Erno Mäkinen

Face Analysis Techniques for Human-Computer Interaction

ACADEMIC DISSERTATION
To be presented with the permission of the Faculty of Information Sciences of the University of Tampere, for public discussion in Pinni auditorium B1096 on December 14th, 2007, at noon.

Department of Computer Sciences
University of Tampere

Dissertations in Interactive Technology, Number 8
Tampere 2007

ACADEMIC DISSERTATION IN INTERACTIVE TECHNOLOGY

Supervisor:	Professor Roope Raisamo, Ph.D., Department of Computer Sciences, University of Tampere, Finland
Opponent:	Professor Matthew Turk, Ph.D., Computer Science Department, University of California, Santa Barbara USA
Reviewers:	Professor Sudeep Sarkar, Ph.D., Department of Computer Science and Engineering, University of South Florida, USA Professor Jouko Lampinen, Dr. Tech., Laboratory of Computational Engineering, Helsinki University of Technology, Finland

Electronic dissertation
Acta Universitatis Tamperensis 686
ISBN 978-951-44-7184-1
ISSN 1456-954X
http://acta.uta.fi

Dissertations in Interactive Technology, Number 8

Department of Computer Sciences
FIN-33014 University of Tampere
FINLAND

ISBN 978-951-44-7050-9
ISSN 1795-9489

Tampereen yliopistopaino Oy
Tampere 2007

Abstract

Until quite recently people have interacted with computers using a mouse and a keyboard. This approach has been quite successful and effective given the importance that computers have gained in our information society. However, the mouse and the keyboard have many limitations. They do not support the natural way for humans to communicate using non-verbal and verbal communication channels, namely, vision and speech. However, this is about to change.

Computers have become powerful enough to process image and speech data. On the other hand, cameras are now inexpensive enough to be bought for home. The major challenge is how to create reliable perceptual technologies that allow applications with multiple modalities to be created. The focus in this dissertation is on one of the perceptual technologies, automatic face analysis.

Experiments were carried out for face detection and gender classification methods. Some of the methods used in the experiments were novel. Automatic face detection has to precede gender classification and other face analysis tasks in typical applications. Therefore, gender classification accuracy depends on the goodness of the detection. I studied how gender classification accuracy is affected when the goodness of the detection varies. It is also possible to use face alignment after face detection and various face alignment methods were also used in the experiments.

The face analysis techniques are intended for use in applications. At the end of the dissertation examples of already existing applications with face analysis are presented. Possible future applications are also considered.

Acknowledgements

It is hard to believe that the dissertation is ready. When I started in Tampere Graduate School in Information Science and Engineering in 2003 I believed that doing a dissertation would be fairly easy. However, I learned the hard way that doing research is actually quite challenging. Luckily, there were numerous people who helped me when needed. Without all these people I would never have finished this work.

First of all I thank you my parents, Laila and Pertti, and my sister Katja for your support in my life. Without you I would not be here now. I also thank my friends who have been an important support in my life and during my studying years in the University of Tampere.

Roope Raisamo deserves a lot of thanks for being my supervisor and for being the person who always believed that I would complete the dissertation even if I did not always believe that I would. Roope seems to have endless energy and a positive attitude when he leads the Multimodal Interaction Research Group that I have worked in.

All the present and past employees of the Multimodal Interaction Research Group deserve my thanks. It has been great to work with you and the same goes for the whole of TAUCHI and the Department of Computer Sciences. Kari-Jouko Räihä as the leader of TAUCHI also deserves my gratitude likewise head of the department, Jyrki Nummenmaa, and all the other administrative people. Veikko Surakka has also always been ready to help me when needed.

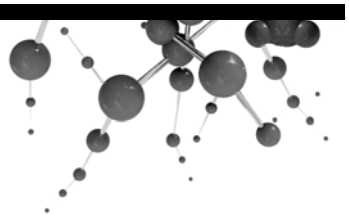
I also thank Markku Renfors and Pertti Koivisto who administrate the Tampere Graduate School in Information Science and Engineering. The yearly meetings as well as the other occasions have been good reminders for what I should do.

Special thanks go to Jukka Raisamo, Saija Patomäki, Poika Isokoski, Yulia Gizatdinova, Jouni Erola, and Stina Boedeker. Jukka and Saija have worked in the same room with me and they listened to me patiently, always. Poika often provided me with practical help and it has been nice to play Go after long days at work. Yulia has also helped me in my work and Jouni's enthusiasm for computer vision research has been delightful. Stina has an ability to make people feel esteemed and she is creating a great atmosphere in TAUCHI.

