracking and estimating the motion of an object such as a finger while it is moved in front of the camera.

### **4.1 Related work on motion based interaction**

The first solutions using camera motion detection as an input for hand-held devices originated from the commercial domain. In 2003, Siemens introduced an augmented reality game called *Mozzies* developed for their SX1 cell phone. The goal of the game was to shoot down the synthetic flying mosquitoes projected onto a real-time background image by moving the phone around and clicking at the right moment. During user movements, the motion of the phone is recorded using a simple optical flow technique.

Not long after the first camera-equipped mobile devices became available, also researchers recognised opportunities for the vision based user input. The work carried out in this field can be roughly divided into marker based and markerless approaches.

### **4.1.1 Approaches using markers**

In marker based methods, the pose of the objects or the camera is estimated using artificial markers or fiducials placed in the field of view of the camera (Lepetit & Fua 2005). The advantage here is that the system can perform fast segmentation and detection of the markers. For example, Möhring *et al.* (2004) presented a tracking system for augmented reality on a mobile phone to estimate 3-D camera pose using special colour coded markers. Other methods of this category use a printed or hand-drawn circle (Hansen *et al.* 2005), a hand-held target (Hachet *et al.* 2005) and a set of squares (Winkler *et al.* 2007) as markers to facilitate the tracking task.

One new solution was presented by Pears *et al.* (2008). The idea of this approach was to use a camera on the mobile device to track markers on the computer display. This technique can compute which part of the display is viewed and the 6-DOF position of the camera with respect to the display.

The main drawback of these approaches, as far as mobile user interfaces are considered, is that it is often impractical or even impossible to modify the scene and place markers in most real usage environments. Therefore more flexible markerless methods have attracted more attention recently.

### **4.1.2 Markerless approaches**

An alternative to markers is to estimate motion between successive image frames with similar methods to those commonly used in video coding (Wang *et al.* 2001). Rohs (2004) divided incoming frames into the fixed number of blocks and then determined the relative x, y, and rotational motion using a simple block matching technique. The computational cost of matching can be reduced significantly by using a hierarchical algorithm and optimising the search space. Based on this fact, Liu *et al.* (2005) proposed a multi-resolution scheme that includes building a pyramid of images and prediction of motion vectors using previous detections. They assumed that the camera motion can be approximated using the mean of the block displacements. Wang *et al.* (2006) presented a similar approach which performed grid sampling and applied full search block matching on temporal adjacent frames. Hua *et al.* (2007) also explored a set of fast matching methods and designed an online scheme to switch among the different matching algorithms, which makes a compromise between quality and computational expenses. The motion gestural events are then defined based on the motion vectors estimated for each

frame. In addition, they proposed overall lightness and blurriness measures for the image to define visual events. Although, these straightforward techniques are fast to compute and have linear complexity in the size of the image, they are known to make erroneous estimates in the absence of texture.

Another possibility is to extract distinctive features such as edges and corners from images which exist naturally in the scene. Haro *et al.* (2005) have proposed a feature based method to estimate movement direction and magnitude. First, edge detection is performed on two successive frames. Then, thresholding is used to find corner-like features and feature correspondences are searched for using template matching. Finally, they performed direction voting to estimate the motion. Compared to the motion estimated using direct block matching methods, these feature based techniques provide clearly more robustness and accuracy.

Instead of using local features, some approaches extract global features such as integral projections from the image. Integral projections are defined to be sums of the grey levels along any fixed direction in the image. Drab & Artner (2005) first computed horizontal and vertical projections from grey-scale images and then searched for the best match between projections of two successive images. Adams *et al.* (2008) first computed a translation by aligning integral projections of edges detected in two images. The estimate is then refined to compute a 2-D similarity motion model using corner features extracted. Even though the salient advantage of projections is simple and fast computation, they often fail in the case of noisy images and repeating textures.

Some recent and generally interesting direction for mobile interaction is to combine information from several different sensors. In their feasibility study, Hwang *et al.* (2006) combined forward and backward movement and rotation around the Y axis data from camera based motion tracking, and tilts about the X and Z axis from the 3-axis accelerometer. Their vision system uses the pyramidal implementation of the Lucas-Kanade feature tracker (Lucas & Kanade 1981). Recently, a technique to couple wide area, absolute, and low resolution global data from a GPS receiver with local tracking using feature based motion estimation was presented by DiVerdi & Höllerer (2007).

## 4.2 Computing 2-D image motion

Measurement of motion in image sequences is based on interpretation of spatial and temporal variations in image intensities. The goal is to determine the 2-D motion field, that is, the projection of the 3-D velocity field of moving points in the scene onto the

image plane (Stiller & Konrad 1999). Accurate estimates for the motion field are often inaccessible due to illumination changes in the image and the well known aperture problem (Horn & Schunck 1981, Verri & Poggio 1989). In fact, the observed motion may not be the same as the true physical motion and only an approximation of the motion field, called apparent motion or optical flow can be computed (Haußecker & Spies 1999). Computing optical flow is often the first stage of computation, for example, where the goal is the recovery of the camera ego-motion or segmentation of the image into parts corresponding to different moving objects through the analysis of the motion of features or brightness patterns (Aggarwal & Nandhakumar 1988).

Barron *et al.* (1994) divided algorithms to compute optical flow into four categories including differential methods, energy-based methods, phase-based techniques and region-based matching. In a comparison of these four techniques, differential techniques (Lucas & Kanade 1981) perform best in terms of efficiency and accuracy. The phase-based approach (Fleet & Jepson 1990) was found to be most accurate, unfortunately it is less efficient in implementation. Energy-based methods (Heeger 1988) perform the hierarchical decomposition of the image sequence in the frequency domain. This provides simultaneous localisation in spatio-temporal and frequency domains. On the other hand, region matching techniques (Anandan 1989) are more robust to noise and less sensitive to illumination changes. Later, Liu *et al.* (1998) also evaluated accuracy versus efficiency trade-offs resulting, for example, from hardware implementation constraints, algorithm flexibility and robustness.

Methods of computing optical flow are also often classified into two groups (Simoncelli *et al.* 1991). This classification effectively groups methods to those that perform computations on the spatio-temporal gradient of the image intensity, and those that match features between successive image frames. In the following, we use this classification for its simplicity.

### **4.2.1 Gradient based methods**

Gradient based methods perform computations on the spatio-temporal partial derivatives to estimate image flow at every position in the image. Horn & Schunck (1981) presented a global method that combines a global smoothness term with a gradient constraint equation to obtain a functional for estimating optical flow. They use the smoothness term that minimises the absolute gradient of the velocity. In general, global approaches supplement the optic flow constraint with a regularising smoothness term.

Global methods yield dense flow fields, but are experimentally known to be sensitive to noise and appearance variations (Barron *et al.* 1994). They are suitable in cases where image motion is small.

Local techniques use spatial constancy assumptions on the optic flow field. Lucas & Kanade (1981) introduced a popular method using a local window to determine the flow of a particular image point. It uses a weighted least-squares (WLS) fit of local first-order constraints to a constant model for motion in each small spatial neighbourhood. One important advantage of this approach is the existence of a confidence measure. This algorithm provides a good accuracy versus efficiency trade-off. In general, local methods typically offer relatively high robustness under noise.

### **4.2.2 Feature based methods**

While gradient based methods generally compute the motion of every pixel of the image and thus produce a dense flow field, it can be advantageous to do this computation only for an exclusive number of distinct features. Under the assumption of a dense sampling rate, feature displacements can be used to achieve an approximation of the optical flow (Barron *et al.* 1994). In these solutions, feature correspondences between successive images are established by feature matching or region matching techniques resulting in a sparse optical flow field (Haußecker & Spies 1999).

In the feature matching approach, interest points (tokens) are first detected from both frames, and correspondences between them are determined. As a result, displacements of these features describing motion between frames are obtained. For example, work related to invariant interest points (Mikolajczyk & Schmid 2004), where a set of region descriptors for detected features is computed and used for matching, belongs to this category. Descriptors invariant to specific geometric and photometric transformations can deal with large displacements and appearance changes of features (Torr & Zisserman 1999). For example, the SIFT (Lowe 2004) feature point detector has become very popular due to its good performance. SIFT features are invariant to image scale and rotation changes.

Another solution is to first select distinctive features from one frame and then measure displacements of these features. From the initial work of Lucas & Kanade (1981), Tomasi & Kanade (1991) developed a widely used technique, the so-called the KLT feature tracker, which first detects features by analysis of image gradients, and then iteratively minimizes the sum of squared differences (SSD) criterion using a pyramidal

implementation in feature displacement estimation. Later, Shi & Tomasi (1994) proposed the use of an affine transformation model instead of using a translation model like in the original KLT tracker. Tommasini *et al.* (1998) also extended this tracker by introducing an automatic scheme for rejecting spurious features.

Anandan (1989) introduced a multi-scale method using a Laplacian pyramid (Burt & Adelson 1983). A coarse-to-fine search is performed such that larger displacements are first determined and then improved with a more accurate higher resolution version of the image. This strategy is well suited to large displacements but is less successful for sub-pixel motion. Another similar approach minimizing SSD and employing a pyramid search is presented by Singh & Allen (1992). Interestingly, they use a three-frame method for the region matching to average out temporal error in the SSD.

Yet another region matching algorithm was presented by Camus (1997). His real-time algorithm is based on the idea that performing a search over time instead of over space is linear in nature rather than quadratic, resulting in a quantised sub-pixel displacement field. The algorithm uses only integer computations and thus may be efficiently implemented in practice.

Overall, feature based methods are suitable especially when image motion is large. Furthermore, the coarser search provides faster implementation and more robust results. However, the cost of the improved efficiency is reduced accuracy (Liu *et al.* 1998).

### **4.2.3 Estimating motion using a sparse set of feature blocks**

There is no algorithm for estimating optical flow which is clearly superior to the others (McCane *et al.* 2001). Given the target platform of our application, hardware accelerators for video encoding may be available to perform block matching computations. Therefore, it is interesting to consider feature based solution, where block matching is used to provide motion estimates. So, we propose an approach where a sparse set of feature blocks is first selected from one image and then the displacements are determined. We pay attention especially to the confidence analysis of block matching, whose results can be utilised in further analysis.

The idea of motion profile analysis was originally presented by Sangi *et al.* (2004). Then, the proposed motion estimation framework was first applied to mobile phones by

Hannuksela *et al.* (2005) and later more comprehensively discussed in Paper III. In the next two sections, this framework is briefly reviewed.

### Selecting distinctive block features

Feature motion estimation begins with the selection of the feature blocks from the first image. The goal is to ensure that the feature blocks are distributed over the image so that the probability of sufficient presentation of overall image motion is high. We use a computationally straightforward way where the image area is split to non-overlapping regions and one block is selected from each region.

Another goal is to select some distinctive features which guarantee high precision in the estimation of the displacement vectors. Various criteria for selecting such good features exist, and they usually measure the richness of the texture within an image area (Shi & Tomasi 1994). Comparison of feature candidates is typically based on analysis of the spatial gradient. For example, measures based on the Harris response function (Harris & Stephen 1988) or eigenvalues of the normal matrix (Tomasi & Kanade 1991) can be used.