Contents

List of Figures

List of Tables



1 Introduction

Computers are nowadays a part of our everyday lives. In developed countries nowadays practically everyone with the exception of young children and some elderly people reads email and browses the web. Even in the developing countries more and more people have access to Internet. Computers are also moving from gray boxes on the table to the entertainment centers at homes, mobile phones are changing to multimedia devices with cameras, many laptops come with integrated web cameras, and so on.

The field that is interested in this change and how people interact with the technology is called Human-Computer Interaction (HCI). Hewett et al. (1992) defined HCI as "a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them". Hewett et al. also noted that there is no general agreement on the topics that belong under HCI.

The ongoing change makes it possible to develop new ways to interact with computers. However, it does not only make new ways of interaction possible: it requires them. There is a demand for easier and more rewarding interaction between humans and computers. One can easily see that entertainment has to be entertaining and new ways of interaction make this possible. One recent example is EyeToy by Sony (EyeToy, 2005). EyeToy is a webcam that can be installed on top of the TV and used in many PlayStation games. One of the games is EyeToy Sports that includes several sports games which can be played alone or by many people at the same time. Another example is EyeToy Kinetic Combat that allows user to

practice combat moves in front of a computer and try to match the moves with those shown on the screen. Another recent example is the Nintendo Wii console (Nintendo, 2006). This includes motion sensing controllers that enhance the gaming experience considerably.

However, entertainment is not the only place where changes are needed. It is also important to include special user groups, such as elderly and visually impaired people in the information society. New ways of interaction also provide tools for this.

How are these new ways of interaction realized in practice? We need tools and applications. Tools are needed for the applications that are provided for users. The tools may be, for example, software tutors (Hakulinen, 2006) that understand speech and provide help for users, they may be new motion sensing controllers, or they may be computer vision components that are added to a mobile phone.

1.1 Context and Problem Statement

Currently with most of the computers we use the interaction is still based on the keyboard and the mouse. This is often hard and causes frustration in users (Castrillón-Santana, 2003). To make the applications easier to use there is guidance on how to design applications, for example the ten usability heuristics by Jakob Nielsen (Nielsen, 1993). The heuristics recommend making the system state visible to the users, using terms familiar to the users and provide undo and redo functionality among others. There are also usability methods for evaluating the applications. The heuristics presented by Nielsen can be used for heuristic evaluation. Another method is usability testing where users are given tasks to do with the application and a usability expert makes notes on the problems encountered by the user and after analyzing the problems makes some suggestions on how to improve the application.

But ultimately, no matter how well we design and create the applications, they will have certain limitations imposed by the user interface. To push these limits further we can introduce new ways for users to interact with computers. These new ways include the use of the techniques such as speech recognition, gaze tracking, haptic feedback and computer vision (Piccardi and Jan, 2003) among others.

In homes where web cameras are becoming common computer vision is one of the technologies that can easily be used to enhance HCI. For example, some games already take advantage of this possibility.

Sports games especially have been developed in which computer vision plays a central part. The EyeToy by Sony was already mentioned. A somewhat similar game is Kick Ass Kung-Fu (Hämäläinen et al., 2005) in

which a player makes martial art movements in front of the camera and fights against computer opponents. There are very different computer vision enhanced games, too. LEGO MINDSTORMS NXT (Mindstorms, 2006) is a Lego robot building kit that includes a light sensor and an ultrasonic sensor to enable the robots built to see. In addition, it includes a touch sensor and a sound sensor to enable the robots to feel, touch, and hear.

However, computer vision can also be applied in other types of applications. Many mobile phones have a camera or even two cameras and some computer vision enabled applications exist for them. For example, in Helsinki, Finland, there is an ongoing experiment on using a mobile phone camera to get real-time bus time table information. There are two-dimensional barcodes installed at bus stops and when a user takes a picture of the barcode the Upcode application (Upcode, 2007) contacts the time table service and presents the time table information to the user.

Smart clothes are another type of application. In this case the user can wear, for example, a hat with a small camera that recognizes the user's hand gestures (Kölsch et al., 2004; Pentland, 2000), or the camera can be attached to eyeglasses and the names of the people the user looks at are identified using face recognition software and the user is reminded of the people's names (Pentland, 2000).

The applications presented above have perceptual user interfaces. Turk and Kölsch (2003) define perceptual interfaces as "highly interactive, multimodal interfaces that enable rich, natural, and efficient interaction with computers." In practice, this means that computer vision, speech and other input and output technologies are used where traditional I/O-devices keyboard, mouse and monitor are insufficient.

However, even though computer vision is already used in many applications there are still many challenges in the use of computer vision. One of the computer vision subfields is face analysis. First of all, many face analysis algorithms are too slow to be used in applications that require real-time or almost real-time responses. However, as computers become faster the problem becomes somewhat smaller (Piccardi and Jan, 2003). The other problem is that most of the existing algorithms are not robust enough to be used in most of the applications. For example, although there is commercial software for identity recognition based on faces available, they have still many limitations. Two of these are that faces to be recognized should be of good quality and frontal (Pentland, 2000).

The pattern recognition and machine learning fields are closely related to the computer vision field. Pattern recognition algorithms and machine learning algorithms are used in computer vision and advances in them often benefit the computer vision field, too. For example, there is a large

number of face analysis algorithms that use neural networks (Abdi et al. 1995; Golomb et al., 1990; Gray et al., 1995; Huang and Shimizu, 2006; Rowley et al., 1998a, 1998b; Tamura et al., 1996), Adaboost (Huang et al., 2004; Shakhnarovich et al., 2002; Sun et al., 2006; Viola and Jones, 2001; Wu et al., 2003a, 2003b), Hidden Markov Models (HMM) (Aleksic and Katsaggelos, 2006; Kohir and Desai, 1998; Yin et al., 2004) or support vector machines (SVM) (BenAbdelkader and Griffin, 2005; Castrillón-Santana et al., 2005; Moghaddam and Yang, 2000; Saatci and Town, 2006; Sun et al., 2002b; Yang et al., 2006b).

Face analysis itself is a wide topic even when approached only from the computer vision point of view. Face detection and tracking, face recognition, gender classification, facial expression and gesture recognition, age classification, and ethnicity classification are topics that belong under automatic face analysis. Although there has been progress within the last few years, all the topics include many unsolved problems. For example, currently a system that would be able to detect faces and recognize the identity reliably in all possible conditions does not exist. A frontal face looks very different from a profile face, lighting conditions affect the look of the face and a face with and without a beard looks different. Sometimes face recognition is hard or impossible even for humans, so it is unrealistic to expect perfect performance from computers. However, the existing systems are still far away from human performance except in very specific situations.

Face analysis is not only interesting from the viewpoint of applications with perceptual user interfaces. Psychologists are interested in human behavior and faces play a crucial part in the communication between humans. Psychologists are therefore interested in the findings of automatic face analysis. Computer scientists are also interested in the psychological research concerning faces and human behavior. For example, knowledge on how humans perform facial expression classification or gender classification generates ideas for automatic face analysis. On the other hand, when computer scientists analyze trained classifiers and find out on what bases the classifiers make classifications the psychologists get valuable information for their research.

Human vision and the vision of animals have also inspired the computer vision field. For example, 2-D Gabor filters (Daugman, 1988) are based on findings on how visual processing happens in cats' visual cortex and they are nowadays commonly used in computer vision algorithms (Huang et al., 2005; Lyons and Akamatsu, 1998; Shen and Bai, 2006a, 2006b).

Affective computing (Picard, 1997) is a field that handles issues of emotion processing on computers. Naturally, facial expression analysis is a part of emotion processing. However, in addition to the face also body postures

and gestures, and emotional pitches in our speech are caused by emotions, and analyzing these is also an interesting topic.

Face analysis is the main topic of this thesis. All the topics presented above are connected to the topic of this thesis although some of the topics are more important from the viewpoint of the thesis. In many places face analysis is considered especially from the HCI point of view. In addition, some pattern recognition and machine learning algorithms are presented in more detail because they have a central role in the experiments that are a part of the thesis. However, all the topics above are given some attention.

1.2 Research Questions

Besides HCI issues in applications that include computer vision there is a lot of research work to be done on computer vision algorithms. The face analysis field itself is also very broad. Naturally, it is impossible to answer all the questions arising from these fields in one thesis.

The main question of the thesis is how to create such face analysis algorithms that they are useful in HCI. We approach this topic from several aspects. The related questions are:

1. What application areas could benefit or have already benefited from applying face analysis in HCI?

2. Which face analysis algorithms and methods are the most applicable ones for HCI?

3. How can the applicability, usefulness and goodness of a method be measured especially from the HCI point of view?

4. How should face analysis methods be combined to be most useful in HCI applications?

5. What are the most problematic issues in combining the methods?

The first question is answered in the background and applications chapters. Questions two, three and, four are addressed mostly in the chapters concerning face and facial feature detection and gender classification. Questions four and five are addressed in a specific chapter that considers how to combine face detection, face alignment and gender classification.

1.3 Contribution

In the thesis, the face analysis field is considered as a whole. I have developed some novel algorithms and tools to be used in face analysis. In

addition, to enhance the understanding of the topic I performed experiments on face detection and gender classification. The algorithms have been used in HCI applications. Descriptions of these and many other computer vision applications are given.

Face analysis is considered from the HCI viewpoint where applicable. I contribute by describing the face analysis field comprehensively and broadly. In addition, the experiments are comprehensive and gender classification methods have been compared fairly and in various conditions. The thesis offers valuable knowledge for all the people doing research on face analysis field in the form of general knowledge on the field and in the form of results gained from the experiments. The results of the experiments are also largely applicable to other face analysis tasks such as face recognition and facial expression classification.