In Paper III, our approach is to consider first-order image derivatives in the horizontal and vertical directions. The sum of squared derivatives provides a computationally simple criterion. An alternative approach is described in Paper IV, where we have used eigenvalue analysis of 2 by 2 normal matrices which can give more reliable features, but requires more computation, and they may also give a strong response for non-corner features. The results of both feature block selection schemes are shown in Figs. 5 (a) and (b), respectively.



**Fig 5. Feature selection: (a) using the sum of squared derivatives (b) using eigenvalue analysis.**

With regard to utilisation of temporal continuity, it is possible to modify the selection scheme described to use also feature tracking. Some improvements in motion estimation accuracy can be achieved with such a scheme. However, this comes at the expense of extra analysis of the feature reliability and there is also a need to keep and update a feature list in tracking. Due to this complexity, tracking is not used in our implementation.

### **Measuring feature displacements**

To estimate the displacement of a selected feature, the dissimilarity between a feature block region in the anchor image and a candidate block in the target image is measured. The best match is found by maximising a similarity measure such as the normalised cross-correlation or minimising distance measures such as the sum of squared differences (SSD) or the sum of absolute differences (SAD) (Aschwanen & Guggenbühl 1992). The selection of particular criteria is a trade-off between the computational load, the algorithm complexity and the performance achieved.

In Paper III, an SSD block matching measure is evaluated for a suitable range of integer displacements in both x- and y-directions. The surface of matching measure values obtained in this way is called motion profile. Instead of using the SSD, Paper IV proposes the use of its variant zero mean sum of squared differences (ZSSD). The latter measure is more robust to illumination changes, which can be crucial in some applications (Aschwanen & Guggenbühl 1992). Exhaustive evaluation of either of these measures gives a motion profile. The displacement that minimises the criterion provides the best match feature displacement which can be refined to sub-pixel accuracy via quadratic interpolation of the motion profile values (Haralick & Shapiro 1993).

Minimisation of the SSD or any other distance measure is based on the assumption that the true motion is close to the displacement giving the best match. However, due to noise and the aperture problem, there might be many other good matches, for example, at any point along a straight edge. This means that there is reliable information available only in the edge normal direction. On the other hand, for homogeneous image regions good match may be obtained for any displacement candidate.

As the reliability of displacement vectors is always questionable, it is often helpful to complement estimates with uncertainty measures so that the weaker features like edges produced can be used in following computations. Barron *et al.* (1994) discuss various confidence measures proposed in the literature. In an early work, Anandan

(1989) computed directional confidence measures for the displacement estimates by analysing the curvature of the motion profile in the vicinity of the best match. Singh & Allen (1992) interpreted the SSD values probabilistically, and computed a weighted displacement estimate and associated error covariance matrix. Probabilistic analysis of SSD-based matching results has also been considered by Nickels & Hutchinson (2002). More recently, Patras *et al.* (2002) proposed a probabilistic framework, where the reliability of the SAD estimate was determined by the analysis of block intensity variation.

Paper III and Paper IV present a statistical method for computing confidence measures where uncertainty of the estimate obtained is analysed by searching for displacements that are close to the measurement giving the minimum matching value. The selection of the set of displacements is based on thresholding of the motion profile. The details of the method can be found in the corresponding publications. The result of this analysis is summarised as covariance matrices. In general, such matrices provide a powerful tool for error treatment (Haußecker & Spies 1999).

As a result of these computational steps, we obtain a set of *motion features*. A motion feature consists of the block centre location in the first image, its displacement estimate, and the uncertainty covariance matrix. This information can be directly used to estimate, for example, the ego-motion of the device in the user's hand or tracking multiple motions as described in following sections. Fig. 6 shows an example of the estimated feature displacements and the related uncertainties.



**Fig 6. Estimates of feature block displacements (lines) and associated error covariances (visualised using ellipses).**

## 4.3 Camera ego-motion estimation

A mobile user interface system controlled through a series of hand movements requires a method for estimating the ego-motion of the device's camera. Camera ego-motion is often estimated from 2-D image motion measured between two successive frames. As the observed motion in an image sequence may consist of multiple motions due to moving objects in a scene and motion parallax, one must consider solutions that estimate the dominant motion.

### 4.3.1 Estimating dominant global parametric motion

To define the dominant motion estimation problem, consider a situation where motion between two frames can be modelled by splitting both frames into regions which exhibit coherent motion components. Coherence here means that motion of each region is piecewise smooth, or it can be approximated with some parametric motion model (Black & Anandan 1996). Dominant motion estimation considers the latter kind of models and tries to estimate a parametric motion model for the largest region. An affine model is typically used. Having such a model, without necessarily knowing the region supporting it, is sufficient for the application considered here. In order to diminish the effect, which other motion regions have on the dominant motion estimate, some robust schemes are needed for processing information.

The camera ego-motion is often estimated directly from image intensities or indirectly from the local 2-D image disparities (Irani *et al.* 1994a). In direct methods, some robust error norm is applied to the motion compensated frame difference in order to evaluate a particular motion model. Gradient based approaches, where minimisation of cost function is performed iteratively using spatial and temporal gradient data in a multi-resolution framework belong to this category (Odobez & Bouthemy 1995, Black & Anandan 1996, Bober & Kittler 1994, Sawhney & Ayer 1996). Burt *et al.* (1991) presented an analysis of gradient based estimation using dissimilarity metric between image regions. Such an estimator is shown to exhibit two selection modes: it may either average the component motions, or it may select one. Burt *et al.* (1991) provided conditions for the latter case to occur. Robust error norms (Black & Anandan 1996, Odobez & Bouthemy 1995) can be used for relaxing such conditions.

Indirect solutions use precomputed 2-D image disparities, as described in Section 4.2, as input, and the need for robust processing is emphasized due to possible invalid

measurements. Voting-based schemes such as random sample consensus (RANSAC) (Fischler & Bolles 1981) or Hough transform (Hough 1959) can be used for extracting inliers from data. The RANSAC method employs a generate-and-test principle, where randomly selected sufficient subsets of available observations are used for generating hypotheses about the underlying model, and then all the data is used for testing those hypotheses. When some hypothesis gets enough support, one may compute a refined estimate using that part of data, which supports the hypothesis. In the Hough transform, the parameter space is discretised, and data is used for computing supports for each discrete set of parameter values.

In the following, we propose a computationally efficient method for estimating dominant global motion. The sparse set of motion features described in Section 4.2.3 are used as an input for the method. In the following, the approach is introduced very briefly and more details can be found from Papers III and IV.

### 4.3.2 *Modelling motion*

The ego-motion estimation generally refers to the computation of 6-DOF motion. However, the choice of a model and the number of parameters for the ego-motion computation is application dependent. For a review of different models used in motion estimation and image registration, the reader is referred to the book by Hartley & Zisserman (2004). We model the device movement using a four parameter similarity model which is considered to be sufficient for approximating motion between frames, as it can represent 2-D motion consisting of translation, rotation, and scaling (Zheng & Chellappa 1993). With this model, the displacement  $\mathbf{d}$  of a feature located at  $\mathbf{p} = [x, y]^T$  is represented using

$$\mathbf{d} = \mathbf{d}(\theta, \mathbf{p}) = \mathbf{H}[\mathbf{p}]\theta = \begin{bmatrix} 1 & 0 & x & y \\ 0 & 1 & y & -x \end{bmatrix} \theta, \quad (5)$$

where  $\theta = [\theta_1, \theta_2, \theta_3, \theta_4]^T$  is a vector of model parameters and  $\mathbf{H}[\mathbf{p}]$  is an observation matrix. Here,  $\theta_1$  and  $\theta_2$  are related to common translational motion, and  $\theta_3$  and  $\theta_4$  encode information about 2-D rotation  $\phi$  and scaling  $s$ ,  $\theta_3 = s \cos \phi - 1$  and  $\theta_4 = s \sin \phi$ .

### 4.3.3 *Dominant global motion estimation*

The global motion describing the device motion is estimated using those motion features which pass an outlier analysis stage. Such analysis is necessary as feature dis-

placement estimates can be erroneous due to image noise, or there may be several independent motions in the scene. It is assumed that the majority of motion features are associated with the global motion we want to estimate. To select those inlier features, we use a RANSAC based scheme where pairs of motion features are used to instantiate motion model hypotheses, which are then voted for by other features.

A feature votes for a hypothesis if the displacement instantiated from the hypothesis is close to the estimated displacement. The covariance matrix described in Section 4.2.3 provides information about the feature motion uncertainty in different directions, and the calculation of votes uses the Mahalanobis distance measure. Once inlier features have been selected, a weighted least squares approach is used to compute the estimate of the device motion. Primarily, weighting is based on measured uncertainties.

Experiments in Paper III, validate the quantitative performance in synthetic image sequences and the correct operation in real image sequences. Also, the performance of outlier analysis was verified with good results. In Paper IV, the usefulness of the uncertainty analysis was validated and the method was compared to the gradient based multi-resolution method (Odohez & Bouthemy 1995). The proposed feature based method achieved better results when testing on real images.

## 4.4 Multiple motion estimation

Approaches for multiple motion analysis from the observed 2-D image motion can be divided into methods that compute the multiple motions without performing segmentation, and those which recover the motion using segmentation (Irani *et al.* 1994b). The dominant motion approach described in Section 4.3.3, for example, estimates the motion without performing the segmentation. Whereas, approaches to multiple motion analysis using segmentation try to segment out different objects or differently moving regions in an image according to their motion coherence.

Instead of performing the segmentation and estimation steps individually for each successive image frame, it is advantageous to increase the temporal interval to more than two frames. Visual tracking of multiple moving objects is often implemented with the Kalman filtering framework, as described in Chapter 3. Despite many advantages of Kalman filtering approaches, the main drawback is the underlying assumption of a uni-modal Gaussian distribution. This assumption is violated in the case of incoherent motion as the probability density is essentially multi-modal and therefore non-Gaussian.

Numerous approaches have been proposed to overcome this limitation. Among

them, particle filtering or factorised sampling method (Isard & Blake 1996) has been successfully applied to track multiple moving objects in cluttered environments. Particle filtering methods are directly applicable to multiple motion tracking because they allow for non-Gaussian probability functions, even multi-modal distributions. However, their computational requirements make it difficult to achieve a real-time performance on mobile devices, even though more efficient modifications (MacCormick & Isard 2000) of the original algorithm are already available. Another approach to tracking multiple motions is the use of Gaussian mixture models (Xiong *et al.* 2006). The unfortunate drawback also in this case is the high computational cost.

#### **4.4.1 Data association**

In general, most of the existing algorithms utilised for visual tracking deal with estimating the state of a single object only. One way to extend traditional algorithms to track multiple object motions and cope with multi-modal distribution is combinatorial data association methods. Rao (1993) groups data association methods into optimal approaches in a Bayesian sense and suboptimal approaches. The former is also known as Multiple Hypothesis data association. The most well-known suboptimal approach for multiple motion tracking is the Joint Probabilistic Data Association filter (Fortmann *et al.* 1983).