In the face detection experiment I analyzed one type of face detector that was created by myself and inspired by the earlier work by Sobottka and Pitas (1996). The frontal face detection accuracy was over 90% in the experiment and the detector could analyze over 20 images per second with a computer that had AMD Athlon 1.14 GHz CPU and 256 MB of memory.

In the gender classification experiments of Chapter 4 I studied how to achieve the best possible gender classification results in terms of classification reliability. I experimented with various different gender classification methods. I also studied how rotation, translation, and scale done to the face images affect gender classification accuracy. With high quality frontal face images which were manually aligned over 90% gender classification accuracy was achievable. Changes in face image rotation, translation, and scale impaired the classification performance. An Adaboost method with Haar-like features was most resistant to the rotation, and there were also differences between the classifiers with varying face image scales and translations.

In addition, in Chapter 5 I considered how face detection and gender classification should be combined. For example, automatic face alignment was used between face detection and gender classification. The results showed that the automatic alignment methods implemented decreased the classification accuracy when compared to the situation where alignment was not used. However, since manual alignment improved the classification accuracies the alignment could be useful if the alignment was reliable enough. In addition, the results showed that the alignment should be done before resizing the face images for classification. Furthermore, the results also showed that the quality of the face images affected gender classification accuracy. About 70% accuracy was achieved with images collected from the WWW while about 80% accuracy was

achieved with FERET database (Phillips et al., 1998) and web camera images.

Finally, the tools created and used to carry out the experiments described are available to other researchers. This enables them to carry out new experiments and study interesting issues in gender classification more easily. For example, the parallel training tool for Adaboost algorithm should be useful for other researchers and could be used in other pattern recognition problems besides face analysis.

1.4 Overview of the Thesis

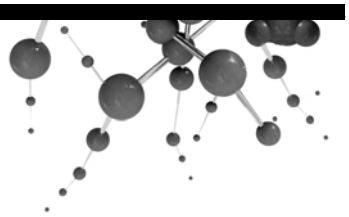
In Chapter 2 existing work and knowledge on the areas that are closely related to the thesis is described in detail, starting from perceptual user interfaces and human vision. The main issue of the thesis is then addressed by considering different aspects of computer vision. Different techniques available for use in computer vision are presented: person tracking, human motion analysis, and hand gesture recognition. Automatic face analysis is described in its own section focusing on the main face analysis tasks. Other modalities such as touch and sound are handled in their own sections.

After the background there is one chapter for each experiment that I carried out. The first experiment, presented in Chapter 3, focused on real-time face and facial feature detection. The experiment was carried out using a novel face and facial feature detection method. The data used in the experiments was collected with a web camera. The experiments in Chapter 4 were carried out to compare a wide variety of gender classification methods in various conditions. The data used was images collected from the WWW and good quality FERET (Phillips et al., 1998) face images. In Chapter 5 I studied how face detection and gender classification should be combined by investigating several gender classification techniques. I also studied the effect of face alignment and image size in gender classification. The FERET face database was used in this experiment.

The tools developed to carry out the research are described in Chapter 6. A face database tool was created to be able to easily edit the face databases used in the experiments. Face analysis tool was used to run tests for various gender classification methods using the data created with the face database tool. It was also used to train and analyze neural networks with various parameters for gender classification. Parallel training tool for discrete Adaboost was necessary to be able to train various Adaboost gender classification methods in reasonable time. The training of the Adaboost classifier is very time consuming, even though the Adaboost classifiers trained were very fast. Since there did not exist (and to the best

of our knowledge still does not exist) a public tool for training I implemented one and used it in the IBM eServer Cluster 1600 to train all the Adaboost classifiers.

Finally, before concluding the thesis applications for face analysis are considered in Chapter 7. At first, the existing applications in HCI are presented. These include the information kiosk developed at the University of Tampere and makes use of a face detection component. Ideas for future applications are presented at the end of the chapter.



2 Background

2.1 Introduction

In this chapter the basis for understanding the topics in the following chapters is given. Perceptual user interfaces that are the most potential target for automatic face analysis in HCI are introduced. Human vision that gives perspective and provides ideas for face analysis is also briefly covered. After that, digital image acquisition, image processing, machine learning and pattern recognition are discussed. Methods and algorithms used in the experiments of Chapters 3, 4, and 5 are described. A brief review of human activity recognition is followed by a more detailed review of face analysis research. Finally, some perceptual interaction techniques such as eye tracking, haptic interaction, and speech recognition are described and automatic face analysis is related to them.