In many tracking problems there is more than one measurement at the same time step available. Data association is a process to assign each of these measurements to the appropriate objects or motion. Assigning measurements can be effective in the case of incoherent motion. For example, Reid (1979) estimates the states of multiple targets from data association hypotheses in a cluttered environment with the Kalman filter. Methods of this kind often perform data association and estimation separately by first assigning the measurements and then estimating the state.

In this thesis, we propose a novel method able to track multiple motions using a sparse set of motion features described in Section 4.2.3. Our approach employs soft assignment to associate the measurement data to the corresponding motion components. The method, originally described in Paper V, embeds the EM algorithm into the Kalman filter stages to estimate multiple states simultaneously. One benefit compared to previous approaches is that no iterations are needed, making the algorithm computationally efficient.

#### 4.4.2 Object tracking using the Kalman-EM algorithm

An intuitive way to interact with mobile devices is achieved by moving some object, such as a finger, in front of the camera and then recognising the observed movements. In this application, we cannot assume that a single motion component is present in the image. With hand-held devices the camera also moves slightly when the user is operating the device. The problem is therefore formulated as a task of estimating two distinct motion components, the camera motion and the object motion. However, we are not so interested in segmenting the observed displacements into coherent regions in an image.

The state-space model of the camera ( $j = 1$ ) and object ( $j = 2$ ) motions is

$$\mathbf{x}_j(k) = \mathbf{x}_j(k-1) + \boldsymbol{\varepsilon}_j(k), \quad (6)$$

where  $\mathbf{x}_j(k) = [u_j(k), v_j(k)]^T$  denotes the motion between the frames  $k-1$  and  $k$ , and  $\boldsymbol{\varepsilon}_j(k)$  is the process noise term, which is assumed to be zero-mean white Gaussian noise with the covariance matrix  $\mathbf{Q}_j = \sigma_j^2 \mathbf{I}$ .

Object tracking uses the motion features described in Section 4.2.3 as input. Observed displacements of those features,  $\mathbf{d}_i(k)$ , are modelled as

$$\mathbf{d}_i(k) = \lambda_i \mathbf{x}_1(k) + (1 - \lambda_i) \mathbf{x}_2(k) + \boldsymbol{\eta}_i(k), \quad (7)$$

where  $\boldsymbol{\eta}_i(k)$  is the observation noise, which is assumed to obey zero-mean Gaussian distribution with a covariance matrix  $\mathbf{R}_i$ , and  $\lambda_i$  is a hidden binary assignment variable, which indicates the object that generates the measurement.

To estimate the motions we use a technique where the Kalman filter (Kalman 1960) and the EM algorithm (Dempster *et al.* 1977) are combined. Note that the Kalman filter could be used to directly estimate  $\mathbf{x}_j(k)$  if the assignments  $\lambda_i$  were known. As these assignments are unknown, the predicted estimates of  $\mathbf{x}_j(k)$  and *a priori* probabilities of associating features to motion components are used to compute soft assignments  $w_{i,j}$  using a Bayesian formulation, as described in Paper V. The assignment step corresponds to the E step of the EM algorithm.

Soft assignments are then used in the computation of the Kalman gains which are needed to get the filtered estimates of  $\mathbf{x}_j(k)$ . The principle is that the lower the value of  $w_{i,j}$  is the higher the observation noise covariance becomes. This weighting of the measurements corresponds to the M step of the EM algorithm. Paper V describes ex-

periments where the approach is successfully applied to finger tracking from images captured with a mobile phone.

## 4.5 Discussion

As already mentioned in Sections 4.1 and 4.2, motion estimation in images has been a subject of computer vision research for decades. The basic problem in applying methods developed for desktop computers or special hardware on mobile phones is to find a good compromise between computational efficiency and estimation accuracy. Many of the state-of-the-art methods need too many resources to be directly transferred to these devices. Other problems are related to robustness against errors and implementation difficulties as discussed in Paper III.

To overcome these problems we have implemented a computationally efficient solution to estimate motion from images, while still retaining reasonable accuracy. The proposed framework includes following steps: distinctive feature selection, feature displacement measurement, outlier analysis, and dominant global motion estimation. The novelty of the approach lies in the use of the new motion feature uncertainty measure in all the consecutive processing steps.

Our feature selection method, which is based on squared derivatives is faster to compute than eigenvalues. The drawback is that the method does not always give reliable features. However, reliability analysis of feature displacements partly compensates for this. The advantage of our method for computing confidence measures is that it only requires thresholding of the SSD surface using gradient information. There is no need to model the matching result using computationally expensive exponential functions as in previous solutions such as (Singh & Allen 1992, Nickels & Hutchinson 2002). Furthermore, the motion estimation framework benefits from the use of uncertainty information also in outlier analysis and global motion estimation. Experiments with synthetic sequences indicate comparable results to the reference method (Odobez & Bouthemy 1995). However, with real images our method obtained higher accuracy. Like all feature based methods, our method assumes that good features can be found. In future work, the framework can be modified to include support for hierarchical motion estimation. Constructing image pyramids is likely to further improve the efficiency of the method.

For multiple motion tracking, we introduce a new solution in Paper V. The proposed Kalman-EM algorithm is closely related to switching Kalman filters (SKFs) (Murphy 1998). The main difference is that our method concentrates on associating measurement

to different objects whereas the SKFs try to model changes of object dynamics using Gaussian Mixture Models by switching between separate filters. The benefit of our method lies in its simplicity. However, the method requires good initial weights that might be difficult to derive in some cases. In future work, this should be considered more profoundly.

## **4.6 Summary**

We have presented here two new techniques for estimating motion in image sequences in order to control the user interface of a mobile device; the first by detecting the ego-motion of the device held in the user's hand and second by measuring simultaneously the ego-motion as well as the movements of an object such as a finger. Both solutions utilise a new method of computing motion features presented in this thesis. This method is based on selecting a sparse set of distinctive feature blocks and measuring their displacements.

In order to enable real-time user interaction, methods have been implemented in software using only fixed-point arithmetic that guarantees sufficient performance also on low-power mobile devices. Apart from their effectiveness, another important feature of both methods is that they can be installed on certain existing camera-equipped devices.

Vision based motion estimation using a sparse set of features of course has its limitations. First, we assume that there is no significant change in the appearance of the objects between image frames. Secondly, the well-known aperture problem can be present due to the limited feature block size in matching. Finally, the search window used in matching has also a limited size, which is a trade-off between the maximum speed of object movement allowed in an image and the computational effort needed. Despite such limitations, we can say that our solutions provide a viable alternative for interacting with hand-held mobile devices.

## 5 Motion analysis in applications

The previous three chapters dealt with computer vision methods to estimate head, device, and object motion. This chapter describes new possibilities that these methods provide: input to mobile user interfaces and applications. Overall, the applications presented show that mobile user interfaces benefit from the enriched interaction modalities that computer vision offers. All the applications are explained in the following Sections 5.1 - 5.3. These solutions are presented here very briefly and many details can be found in the original publications. In Section 5.4, some practical matters are discussed. Finally, Section 5.5 briefly discusses the contents of this chapter.

### 5.1 Controlling spatially aware user interfaces

Navigating large information spaces can be disorienting even on a large screen. In mobile devices with a small screen, the user often encounters situations where the content that is needed for display exceeds what can be shown on the screen. For example, large digital images are becoming commonplace due to the increasing availability of high resolution imaging and map navigation applications.

A viable alternative to improve interaction capabilities is spatially aware displays (Fitzmaurice 1993). The solution is to provide a window on a larger virtual workspace where the user can access more information by moving the device around. Impressed by this approach, Yee (2003) presented a peephole display technique to control, for example, the 3-D image viewer and the 3-D calendar applications on a hand-held computer using several additional position sensors. Recent work on small sized devices such as mobile phones has focused on employing motion data measured using accelerometers (Hinckley *et al.* 2000) or cameras, as described in Chapter 4, for this same purpose.

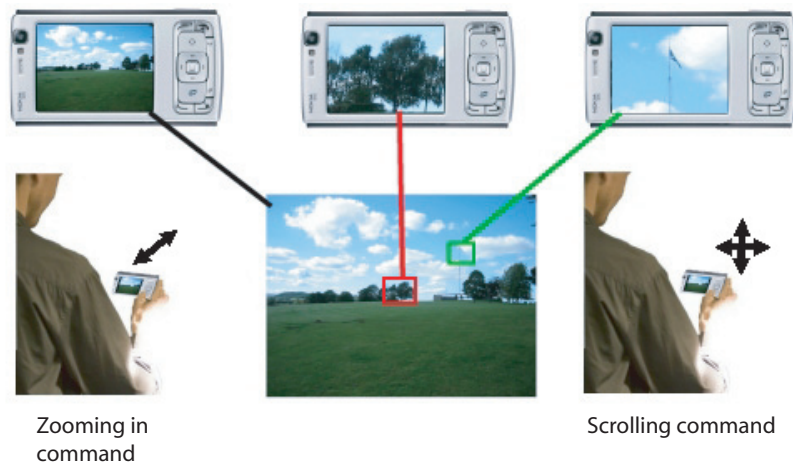
In this thesis, two possible camera based user interaction approaches for mobile devices are considered. Section 5.1.1 introduces a technique which uses the ego-motion estimation of the device. The relative motion information acquired can be used directly or after integration as absolute positional input for controlling the spatially aware display.

However, when integrating motion data, errors typically accumulate over time. In some cases, rough motion estimates are not enough and more accurate data is required.

A method presented in Section 5.1.2 determines the position and orientation (pose) of the device with respect to the user's face and utilises this information for browsing the virtual workspace of the mobile terminal in its display.

### **5.1.1 Device motion as an input device**

With motion input, the user can operate the phone through a series of hand movements whilst holding the device. During these movements the motion is extracted from the image sequence captured by the camera. As an application example, the solution has been implemented on Nokia N-series mobile phones, allowing the user to browse large image documents on small screens as shown in Fig. 7. In this solution, the camera is pointing towards the user's face, which guarantees that there are always good features available in the view.



**Fig 7. Motion based user interface estimates the motion of the device relative to the user or the scene enabling browsing and zooming functionalities.**

In the solution, originally presented in Paper III, only a small part of the high resolution image is visible at a time and the measured motion information is used as input. For instance, the lateral movement upwards scrolls the focus towards the upper part of the display, and back and forth motion is interpreted as zooming in and out. The rotation component is utilized to change the orientation of the display. In practise, the user can also tilt the device in order to navigate over the display, which is a more natural and

convenient way of controlling the device. Compared to the use of HW accelerometers alone (Eslambolchilar & Murray-Smith 2004), a camera based approach allows a more convenient way of controlling zooming effects and adapts better to cases where the user is moving, such as walking, since the global motion is essentially estimated from the face location.

The device ego-motion estimation approach used in this application was already introduced in Section 4.3. In Paper III, also effective software implementation is discussed. We have implemented our method using only fixed-point arithmetic due to the lack of a floating-point unit in most current devices. The use of integer operations in the inner loops guarantees high performance. The solution can also take advantage of the hardware acceleration used with other video processing applications. Acceleration hardware is designed to support the block-based and pixel-level processing tasks that are not efficiently handled by the CPU architecture (Kuhn 1999). Typically such hardware contains highly optimised motion estimation instructions on blocks from 16 by 16 to 4 by 4 pixels, which are also the usual sizes for the blocks in our method.