2.2 Perceptual User Interfaces

Turk and Kölsch (2004) define Perceptual User Interfaces (PUIs) broadly as follows: "highly interactive, multimodal interfaces that enable rich, natural, and efficient interaction with computers." What makes perceptual user interfaces different from traditional graphical user interfaces is that they perceive their surroundings and use this new knowledge to enhance the interaction between humans and computers. These interfaces make use of several input and output modalities, such as speech and sound, vision, and touch. In many cases they are active and adapt to user needs and usage situation. However, the main point is that they make the interaction more natural using perceptual technologies than is possible with the traditional means using a keyboard, a mouse, and a display.

Keyboards and mice are useful with the graphical user interfaces (GUIs) because GUIs have been designed to be used with them. However, probably every computer user can say that interaction could be much easier, more enjoyable and more natural. This is even truer when we consider mobile phones that have small displays and keyboards. As described in Chapter 1, there is an increasing need for perceptual interfaces because computers are changing rapidly, they are used in novel situations and places, and there is a need to include special user groups in the information society.

Turk and Kölsch (2004) stated that interaction with a PUI should be like communication between humans. It should include similar social rules that are used in communication between humans. They also noted that there are some studies that consider traditional command-and-control interaction with computers more desirable. It may be that command-and-control interfaces fit well for specific applications. However, the studies that have investigated social aspects when people interact with computers indicate that people have the same social responses as when they communicate with other people (Turk and Kölsch, 2004). It seems obvious that perceptual user interfaces have plenty of applications where a command-and-control interface is insufficient.

In addition to developing novel and robust input and output modalities, there is a need for research in other fields, such as psychology, social, and cognitive sciences to make the best possible use of PUIs. The face analysis topic belongs under the broader computer vision field that focuses on understanding people and their activities, and vision is one of the modalities used in perceptual user interfaces. The focus in this thesis is on automatic face analysis. However, an introduction to topics such as speech, touch, human vision and computer vision in general is given because they are closely related to the main topic of the thesis: automatic face analysis in HCI.

2.3 HUMAN VISION

To understand challenges in computer vision and automatic face analysis, knowledge of human vision is insightful. A short introduction to the topic is given next and at the same time issues relevant to the face analysis are considered.

The human eye is depicted in Figure 2.1. The image formation starts so that the light emitted or reflected from an object passes through the lens. The lens is flattened or thickened by the ciliary muscles so that the lens is focused on the object of interest. The retina contains two kinds of receptors, rods and cones that sense the light and transform it to electrical impulses. Most of the cones, there are from 6 to 7 million of them, are located at the

center of the retina at the area called the fovea and are responsible for color vision as well as allowing us to see fine details. There are from 75 to 150 million rods and they are distributed all over the retina. The rods are sensitive to illumination and allow us to see large (and unfocused) area.

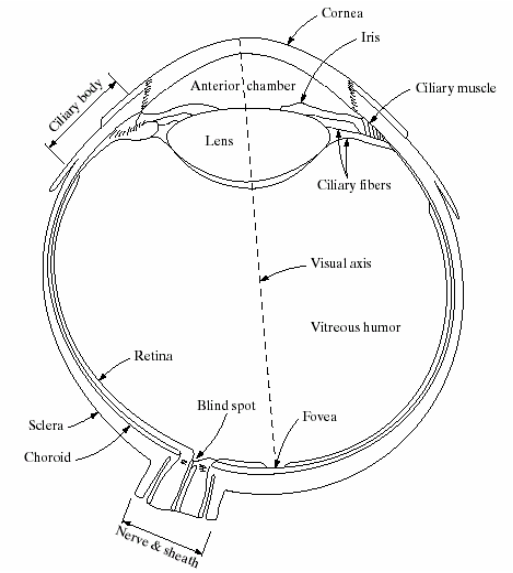


Figure 2.1. Human eye (Gonzalez and Woods, 2002, pp. 35).

The fovea has a diameter of 1.5 mm and because it is rather small the eyes can only focus on a small area at the time. Eye movements are called saccades and there is a fixation when the eyes are stationary and focused on a small area. Understanding of the whole scene or the big picture is formed during several fixations.

The electrical impulses emitted by the cones and rods are transferred through the optical nerve that starts from the blind spot and ends at the *lateral geniculate nucleus* (LGN) inside the brain. Both eyes are connected to LGN and it is further connected to the visual cortex. Not all the functions of this organ are known. However, besides sending information to the visual cortex it also receives feedback from the cortex. One known function is that it separates (decorrelates) visual information temporally. In other words, the electric impulses received from the eyes at different times are not mingled together.

Visual cortex has many regions with specific functions. Some parts are specialized in the detection of motion while others analyze color, or the meaning of the received signal. When the visual cortex processes visual information it differentiates between edges, regions, and decides the connections between them. This analyzing includes more than just seeing colors, shapes, and motion. It includes determining what we see on a higher level. In this phase meaning is given to the electrical signals

received from the eyes. This process is complex and requires our understanding of the world. It also includes combining the information that is received during several fixations.

Different visual clues are used in this process. For example, although both eyes are helpful in perceiving depth information, perspectives and other learned knowledge give hints of the depth. The 3D effects in 2D display are only possible because our brain processes the visual information so that we perceive the 2D objects as 3D objects. Our brains also tend to group objects that are close to each other, tend to complete patterns in certain situations, and so on. Many of these clues are known as Gestalt principles. Examples of various clues that are used in visual processing are given in Figure 2.2.

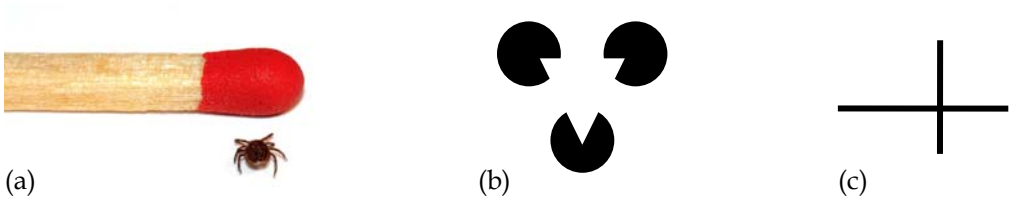


Figure 2.2. Examples of clues used by human vision system in perceiving the world. (a) Knowledge. We know the size of the tick because we know the size of the match (Photo taken by Karwath (2005)). (b) Closure. We see the white triangle because our brain completes the pattern. (c) Continuity. We see two lines rather than two arrowheads.

Face processing receives somewhat special treatment in brains. It has been known for quite a long time that a certain brain injury can impair face recognition abilities while a human can still recognize other objects. The condition is known as prosopagnosia or face blindness. Furthermore, many studies (Sergent et al., 1992; Kanwisher et al., 1997; Tsao, 2006) have shown that there are brain regions and cells which are more sensitive to faces than to other objects. However, there are also studies (Gauthier et al., 1999) suggesting that specific brain regions are sensitive to faces partly because humans learn to be experts in face perception.

A recent study by Tsao et al. (2006) showed that specific brain cells in macaque monkeys are sensitive to the specific properties of the face. In the experiments they used cartoon faces and varied 19 properties including locations of the eyes, nose and mouth, and face width and height. One cell produced a high response to eight properties at most. The most popular properties were face width and height, and iris size. Extreme features typically produced the highest cell responses (see Figure 2.3).

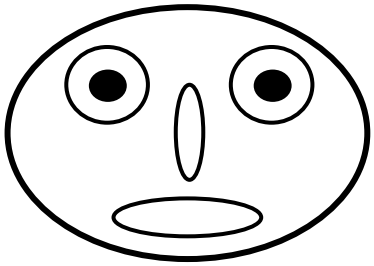


Figure 2.3. A cartoon face that causes high activity in the face specific brain regions.

From the above one can get an idea of what a complex thing vision is. The main complexity with computer vision lies in analyzing the captured image: separating different parts of the image, grouping certain parts together, and giving meaning to the objects formed from the parts. By understanding human vision one can apply rules and clues from human vision to the development of computer vision algorithms.

2.4 COMPUTER VISION

The computer vision field is rather extensive. It has applications from industry to homes. However, many of the underlying processes and techniques are the same for all application areas. Next, an overview of these processes and techniques is given. Digital image acquisition and processing are the first topics since they form the basis for higher level processing, such as pattern recognition, when computer vision is considered. Machine learning and pattern recognition are the second two topics, since they are also applied extensively in computer vision. In fact, many machine learning and pattern recognition techniques, such as neural networks and support vector machines (SVM) are also used in many other fields than computer vision.

2.4.1 Digital Image Acquisition and Processing

Digital image acquisition is the first step in any computer vision system. There are several ways to acquire an image. The image may be acquired in visible, infrared, ultraviolet, x-ray, gamma-ray, and radio-wave bands or it may be formed from sound as in medical ultrasound imaging or from some other source. In the scope of this thesis the images are usually acquired in a visible band because most cameras and video cameras work in this band and a visible band is a fairly natural choice for human-computer interaction. However, visible band is by no means the only medium for use in HCI. For example, infrared sensors are typically used in gaze tracking.

When a light source emits light, the light is partially absorbed and partially reflected from the objects in the scene. The camera senses a part of the light in the scene when the light passes through the camera lens and

the lens refracts the light to the sensors that transform the sensed visible light (or some other energy) into electrical form, voltage. The intensity of the light determines the strength of the voltage.