In the experiments in Paper III, the user made some typical controlling movements. In these tests, there was no ground truth available, but the trajectories followed the movements that the user made with reasonable accuracy. A frame rate of 10 fps was achieved on a Nokia 3650 smart phone. Later, frame rates of 15 fps and 30 fps were achieved using the Nokia N95 mobile phone using the front and the back camera, respectively.

### **5.1.2 Device pose as an input device**

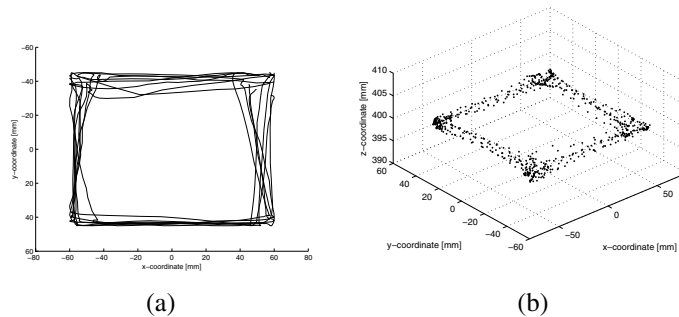
Faces and other facial features are important information sources for hand-held communication devices. The camera directed towards the user as in the model shown in Fig. 7 is usually intended for video call purposes. Therefore, the field of view of the camera is optimized for the user's facial regions. This provides advantages for various HCI solutions.

The detection of the user's face enables another novel interface technique where the absolute pose of the camera with respect to the user's face is determined. This information can be used for browsing the virtual workspace of the mobile terminal in its display. The pose is measured continuously, which means that the user interface is spatially aware of the user all the time.

Papers I and II present real-time face tracking methods to control user interfaces in

resource limited devices such as mobile phones. In Paper I, the head tracking system developed was applied to cursor control on a computer display. Experiments show that the method is capable of detecting gaze position on the display in real-time when the user makes dedicated head movements in order to control the computer.

Paper II evaluates the feasibility of the proposed method for controlling the spatially aware user interface with a camera-equipped mobile phone the Nokia N95. In order to test the repeatability of the method, a rectangle was first drawn by hand on a white board. Then, we asked a subject to track the outline of the rectangle with the device in his hand, like a pointer. The user repeated the experiment 10 times. The trajectory for the movement was obtained by solving the camera position from the estimated 3-D pose of the face. Fig. 8 (a) shows the obtained 3-D trajectories of the device as a projection in the  $x - y$  plane and Fig. 8 (b) illustrates the same result in 3-D coordinates. The trajectories are uniformly scaled and aligned with each other. As shown in the figure, the repeatability achieved for the method is reasonably good and it can enable the position based control of user interfaces.



**Fig 8. Trajectories for the repeability test: (a) trajectory in the x-y plane (b) trajectory in 3-D coordinates. Revised from [II]. © 2008 Springer.**

## 5.2 Recognising device movements

Input devices are sometimes difficult to operate on mobile systems, as there is no space for a full keyboard and mouse. Many hand-held devices use a stylus to write characters on a touch screen. This can be sometimes cumbersome, as the device and the stylus are both moving, making accurate positioning difficult. On the other hand, there has

been little use of physical gestures as an input modality for mobile devices. In the commercial domain, the Samsung SCH-S310 smart phone uses motion sensors to measure device motion and then interprets recognised movements as commands to the interface. Also, Harrison *et al.* (1998) showed that simple gestures can be used in a range of different situations. Such gestures are advantageous because users do not need to look at a display to make them.

This section presents three vision based approaches for the control of mobile devices based on motion gestures and using existing camera technology. These approaches have originally been introduced in Papers VI, VII, and VIII. In all cases, motion trajectories are obtained first and these sequences are then recognised. In the first and second application, described shortly in Sections 5.2.1 and 5.2.2, the sequences of motion features are produced by estimating the ego-motion of the device held in the user's hand. In the third application, presented briefly in Section 5.2.3, trajectories are produced by tracking the motion of the user's finger in front of the mobile phone camera.

### **5.2.1 Handwriting recognition**

The first application, originally presented in Paper VI, allows the user to write characters through a series of hand movements whilst holding the device. During these movements the device's camera is used to record the ego-motion of the camera with the method introduced in Section 4.3. The DCT is utilised for computing discriminative features from the motion trajectories and then, the  $k$ -NN rule is applied for classification. In order to show the practicality of our method, a real-time implementation was developed for a mobile phone.

The new interaction technique presented here makes writing easy and fast because only single isolated strokes are considered. The character models are similar to the Graffiti™ like alphabet used in the Palm™ devices. Single stroke characters also simplify the recognition task and make it more reliable than using ordinary characters. Moreover, the set of movements allowed can be extended to more general signs and commands used as a control input for the device. A possible shortcoming is that users have to learn a special way of writing characters or moving the device (MacKenzie & Soukoreff 2002).

In the experiments described in Paper VI, the recognition system was evaluated on a Nokia 7610 mobile phone. A total of 10 subject were asked to write 10 digits and 26 letters twice. First, the performance was tested using random selection. Then, the same

test was performed using the training set containing one sample from each test subject. In both cases recognition rates ranging from 92 % to 98 % were achieved. Despite the good results achieved, the experiments indicate that the recognition rate can still be improved through user specific adaptation.

### **5.2.2 Sign recognition**

Another device motion recognition application, originally introduced in Paper VII, is the recognition of a series of unique motion commands. Again, the sequence of motion features is produced by estimating the ego-motion of the camera, as presented in the previous section. These sequences of motion features are classified using HMMs (Rabiner 1989). The classification results are filtered using a likelihood ratio and the velocity entropy of the sequence to reject possible incorrect sequences. Our hypothesis here is that ambiguous and random sequences are characterised by a higher entropy value for their histogram than the valid trajectories.

In order to evaluate the technique described here, a hypothetical control system of mobile phone functions was devised. In this system, a series of control commands are composed of seven simple elements based on seven different motions. More complex commands are then constructed from these basic elements. In practice, these commands can correspond to, for example, *delete* in voice mail, *refresh* or *home* in a web browser, *shuffle* in MP3 player etc.

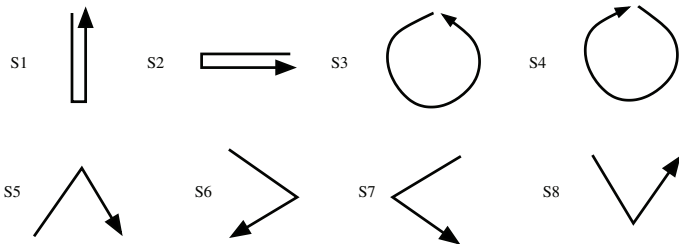
Paper VII presents experiments where the data was collected from 35 subjects. All subjects were asked to draw each command using a Nokia N90 camera equipped mobile phone. During drawing the user pressed the button for each command and there was no need for segmenting commands from captured trajectories. The result of running the system on the 570 test sequences show that finally only 5 sequences are incorrectly classified. This result confirms the good performance of the method, and justifies the use of entropy in the rejection of ambiguous sequences.

### **5.2.3 Finger gesture recognition**

In addition to the recognition of device movements, another intuitive way to interact can be achieved by moving a finger or some other object like a pen in front of the camera and then recognising the observed gestures as discussed in Paper VIII. In this case, the motion is recorded using the tracking method introduced in Section 4.4. The motion

trajectories are again recognised using HMMs. However, due to diversity in how people make the gestures it may be difficult to create general models for each class that will perform well for many different users. In order to improve the model performance we propose using unsupervised Maximum a posteriori (MAP) adaptation (Gauvain & Lee 1994) to tailor the general models for a specific user. Paper VIII demonstrates how this approach can significantly improve the performance in the task of finger gesture recognition.

Also in this system, a series of simple control commands are proposed. The eight possible commands are shown in Fig. 9. These commands are formed by the user drawing the sign in the air with an extended index finger in front of the mobile phone camera. So there are two challenges to overcome, first the tracking of the finger and secondly the classification of the motion sequences produced by tracking.



**Fig 9. The eight signs chosen to represent mobile phone commands. In the experiments each user was asked to draw the sign in the air in front of the mobile phone camera.**

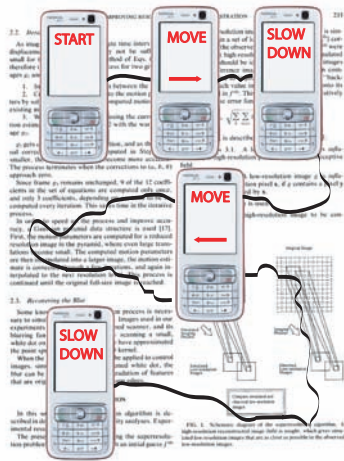
In Paper VIII, the experimental data was collected from 10 subjects using a Nokia N73 mobile phone. During drawing, the user pressed the button and therefore there was no need to segment saved trajectories afterwards. We first ran the baseline test using a training set of four subjects, a validation set of two subjects, and a test set of four different subjects. This produced a sequence recognition rate of 82 %. It was observed that errors show a distinct pattern of confusion between certain signs. This was due to the variability between different subjects when making the signs. Therefore, we tailored the model to an individual user using unsupervised MAP adaptation. In this experiment, a recognition rate of almost 94 % was achieved. It shows that by applying user specific adaptation we can significantly improve the results.

### **5.3 Interactive system for document image mosaicing**

A document image mosaic builder is essentially a camera based scanner. Instead of using devices such as flatbed scanners, the users can capture high quality images with their mobile phones. Mobile cameras enable portable and non-contact image capture of any kind of documents. Although they cannot replace flatbed scanners, they are more suitable for several scanning tasks in less constrained situations.

Paper IV presents an application where the system interactively guides the user to move the device over a newspaper page, for example, in a manner that enables a high quality image to be assembled from individual video frames. During online scanning, motion determined from low-resolution image sequences is used to control the interaction process. As a result, good high-resolution images of the document page can be captured for stitching. Images with coarse alignment information are used to construct a mosaic automatically using a feature based alignment method.

In the first stage, partial images of the document are captured with the help of user interaction. The basic idea is to apply the camera motion estimation technique, presented in Section 4.3, to the mobile phone to assist the user in the image scanning process. The user starts the scanning by taking an initial image of some part of the document. Then, the user is asked to move the device to the next location. The scanning direction is not restricted. One possible way is to use a zig-zag style scanning path shown in Fig. 10 a. The camera motion is estimated during movement and the user is informed when a suitable overlap, for example 25 %, between images is achieved. When the device motion is small enough, a new image is taken. The movement should then be stopped because otherwise the images became blurred.



(a)



(b)

**Fig 10. An example is illustrated for building a large document page image. (a) An interactive user interface helps to acquire good quality initial images. One possible scan style is zig-zag scanning. (b) A final mosaic obtained. Revised from [IX]. © 2007 IEEE.**

After online image capturing, the partial images of the document page can be stitched together. The automatic mosaicing is based on a RANSAC robust estimator (Fischler & Bolles 1981) with a SIFT feature point detector (Lowe 2004). Also, graph based global alignment and bundle adjustment steps are performed in order to minimize image registration errors and to further improve quality. Finally, warped images are blended to the mosaic using simple Gaussian weighting.

In experiments, described in Paper IX, the accuracy of the motion estimates used for computing cumulative displacements was measured using annotated data with good results. The requirement here is that the error in the estimate should not become too high, so that sufficient overlap between stored images can be guaranteed. In addition, the document mosaicing system was tested on real image sequences captured using a Nokia 6670 mobile phone. Fig. 10 b) illustrates the mosaic (1397 by 1099 pixels) constructed for an A4 document page from eight low resolution images.

## 5.4 Practical considerations

The development of the demonstration applications has contributed to the identification of features that on mobile platforms would help in implementing vision based user interfaces. We have also discovered latency and computing bottlenecks that can be removed via software and hardware developments.

### 5.4.1 Usability

As mentioned in Section 1.1, the usability of a user interface critically rests on its latency. This is most obvious with computer games in which many perceive joystick action-to-display delays exceeding about 100-150 milliseconds disturbing (Dabrowski & Munson 2001), but this applies even to key press-to-sound or display. Sometimes it is even argued that a maximum latency of 45 ms between event occurrence and system response is experienced as no delay (Sheridan & Ferrell 1963). According to Kölsch (2004), a latency of 300 ms is the threshold when interaction starts to feel sluggish. If we employ a camera as a user interface component, its integration time will add to the latency, as well as the image analysis. If we sample the scene at 15 fps rate, our base latency is 67 ms. Assuming that the integration time is 33 ms, the information in the pixels read from the camera is on average 17 ms old. Consequently, attaining computing and display latencies that limit the total latency to below 100-150 ms represents a challenge.

The typical frame rates of current mobile devices are 15 and 30 fps. At the lower frame rate less time is available, while on the other hand, the camera operated at a higher rate demands more power. An obstacle to camera based user interfaces is the turn-on time that is not only dependent on the power-up delay of the camera, but is mostly caused by software. The current multimedia frameworks intended for use on mobile platforms have substantial latencies when the resources are reconfigured for the applications.

In addition to real-time concerns, one important issue is low-light illumination conditions, where the image can be too noisy to perform any image processing. In user interfaces, special attention needs to be paid to designing a proper lighting system. One possibility is perhaps to use special infrared LEDs for illuminating the user's face. Energy consumption is, of course, the main limitation of such designs. Therefore, if cam-

eras become standard user interface components in mobile devices, energy efficiency requires that the bulk of computing is carried out using hardware acceleration.

#### **5.4.2 Platform support for the new techniques**

In this section, we list features that on mobile platforms would help in implementing computationally intensive vision algorithms. First, it should be possible to use two cameras at the same time or quickly alternate between cameras as image sources. The motivation for this capability is a practical one: sunlight, lamps, or reflections may saturate one of the cameras, so the trivial automatic adaptation method is to switch to another image source, although that may be a more power consuming high resolution device. However, the current mobile devices have single camera interfaces, and alternating between cameras requires reconfiguration that may take hundreds of milliseconds.

Second, a stand-by mode for the cameras should exist, perhaps initiated by the handling of the device recognized by built-in accelerometers, to reduce the start-up latency of the vision based user interfaces. In the stand-by mode, the camera could capture images, say, at the rate of a frame per second, adjusting to the ambient illumination. The miniature VGA camera modules used in mobile devices require about 1-1.5 mW/frame/s, a cost that needs to be weighted against the benefits gained. The cold start power-up latencies of the camera hardware modules alone are around 100 ms. At least two images are needed to determine the first motion estimates even if no gain correction is needed to bring the image information into the useful dynamic region. These plain hardware dependent delays amount to 150-200ms in total, but would be only 50-100 ms from stand-by.

Third, the data formats of the camera and graphical processing units (GPUs) should be compatible for a number of image processing functions, such as interpolations and warps. In that case, it is desirable to use the GPU as a hardware accelerator. The OpenGL ES application programming interface (API) is highly efficient, but the necessary format changes result in needless copying of data, and consequently reduced energy efficiency, increased computational burden, and latency.

Finally, motion estimation and face detection are potential platform level services to be offered via multimedia APIs. They play key roles in the demonstrated applications, and are most likely to be employed in many other applications as well. Implementing them in camera modules, which contain their own processor, would reduce the power

hungry data transfers over the system interconnects, and this can also result in lower latency.

## 5.5 Discussion

We were among the first to bring computer vision based motion estimation for controlling the user interfaces of mobile phones. Unlike other solutions, which used only lateral motion, our solution was the first to utilise also scaling and rotation. Moreover, the estimation was performed without any artificial markers. Afterwards many other similar systems have been presented in the literature. During testing it was noticed that tilting is a more natural way to control a phone than lateral movements. It was also observed that low lighting conditions were difficult for our vision based solution. Possible improvements to this situation were already discussed in Section 5.4. Recently, we extended the motion based control to spatially aware displays, which have been used in desktop computers (Fitzmaurice 1993) earlier. According to our knowledge, this was the first attempt to implement such an interface to mobile phones. Also, other similar solutions typically use other sensors than a camera.

We were also among the first to present vision based motion recognition interfaces for mobile phones. Other similar work represented about the same time was introduced by Wang *et al.* (2006). Earlier, these kinds of solutions have been experimented on using different sensors such as an accelerometer. In this thesis, we have introduced three different systems: motion based handwriting recognition, device motion recognition, and finger gesture recognition.

For motion based handwriting recognition, we introduced a new idea of using DCT to extract discriminative features for recognition. This solution is closely related to Fourier descriptors (Trier *et al.* 1996) but instead of using complex numbers we can compute features using real numbers. Another benefit is that we can also classify shapes that are not closed curves. To our knowledge, DCT has not previously been used for handwritten character recognition.

To recognise device and finger movements, we employed a standard sequence recognition solution using HMMs. From a novelty point of view, we use the velocity entropy of the sequence to filter classification results and to control the unsupervised MAP adaptation in on-line learning. Using entropy as an indicator of badly formed sequences we are able to filter out the majority of such sequences from the final result. In adaptation, we also demonstrated the improved performance of the system when using entropy.

During experiments, we observed that it was sometimes difficult for users to work with new vision based interfaces. Especially in finger tracking some subjects said that it is difficult to move their finger in an empty space without feeling the strength of their movement. Therefore, special attention should be paid to consideration of an appropriate set of possible commands. The limitation common to both finger tracking and device pose estimation was the narrow field of view of the front camera. This camera is designed for video telephony to display the user's face. However, for our purposes this property limited the magnitude of possible movements.

We were the first to introduce an interactive document panorama builder on mobile phones. Other similar ideas have been presented for different platforms earlier. Nakao *et al.* (1998) presented a method where a camera was attached to the mouse. Zappala *et al.* (1999) used an over-the-desk camera to take images of a document that is moved on the desk. Recently, Liang *et al.* (2006) proposed a method for camera captured document images. The main contribution of our system is the use of a motion estimation framework to interactively guide the user to capture images of the document to be scanned.

## 5.6 Summary

This chapter presented the use of computer vision techniques developed in new mobile interaction solutions. In the cases described, the cameras of mobile devices are employed as motion sensors in user interfaces. Object or device motion information extracted from the image sequence helps in implementing new interaction systems, and provides a self-intuitive means of interacting with a small screen and minimal keypad.

The proposed solutions, head tracking, object motion analysis, and extraction of device ego-motion from sequential video frames, clearly have potential to become a practical means for interacting with mobile devices, and they can augment the information provided by accelerometers and touch screens in a complementary manner. For example, face or head detection and tracking from images is an exceptionally powerful user interface component, as the user is almost invariably looking at the device.

Although, the speed and accuracy of the algorithms are mature enough for real applications, some problems in mobile use still remain. Energy efficiency is a significant challenge in exploiting camera based user interface concepts, but in our judgement a solvable one. Camera sub-systems on mobile device platforms are a rather recent addition, and designed simply for capturing still and video frames. At the same time the

energy efficiency features of the platform architectures, computing resources, and displays have been optimized for video playback. From the demonstrated applications point of view, lower camera start-up latencies would improve the usability, but require careful balancing with energy efficiency demands. In addition, low lighting conditions need some further research. The use of IR illumination is one solution to be considered in the future.

## 6 Discussion

In 2003, when this work was started there were only few studies related to vision based HCI on hand-held mobile devices. Today, the situation is totally different. There is increasing interest among researchers in mobile user interfaces. During the past five years, we have seen the multimedia capabilities of mobile phones advance enormously. Mobile phones with high-resolution digital cameras are now inexpensive, widely available, and very popular. The rapid evolution of image sensors and computing hardware on mobile phones has made it convenient to apply computer vision methods to create new user interface techniques and concepts for future HCI.

This thesis attempts to respond to the challenge of bringing these solutions closer to real user applications. It proposed new methods of using a camera as a sensor for controlling user interfaces of mobile phones. In fact, we were among the first to explore this new field of research when this work was started. From the novelty point of view, in this thesis several new vision based interaction solutions were proposed. In Chapters 2 and 3, two solutions suitable for mobile HCI were presented using facial feature tracking and head pose estimation, and their efficiency was demonstrated in the user interface control. Although a number of other solutions have been proposed in the literature, only a few of them have addressed this problem on mobile phones (for example Venkataramani *et al.* (2005), Hansen *et al.* (2006)) so far. The contributions presented in Chapter 4 were the computationally efficient motion estimation framework and the multiple motion tracking approach. Our framework has been mentioned as one of the state-of-the-art methods in motion estimation on mobile phones (Capin *et al.* 2008). Moreover, the framework was recently added to the Nokia computer vision library (NokiaCV), which provides many image processing algorithms for mobile application developers. The multiple motion tracking method was developed for the finger tracking application, but it can also be used for other purposes. In Chapter 5, the methods developed have been applied in novel user interface solutions. Although similar concepts have been presented earlier in the literature, the contributions of our work was to bring these solutions to mobile devices.

One can always ask how useful the developed solutions are in practise when controlling the user interface of the mobile phone. The vision based interaction techniques presented are unlikely to completely replace traditional interaction devices. Keyboards

and mice are probably going to be the preferred user input devices for many user interfaces and applications. The limitations of computer vision must be kept in mind when looking at possible applications. While the methods presented were applied successfully in our laboratory experiments, there are some situations that cannot be handled by vision algorithms, as discussed in previous chapters.

In this thesis, we have focused on computer vision based interaction. However, the addition of other modalities is certainly going to be beneficial for many aspects of usability and flexibility. Combining information from different sensors such as accelerometers and touch screens can enable more powerful and innovative interaction techniques. To be fair, accelerometers can be even more relevant technology for sensing hand-held device motion in certain cases. On the other hand, computer vision can provide advantages if positional information is needed instead of velocity or acceleration. We believe that together with cameras, the use of other sensors can further improve and enrich the user experience and performance in many tasks.