Digital cameras have either CMOS (Complementary Metal–Oxide–Semiconductor) or CCD (Charge-Coupled Device) arrays that sense light and transform it into voltage. The array is a group of sensors arranged in a grid. The number of sensors in the array depends on the camera but a typical web camera has 640*480 size array and, for example, Canon PowerShot SD600 pocket camera has the largest image size of 2,816*2,112 pixels (6 Megapixels) meaning that the CCD sensor array is close to that size too.[1]

After the sensed light has been transformed to the voltage it is further quantized. Quantization means that voltages at a certain range are defined to have a certain same value. For example, there could be 256 distinct values for the quantized voltage. Finally, after quantization the image is in digital form and can be stored or further processed.

The low-level image processing may include filtering the image in spatial or frequency domain, doing histogram equalization for it or transforming its intensities by a log-transform, for example. The common purpose of low-level processing is to enhance the image (Gonzalez and Woods, 2002, pp. 25-28). There can also be some morphological processing and segmentation of the image parts. Not all this processing does need to happen before higher-level processing that in the case of this thesis is pattern recognition. Instead, the processes are usually interleaved. For example, faces can be detected from an image using a pattern recognition algorithm and after the faces have been detected histogram equalization can be performed for each of them.

In the experiments I used histogram equalization and connected component labeling, so these image processing techniques are described next. Some feature extraction and machine learning techniques are described in the next subsection because they fit well under the machine learning topic.

Histogram equalization spreads the intensity values of the images over a larger range. It is used because it decreases the effect of different imaging conditions, for example different camera gains and it may also increase image contrast (Rowley et al., 1998a). When analyzing faces it is important that there is as little variation due to external conditions (such as imaging

[1] The CCD array in this case has actually a rather more sensors than the maximum image size is. The reason for this is cheaper mass production.

conditions) as possible, so that the variations between the faces become more visible, that is the issue we are interested in.

The classifier and data representation used as input to a classifier determines if histogram equalization should be used. For pixel-based input it is often useful. Haar-like features (see Subsection 2.4.2) can also benefit from histogram equalization although they use intensity differences between pixels. The reason is that intensity differences for images with different intensity distributions produce different results. Gabor features[2] on the other hand are robust against local distortions caused by variance in illumination (Shen and Bai, 2006a), so histogram equalization is not necessary when using them.

Histogram equalization is simple to implement and computationally inexpensive. The function that maps image pixel intensity to a histogram equalized value is

$$s_k = \sum_{i=0}^{k} \frac{n_i}{n}, k = 0,1,2,...,L-1$$

where s_k is histogram equalized intensity value for kth intensity value in the range L of total number of possible intensity values in the original and target image, n is the number of pixels in the original and target image, and n_i is the number of image pixels that have intensity value i in the original image.

Examples of histogram equalized face images with their original counter parts are shown in Figure 2.4. As can be seen, face intensities look more uniform and in one case contrast has improved dramatically.

[2] A comprehensive introduction to the Gabor features will be found in the thesis by Kämäräinen (2003)

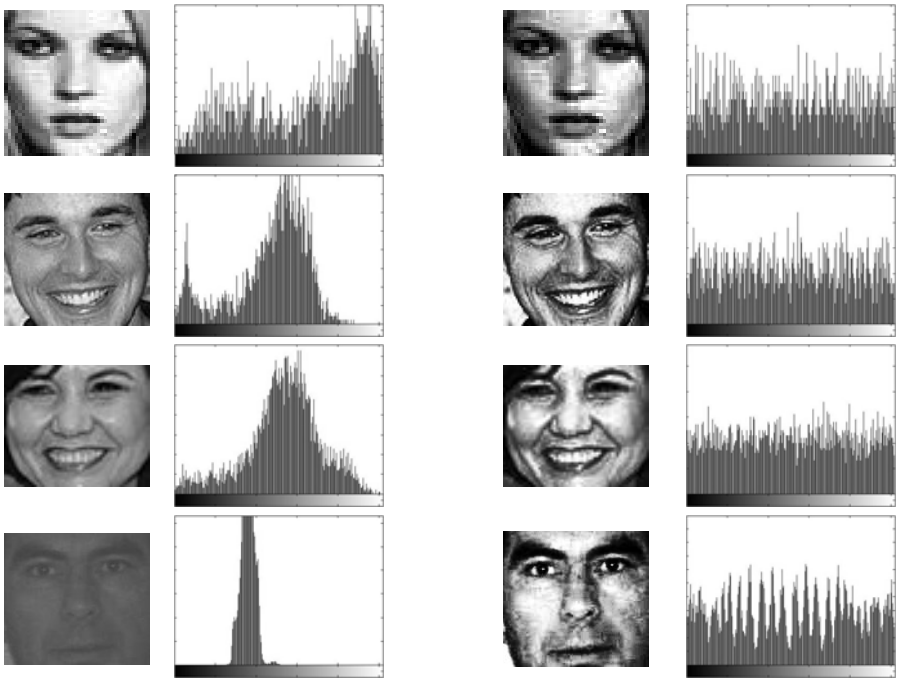


Figure 2.4. Original images are shown on the left and corresponding histogram equalized face images are shown at the right. The histogram of each face image is shown at the right side of the image.