From the hardware viewpoint, it is clear that the CPU's clock frequency and the amount of memory will not limit the applicability of computer vision based interfaces on mobile phones in the future. Processing performance is certain to increase thanks to the rapid evolution of mobile technology. For example, the first phone, the Nokia 3650, that we used for our experiments in 2003 included a 104 MHz CPU, 3.4 MB shared memory, and 0.3 Megapixel camera. Whereas the Nokia N95 used in 2008 contained a 332 MHz CPU, 64 MB RAM, and 5 Megapixel camera. This device included also a graphics hardware accelerator (GPU), an accelerometer and a low-resolution front camera. These new computing resources and sensors in a mobile phone open up new possibilities for developing user applications. For instance, computer vision algorithms can be mapped to work on a GPU, which releases computing resources for other purposes. In Chapter 5 we also discussed other features that on mobile platforms would help in implementing vision based user interfaces. Hopefully, the items on the wish list created come true in the future.

In this thesis, we demonstrated the use of a camera as an alternative user interface input device in some applications. We believe that many more vision based applications will follow. The user interface in mobile phones is a segment where new innovations are needed at every level. It is just matter of time before the built-in cameras of mobile phones are also widely used as interaction devices.

## 7 Conclusions

Control of hand-held mobile devices can be sometimes difficult, given the small size of the equipment. In the systems, traditional input devices, such as keyboards, or screen based interfaces operated with a mouse, are not generally practical. With these constraints, the giving of controlling commands to the device is sometimes cumbersome. Considering the prevalence of mobile devices equipped with cameras and high processing capabilities, computer vision based techniques are an attractive option to improve and enrich interaction with hand-held devices.

The main contribution of this thesis is the development of computer vision approaches to allow efficient interaction with mobile devices. The proposed techniques include facial feature based head tracking, object motion analysis, and device ego-motion estimation. The key ideas of new approaches presented mainly rest on the utilisation of the hand-held nature of the equipment. The real-time applicability of the methods was enhanced via fast software implementations. Experiments with synthetic and real image sequences clearly indicate that the methods presented can be applied for user interface purposes on platforms with limited computational resources.

The usage potential of the methods developed is demonstrated in three new mobile interface solutions. One of them estimates the ego-motion of the device with respect to the user's face, and utilises that information for browsing large documents or bitmaps on small displays. The second solution is to use device or finger motion to recognize simple gestures. In addition to these applications, a novel interactive system to build document panorama images is introduced. The demonstrations presented are by no means the only ways to apply vision to mobile user interfaces, and one may find new interesting possibilities in further research.

While the methods presented in this thesis have been successfully used in various experiments, more research is still needed to implement vision based solutions for consumer-grade products. Although the speed and accuracy of the algorithms are mature enough for real applications, the robustness in mobile use under low light conditions is still a problem. One solution to solve this issue might be the use of IR illumination. In addition, one must consider energy efficiency, and therefore some platform support is needed to use a camera as a sensor for the user interface. Furthermore, sensors such as accelerometers are steadily making their way into mobile devices. Such

sensors might be used to enhance image motion estimation in order to improve the robustness and to reduce computational power needed by computer vision algorithms.

We conclude that the computer vision based motion estimation and recognition techniques presented in this thesis have clear potential to become practical means for interacting with mobile devices. They can possibly also augment the information provided by other sensors, such as accelerometers and touch screens, in a complementary manner. In fact, the cameras in future mobile devices may, for most of time, be used as sensors for self intuitive user interfaces rather than using them for digital photography.

## References

- Adams A, Gelfand N & Pulli K (2008) Viewfinder alignment. Proc. Eurographics 2008, 597–606.
- Aggarwal JK & Nandhakumar N (1988) On the computation of motion from sequences of images - a review. Proceedings of the IEEE 76(8): 917–935.
- Anandan P (1989) A computational framework and an algorithm for the measurement of visual motion. International Journal of Computer Vision 2(3): 283–310.
- Aschwanen P & Guggenbühl W (1992) Experimental results from a comparative study of correlation-type registration algorithms. In: Förstner W & Ruwiedel S (eds) Robust Computer Vision: Quality of Vision Algorithms, 268–289. Wichmann, Karlsruhe.
- Azarbayejani A, Starner T, Horowitz B & Pentland A (1993) Visually controlled graphics. IEEE Transactions on Pattern Analysis and Machine Intelligence 15(6): 602–605.
- Bala LP, Talmi K & Liu J (1997) Automatic detection and tracking of faces and facial features in video sequences. Proc. Picture Coding Symposium, Berlin, Germany, 251–256.
- Ballard P & Stockman G (1995) Controlling a computer via facial aspect. IEEE Transactions on Systems, Man, and Cybernetics 25(4): 669–677.
- Barron JL, Fleet DJ & Beauchemin SS (1994) Performance of optical flow techniques. International Journal of Computer Vision 12(1): 43–77.
- Basu S, Essa I & Pentland A (1996) Motion regularization for model-based head tracking. Proc. the 13th IEEE International Conference on Pattern Recognition, 611–616.
- Birchfield S (1999) Elliptical head tracking using intensity gradients and color histograms. Proc. IEEE Conference on Computer Vision and Pattern Recognition, 232–237.
- Black MJ & Anandan P (1996) The robust estimation of multiple motions: Affine and piecewise-smooth flow fields. Computer Vision and Image Understanding 63(1): 75–104.
- Black MJ & Yacoob Y (1995) Tracking and recognizing rigid and non-rigid facial motions using local parametric models of image motion. Proc. IEEE International Conference on Computer Vision, 374–381.
- Black MJ, Yacoob Y & Ju SX (1997) Motion-based recognition. In: Shah M & Jain R (eds) Computational Imaging and Vision, volume 9, 245–269. Kluwer Academic Publishers.
- Bober M & Kittler J (1994) Robust motion analysis. Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 947–952.
- Bradski G (1998) Real time face and object tracking as a component of a perceptual user interface. Proc. IEEE Workshop on the Applications of Computer Vision, 214–219.
- Brox T, Rosenhahn B, Cremers D & Seidel HP (2006) High accuracy optical flow serves 3-d pose tracking: Exploiting contour and flow based constraints. Proc. 9th European Conference on Computer Vision, 98–111.
- Brunelli R & Poggio T (1993) Face recognition: Features versus templates. IEEE Transactions on Pattern Analysis and Machine Intelligence 15(10): 1042–1052.
- Burt PJ & Adelson EH (1983) The Laplacian pyramid as a compact image code. IEEE Transactions on Communication 31(4): 532–540.
- Burt PJ, Hingorani R & Kolczynski RJ (1991) Mechanisms for isolating component patterns in the sequential analysis of multiple motion. Proc. IEEE Workshop on Visual Motion, 187–193.
- Camus T (1997) Real-time quantized optical flow. Real-Time Imaging 3(2): 71–86.

- Capin T, Pulli K & Akenine-Möller T (2008) The state of the art in mobile graphics research. *IEEE Computer Graphics and Applications* 74–84.
- Cascia ML, Sclaroff S & Athitsos V (2000) Fast, reliable head tracking under varying illumination: An approach based on registration of texture-mapped 3d models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22(4): 322–336.
- Comaniciu D, Ramesh V & Meer P (2000) Real-time tracking of non-rigid objects using mean shift. *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2: 142–149.
- Cootes T & Taylor C (1992) Active shape models - smart snakes. *Proc. British Machine Vision Conference*, 266–275.
- Dabrowski J & Munson E (2001) Is 100 milliseconds too fast? *Proc. Conference on Human Factors in Computing Systems*, 317–318.
- Davison AJ, Reid ID, Molton ND & Stasse O (2007) Monoslam: Real-time single camera slam. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29: 2007.
- DeCarlo D & Metaxas D (1996) The integration of optical flow and deformable models with applications to human face shape and motion estimation. *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 231–238.
- Dempster A, Laird N & Rubin D (1977) Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society* 39(1): 1–38.
- DiVerdi S & Höllerer T (2007) Groundcam: A tracking modality for mobile mixed reality. *Proc. IEEE Virtual Reality*, 75–82.
- Dornaika F & Davoine F (2006) On appearance based face and facial action tracking. *IEEE Transactions on Circuits and Systems and Video Technology* 16(9): 1107–1124.
- Dornaika F & Orozco J (2007) Real time 3d face and facial feature tracking. *Real-time image processing* 2(1): 35–44.
- Drab SA & Artner NM (2005) Motion detection as interaction technique for games & applications on mobile devices. *Proc. Pervasive Mobile Interaction Devices*, 52–55.
- Eslambolchilar P & Murray-Smith R (2004) Tilt-based automatic zooming and scaling in mobile devices - a state-space implementation. *Proc. Mobile Human-Computer Interaction*, 120–131.
- Fischler MA & Bolles RC (1981) Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM: Graphics and Image Processing* 24(6): 381–395.
- Fitzmaurice GW (1993) Situated information spaces and spatially aware palmtop computers. *Communications of the ACM* 36(7): 38–49.
- Fleet DJ & Jepson AD (1990) Computation of component image velocity from local phase information. *International Journal of Computer Vision* 5(1): 77–104.
- Fortmann TE, Bar-Shalom Y & Scheffe M (1983) Sonar tracking of multiple targets using joint probabilistic data association. *IEEE Journal of Oceanic Engineering* 8(3): 173–184.
- Freeman W, Beardsley P, Kage H, Tanaka K, Kyuma K & Weissman C (2000) Computer vision for computer interaction. *ACM SIGGRAPH Computer Graphics* 33(4): 65–68.
- Freund Y & Schapire R (1995) A decision-theoretic generalization of on-line learning and an application to boosting. *Proc. European Conference on Computational Learning Theory*, 23–37.
- Gauvain JL & Lee CH (1994) Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains. *IEEE Transactions on Speech Audio Processing* 2: 291–298.
- Gonzalez RC & Woods RE (2002) *Digital Image Processing*. Prentice Hall, New Jersey.

- Graf H, Cosatto E, Gibbon D, Kocheisen M & Petajan E (1996) Multimodal system for locating heads and faces. Proc. the Second IEEE International Conference on Automatic Face and Gesture Recognition, 88–93.
- Hachet M, Pouderoux J & Guitton P (2005) A camera-based interface for interaction with mobile handheld computers. Proc. I3D'05 - ACM SIGGRAPH 2005 Symposium on Interactive 3D Graphics and Games, ACM Press, 65–71.
- Hadid A, Pietikäinen M & Ahonen T (2004) A discriminative feature space for detecting and recognizing faces. Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2: 797–804.
- Hannuksela J, Sangi P & Heikkilä J (2005) A vision-based approach for controlling user interfaces of mobile devices. Proc. IEEE Conference on Computer Vision and Pattern Recognition, Workshop on Vision for Human-Computer Interaction, San Diego, CA, USA. 6 p.
- Hansen T, Eriksson E & Lykke-Olesen A (2006) Use your head: exploring face tracking for mobile interaction. Proc. CHI '06 extended abstracts on Human factors in computing systems, ACM Press, New York, NY, USA, 845–850.
- Hansen TR, Eriksson E & Lykke-Olesen A (2005) Mixed interaction space - designing for camera based interaction with mobile devices. Proc. CHI '05 extended abstracts on Human factors in computing systems, 1933–1936.
- Haralick RM & Shapiro LG (1993) Computer and Robot Vision, vol. 2. Addison-Wesley, Reading MA.
- Haro A, Mori K, Capin T & Wilkinson S (2005) Mobile camera-based user interaction. Proc. IEEE International Conference on Computer Vision, Workshop on Human-Computer Interaction, Beijing, China, 79–89.
- Harris C & Stephen M (1988) A combined corner and edge detection. Proc. ALVEY Vision Conference, 147–151.
- Harrison BL, Fishkin KP, Gujar A, Mochon C & Want R (1998) Squeeze me, hold me, tilt me! an exploration of manipulative user interfaces. Proc. CHI '98: Proceedings of the SIGCHI conference on Human factors in computing systems, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 17–24.
- Hartley RI & Zisserman A (2004) Multiple View Geometry in Computer Vision. Cambridge University Press, second edition.
- Haußecker H & Spies H (1999) Motion. In: Jähne B, Haußecker H & Geißler P (eds) Handbook of Computer Vision and Applications, Volume 2, chapter 13, 309–396. Academic Press, San Diego.
- Heeger DJ (1988) Optical flow using spatio-temporal filters. International Journal of Computer Vision 1: 179–302.
- Herpers R, Michaelis M, Lichtenauera KH & Sommer G (1996) Edge keypoint detection in facial regions. Proc. Proceedings of the Second International Conference on Automatic Face and Gesture Recognition, 212–217.
- Hinckley K, Pierce JS, Sinclair M & Horvitz E (2000) Sensing techniques for mobile interaction. Proc. 13th annual ACM symposium on User Interface Software and Technology, 91–100.
- Hjelmås E & Low BK (2001) Face detection: A survey. Computer Vision and Image Understanding 83: 236–274.
- Horn BKP & Schunck BG (1981) Determining optical flow. Artificial Intelligence 17(1-3): 185–203.
- Hough PVC (1959) Machine analysis of bubble chamber pictures. Proc. International Conference

- on High Energy Accelerators and Instrumentation.
- Hua G, Yang TY & Vasireddy S (2007) Pey: Toward a visual motion based perceptual interface for mobile devices. Proc. Lew MS, Sebe N, Huang TS & Bakker EM (eds) IEEE International Conference on Computer Vision, Workshop on Human-Computer Interaction, LNCS 4796, Springer, 39–48.
- Huang C, Ai H, Li Y & Lao S (2005) Vector boosting for rotation invariant multi-view face detection. Proc. The Tenth IEEE International Conference on Computer Vision Volume 1, IEEE Computer Society, Washington, DC, USA, 446–453.
- Hwang J, Jung J & Kim GJ (2006) Hand-held virtual reality: A feasibility study. Proc. ACM Virtual Reality Software and Technology, 356–363.
- Irani M, Rousso B & Peleg S (1994a) Computation of egomotion using image stabilization. Proc. Computer Vision and Pattern Recognition, 454–460.
- Irani M, Rousso B & Peleg S (1994b) Computing occluding and transparent motions. International Journal of Computer Vision 12(1): 5–16.
- Isard M & Blake A (1996) Contour tracking by stochastic propagation of conditional density. Proc. European Conference on Computer Vision, 1: 343–356.
- Jaimes A & Sebe N (2007) Multimodal human-computer interaction: a survey. Computer Vision and Image Understanding 108(1-2): 116–134.
- Jebara T, Azarbayejani A & Pentland A (1999) 3d structure from 2d motion. IEEE Signal Processing Magazine 16: 66–84.
- Jebara TS & Pentland A (1997) Parametrized structure from motion for 3d adaptive feedback tracking of faces. Proc. IEEE Conference on Computer Vision and Pattern Recognition, 144–150.
- Jeng SH, Liao HYM, Han CC, Chern MY & Liu YT (1998) Facial feature detection using geometrical face model: An efficient approach. Pattern Recognition 31(3): 273–282.
- Jones M & Viola P (2003) Fast multi-view face detection. Technical Report TR2003-096, Mitsubishi Electric Research Laboratories.
- Kalman RE (1960) A new approach to linear filtering and prediction problems. Transactions of the ASME-Journal of Basic Engineering Series D(82): 35–45.
- Kass M, Witkin A & Terzopoulos D (1988) Snakes: Active contour models. International Journal of Computer Vision 1(4): 321–331.
- Kölsch M (2004) Vision based hand gesture interfaces for wearable computing and virtual environments. Ph.D. thesis, University of California, Santa Barbara.
- Kuhn PM (1999) Algorithms, Complexity Analysis and VLSI Architectures for MPEG-4 Motion Estimation. Kluwer Academic Publishers, Norwell, MA, USA.
- Lee CH, Kim JS & Park KH (1996) Automatic face location in a complex background using motion and color information. Pattern Recognition 29(11): 1877–1889.
- Lepetit V & Fua P (2005) Monocular model-based 3d tracking of rigid objects: A survey. Foundations and Trends in Computer Graphics and Vision 1(1): 1–89.
- Liang J, DeMenthon D & Doermann D (2006) Camera-based document image mosaicing. Proc. International Conference on Pattern Recognition, 476–479.
- Licklider JCR (1960) Man-computer symbiosis. IRE Transactions on Human Factors in Electronics HFE-1: 4–11.
- Lienhart R & Maydt J (2002) An extended set of haar-like features for rapid object detection. Proc. IEEE International Conference on Image Processing, 1: 900–903.
- Linnainmaa S, Harwood D & Davis LS (1988) Pose determination of a three-dimensional object

- using triangle pairs. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 10(5): 634–647.
- Liu H, Hong TH, Herman M, Camus T & Chellappa R (1998) Accuracy vs efficiency trade-offs in optical flow algorithms. *Computer Vision and Image Understanding* 72(3): 271–286.
- Liu X, Doermann D & Li H (2005) Fast camera motion estimation for hand-held devices and applications. *Proc. 4th International Conference on Mobile and Ubiquitous Multimedia*, 103–108.
- Lowe D (2004) Distinctive image feature from scale-invariant keypoints. *International Journal of Computer Vision* 60(2): 91–110.
- Loy G & Zelinsky A (2003) Fast radial symmetry for detecting points of interest. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25(8): 959–973.
- Lucas BD & Kanade T (1981) An iterative image registration technique with an application to stereo vision. *Proc. 7th International Joint Conference on Artificial Intelligence*, 674–679.
- MacCormick J & Isard M (2000) Partitioned sampling, articulated objects, and interface-quality hand tracking. *Proc. European Conference on Computer Vision*, 3–19.
- MacKenzie IS & Soukoreff RW (2002) Text entry for mobile computing: Models and methods, theory and practice. *Human-Computer Interaction* 17: 147–198.
- Martinkauppi B, Soriano M & Pietikäinen M (2003) Detection of skin color under changing illumination: A comparative study. *Proc. 12th International Conference on Image Analysis and Processing*, 652–657. Mantova, Italy.
- McCane B, Novins K, Crannitch D & Galvin B (2001) On benchmarking optical flow. *Computer Vision and Image Understanding* 84(1): 126–143.
- Mendel JM (1995) *Lessons in Estimation Theory for Signal Processing, Communications and Control*. Prentice Hall, Englewood Cliffs, New Jersey.
- Mikolajczyk K & Schmid C (2004) Scale & affine invariant interest point detectors. *International Journal of Computer Vision* 60(1): 63–86.
- Möhring M, Lessig C & Bimber O (2004) Optical tracking and video see-through ar on consumer cell phones. *Proc. Workshop on Virtual and Augmented Reality of the GI-Fachgruppe AR/VR*, 193–204.
- Murphy KP (1998) Switching kalman filters. Technical Report Tech Report 98-10, Compaq Cambridge Research Lab.
- Myers BA (1998) A brief history of human-computer interaction technology. *ACM interactions* 5(2): 44–54.
- Nakao T, Kashitani A & Kaneyoshi A (1998) Scanning a document with a small camera attached to a mouse. *Proc. WACV'98*, 63–68.
- Nickels K & Hutchinson S (2002) Estimating uncertainty in ssd-based feature tracking. *Image and Vision Computing* 20: 47–58.
- Odobez JM & Bouthemy P (1995) Robust multiresolution estimation of parametric motion models. *Journal of Visual Communication and Image Representation* 6(4): 348–365.
- Ojala T, Pietikäinen M & Mäenpää T (2002) Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24(7): 971–987.
- Osadchy M, Cun YL & Miller ML (2007) Synergistic face detection and pose estimation with energy-based models. *The Journal of Machine Learning Research* 8: 1197–1215.
- Pantic M & Rothkrantz LJM (2000) Automatic analysis of facial expressions: The state of the art. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22(12): 1424–1445.

- Patras I, Hendriks EA & Lagendijk RL (2002) Confidence measures for block matching motion estimation. *Proc. International Conference on Image Processing*, 277–280.
- Pears N, Olivier P & Jackson D (2008) Display registration for device interaction. *Proc. 3rd International Conference on Computer Vision Theory and Applications*, 446–451.
- Pentland A (2000) Looking at people: Sensing for ubiquitous and wearable computing. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22(1): 107–119.
- Rabiner LR (1989) A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proceedings of the IEEE* 77(2): 257–286.
- Rao B (1993) Data association methods for tracking systems. MIT Press, Cambridge, MA, USA.
- Reid DB (1979) An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control* AC-24(6): 843–854.
- Reisfeld D & Yeshurun Y (1992) Robust detection of facial features by generalized symmetry. *Proc. Proceedings of the 11th International Conference on Pattern Recognition*, 117–120.
- Rekimoto J (1996) Tilting operations for small screen interfaces. *Proc. 9th annual ACM symposium on User interface software and technology*, ACM Press, 167–168.
- Rohs M (2004) Real-world interaction with camera-phones. *Proc. 2nd International Symposium on Ubiquitous Computing Systems*, 39–48.
- Rowley H, Baluja S & Kanade T (1998) Rotation invariant neural network-based face detection. *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 38–44.
- Sangi P, Heikkilä J & Silvén O (2004) Motion analysis using frame differences with spatial gradient measures. *Proc. 17th International Conference on Pattern Recognition*, Cambridge, UK, 4: 733–736.
- Sawhney HS & Ayer S (1996) Compact representations of videos through dominant and multiple motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 18(8): 814–830.
- Schneiderman H (2004) Learning a restricted bayesian network for object detection. *Proc. IEEE Conference on Computer Vision and Pattern Recognition*.
- Schneiderman H & Kanade T (2000) A statistical method for 3d object detection applied to faces and cars. *Proc. IEEE Conference on Computer Vision and Pattern Recognition*.
- Sheridan TB & Ferrell WR (1963) Remote manipulative control with transmission delay. *IEEE Transactions on Human Factors in Electronics* HFE-4: 25–29.
- Shi J & Tomasi C (1994) Good features to track. *Proc. Computer Vision and Pattern Recognition*, 593–600.
- Simoncelli EP, Adelson EH & Heeger DJ (1991) Probability distributions of optical flow. *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 310–315.
- Singh A & Allen P (1992) Image-flow computation: an estimation-theoretic framework and a unified perspective. *Computer Vision, Graphics and Image Processing* 56(2): 152–177.
- Sirohey SA & Rosenfeld A (2001) Eye detection in a face image using linear and nonlinear filters. *Pattern Recognition* 34: 1367–1391.
- Sobottka K & Pitas I (1998) A novel method for automatic face segmentation, facial feature extraction and tracking. *Signal Processing: Image Communication* 12(3): 263–281.
- Stiller C & Konrad J (1999) Estimating motion in image sequences, a tutorial on modeling and computation of 2d motion. *IEEE Signal Processing Magazine* 16: 70–91.
- Ström J, Jebara T, Basu S & Pentland A (1999) Real time tracking and modeling of faces: An ekf-based analysis by synthesis approach. *Proc. International Conference on Computer Vision: Workshop on Modeling People*, 55–61. Kerkyra, Greece.

- Sung KK & Poggio T (1998) Example-based learning for view-based human face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20(1): 39–51.
- Tomasi C & Kanade T (1991) Detection and tracking of point features. Technical Report CMU-CS-91-132, Carnegie-Mellon University.
- Tommasini T, Fusiello A, Trucco E & Roberto V (1998) Making good features track better. *Proc. Computer Vision and Pattern Recognition*, 178–183.
- Torr PHS & Zisserman A (1999) Feature based methods for structure and motion estimation. *Proc. International Workshop on Vision Algorithms*, 278–295.
- Toyama K & Hager G (1999) Incremental focus of attention for robust vision-based tracking. *International Journal of Computer Vision* 35(1): 45–63.
- Trier OD, Jain AK & Taxt T (1996) Feature extraction methods for character recognition - a survey. *Pattern Recognition* 29(4): 641–662.
- Turk M (2004) Computer vision in the interface. *Communications of the ACM* 47(1): 60–67.
- Turk M (2005) Rtv4hci: A historical overview. In: Kisacanin B, Pavlovic V & Huang T (eds) *Real-Time Vision for Human-Computer Interaction*, 3–13. Springer.
- Vacchetti L, Lepetit V & Fua P (2004) Stable real-time 3d tracking using online and offline information. *IEEE Transactions Pattern Analysis and Machine Intelligence* 26(10): 1385–1391.
- Venkataramani K, Qidwai S & Kumar BVKV (2005) Face authentication from cell phone camera images with illumination and temporal variations. *IEEE Transactions on Systems, Man, and Cybernetics, Part C* 35(3): 411–418.
- Verri A & Poggio T (1989) Motion field and optical flow: Qualitative properties. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 11(5): 490–498.
- Viola P & Jones M (2001) Rapid object detection using a boosted cascade of simple features. *Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, 511–518.
- Wang J, Zhai S & Canny J (2006) Camera phone based motion sensing: interaction techniques, applications and performance study. *Proc. 19th annual ACM symposium on User interface software and technology*, 101–110.
- Wang P & Ji Q (2007) Multi-view face and eye detection using discriminant features. *Computer Vision and Image Understanding* 105: 99–111.
- Wang Y, Zhang Y & Ostermann J (2001) *Video Processing and Communications*. Prentice Hall PTR, Upper Saddle River, NJ, USA.
- Winkler S, Rangaswamy K & Zhou Z (2007) Intuitive map navigation on mobile devices. In: Stephanidis C (ed) *4th International Conference on Universal Access in Human-Computer Interaction, Part II, HCI International 2007, LNCS 4555*, 605–614. Springer, Beijing, China.
- Wong KW, Lam KM & Siu WC (2003) A robust scheme for live detection of human faces in color images. *Signal Processing: Image Communication* 18: 103–114.
- Xiao J, Moriyama T, Kanade T & Cohn J (2003) Robust full motion recovery of head for facial expression analysis. *International Journal of Imaging Systems and Technology* 13: 85–94.
- Xiong G, Feng C & Ji L (2006) Dynamical gaussian mixture model for tracking elliptical living objects. *Pattern Recognition Letters* 27(7): 838–842.
- Yang J, Stiefelhagen R, Meier U & Waibel A (1998) Real-time face and facial feature tracking and applications. *Proc. Auditory-Visual Speech Processing, Terrigal Australia*, 79–84.
- Yang MH & Ahuja N (1998) Detecting human faces in color images. *Proc. International Conference on Image Processing*, 1: 127–130.
- Yang MH, Kriegman D & Ahuja N (2002) Detecting faces in images: A survey. *IEEE Transac-*

- tions on Pattern Analysis and Machine Intelligence 24(1): 34–58.
- Yang T, Li SZ, Pan Q, Li J & Zhao C (2006) Reliable and fast tracking of faces under varying pose. Proc. 7th International Conference on Automatic Face and Gesture Recognition, IEEE Computer Society, Washington, DC, USA, 421–428.
- Yee KP (2003) Peephole displays: pen interaction on spatially aware handheld computers. Proc. SIGCHI conference on human factors in computing systems, 1–8.
- Yow KC & Cipolla R (1997) Feature-based human face detection. Image and Vision Computing 15(9): 713–735.
- Yuille A, Hallinan P & Cohen D (1992) Feature extraction from faces using deformable templates. International Journal of Computer Vision 8: 99–111.
- Zappala A, Gee A & Taylor MJ (1999) Document mosaicing. Image and Vision Computing 17: 585–595.
- Zhang ZQ, Zhu L, Li SZ & Zhang HJ (2002) Real-time multi-view face detection. Proc. Fifth IEEE International Conference on Automatic Face and Gesture Recognition, 142–147.
- Zhao W, Chellappa R, Phillips PJ & Rosenfeld A (2003) Face recognition: A literature survey. ACM Computing Surveys 34(4): 399–458.
- Zheng Q & Chellappa R (1993) A computational vision approach to image registration. IEEE Transactions on Image Processing 2: 311–326.

## Original articles

- I Hannuksela J, Heikkilä J & Pietikäinen M (2004) A real-time facial feature based head tracker. Proc. 6th International Conference on Advanced Concepts for Intelligent Vision Systems. Brussels, Belgium: 267–272.
- II Hannuksela J, Sangi P, Turtinen M & Heikkilä J (2008) Face tracking for spatially aware mobile user interfaces. Proc. International Conference on Image and Signal Processing. Cherbourg-Octeville, Normandy, France. Lecture Notes in Computer Science 5099: 405–412.
- III Hannuksela J, Sangi P & Heikkilä J (2007) Vision-based motion estimation for interaction with mobile devices. Computer Vision and Image Understanding 108(1–2): 188–195.
- IV Sangi P, Hannuksela J & Heikkilä J (2007) Global motion estimation using block matching with uncertainty analysis. Proc. 15th European Signal Processing Conference, Poznan, Poland: 1823–1827.
- V Hannuksela J, Huttunen S, Sangi P & Heikkilä J (2007) Motion-based finger tracking for user interaction with mobile devices. Proc. 4th European Conference on Visual Media Production. London, UK.
- VI Hannuksela J, Sangi P & Heikkilä J (2006) Motion-based handwriting recognition for mobile interaction. Proc. 18th International Conference on Pattern Recognition. Hong Kong, China 4: 397–400.
- VII Barnard M, Hannuksela J, Sangi P & Heikkilä J (2007) A vision based motion interface for mobile phones. Proc. 5th International Conference on Computer Vision Systems. Bielefeld, Germany: 1-10.
- VIII Hannuksela J, Barnard M, Sangi P & Heikkilä J (2008) Adaptive motion-based gesture recognition interface for mobile phones. Proc. 6th International Conference on Computer Vision Systems. Santorini, Greece. Lecture Notes in Computer Science 5008: 271–280.
- IX Hannuksela J, Sangi P, Heikkilä J, Liu X & Doermann D (2007) Document image mosaicing with mobile phones. Proc. 14th International Conference on Image Analysis and Processing, Modena, Italy: 575–580.

Reprinted with permission from Springer Science and Business Media (II, VIII), Elsevier (III), EURASIP (IV), IET (V) and IEEE (VI, IX).

Original publications are not included in the electronic version of the dissertation.



296. Tölli, Antti (2008) Resource management in cooperative MIMO-OFDM cellular systems
297. Karkkila, Harri (2008) Consumer pre-purchase decision taxonomy
298. Rabbachin, Alberto (2008) Low complexity UWB receivers with ranging capabilities
299. Kunnari, Esa (2008) Multirate MC-CDMA. Performance analysis in stochastically modeled correlated fading channels, with an application to OFDM-UWB
300. Särkkä, Jussi (2008) A novel method for hazard rate estimates of the second level interconnections in infrastructure electronics
301. Mäkelä, Juha-Pekka (2008) Effects of handoff algorithms on the performance of multimedia wireless networks
302. Teräs, Jukka (2008) Regional science-based clusters. A case study of three European concentrations
303. Lahti, Markku (2008) Gravure offset printing for fabrication of electronic devices and integrated components in LTCC modules
304. Popov, Alexey (2008) TiO<sub>2</sub> nanoparticles as UV protectors in skin
305. Illikainen, Mirja (2008) Mechanisms of thermomechanical pulp refining
306. Borkowski, Maciej (2008) Digital  $\Delta$ - $\Sigma$  Modulation. Variable modulus and tonal behaviour in a fixed-point digital environment
307. Kuismanen, Kimmo (2008) Climate-conscious architecture—design and wind testing method for climates in change
308. Kangasvieri, Tero (2008) Surface-mountable LTCC-SiP module approach for reliable RF and millimetre-wave packaging
309. Metsärinta, Maija-Leena (2008) Sinkkivälkkeen leijukerrosasuituksen stabiilisuus
310. Prokkola, Jarmo (2008) Enhancing the performance of ad hoc networking by lower layer design
311. Löytynoja, Mikko (2008) Digital rights management of audio distribution in mobile networks
312. El Harouny, Elisa (2008) Historiallinen puukaupunki suojelukohteena ja elinympäristönä. Esimerkkeinä Vanha Porvoo ja Vanha Raahe. Osa 1
312. El Harouny, Elisa (2008) Historiallinen puukaupunki suojelukohteena ja elinympäristönä. Esimerkkeinä Vanha Porvoo ja Vanha Raahe. Osa 2

Book orders:  
OULU UNIVERSITY PRESS  
P.O. Box 8200, FI-90014  
University of Oulu, Finland

Distributed by  
OULU UNIVERSITY LIBRARY  
P.O. Box 7500, FI-90014  
University of Oulu, Finland

S E R I E S E D I T O R S

**A**  
**SCIENTIAE RERUM NATURALIUM**

*Professor Mikko Siponen*

**B**  
**HUMANIORA**

*University Lecturer Elise Kärkkäinen*

**C**  
**TECHNICA**

*Professor Hannu Heusala*

**D**  
**MEDICA**

*Professor Olli Vuolteenaho*

**E**  
**SCIENTIAE RERUM SOCIALIUM**

*Senior Researcher Eila Estola*

**F**  
**SCRIPTA ACADEMICA**

*Information officer Tiina Pistokoski*

**G**  
**OECONOMICA**

*University Lecturer Seppo Eriksson*

**EDITOR IN CHIEF**

*Professor Olli Vuolteenaho*

**PUBLICATIONS EDITOR**

*Publications Editor Kirsti Nurkkala*

ISBN 978-951-42-8977-4 (Paperback)

ISBN 978-951-42-8978-1 (PDF)

ISSN 0355-3213 (Print)

ISSN 1796-2226 (Online)