Although histogram equalization usually produces good results, it is worth noting that in some rare cases it does not work well. Such examples are shown in Figure 2.5 and Figure 2.6. The result in Figure 2.5 is poor because there is a large number of black pixels (intensity value 0) in the original image and intensity values are concentrated at the low end of the intensity range. As a result the values of the histogram equalized image are not spread over the whole intensity range (see Figure 2.5d). Instead, the lowest value is around 75 on a range 0-255. However, this is an extreme case and it was necessary to modify the example image manually to demonstrate the possibility of unsuccessful histogram equalization. Modification was done by adding black background to the left side of the image and by flattening dark intensity levels of the image. Histogram specification (also known as histogram matching) would work better in this case but, unlike histogram equalization, it requires manual parameter setup. From this it follows that histogram equalization is more useful in automatic face analysis systems.

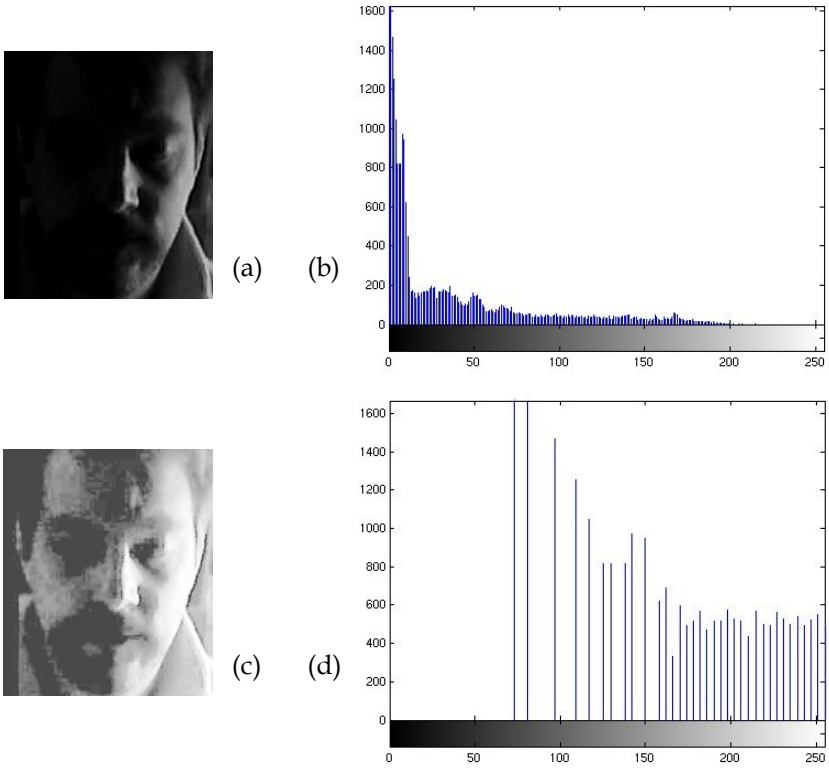


Figure 2.5. An example face image that histogram equalization does not work with. (a) Original image. (b) Histogram of the original image. (c) Histogram equalized image. (d) Histogram of the histogram equalized image.

The example in Figure 2.6 is a more realistic one than the previous example. In this case the face has strong shadows and after histogram equalization the right part of the face has burned out. To remove the effect, a method to remove shadows could be used (see Subsection 2.6.3) before doing the histogram equalization.

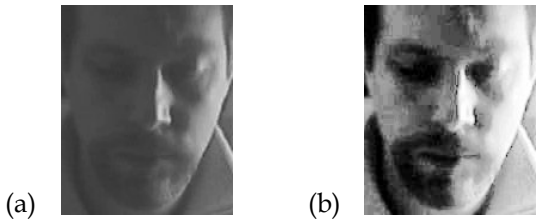


Figure 2.6. (a) Original image with strong shadows. (b) Image after histogram equalization. Histogram equalization does not remove shadows and the right side of the face has burned out. For example, illumination gradient correction (Sung and Poggio, 1998) could be used in addition to histogram equalization.

Connected component labeling is a method used for finding regions from an image. It can be used in face detection to find skin colored regions from an image. We used connected component labeling for that purpose with our face detection method described in Chapter 3. After regions have

been found further processing with some other method can be used to separate faces from other skin colored regions.

The algorithm for connected component labeling at the general level is shown in Figure 2.7.

1. Set criteria that define when two pixels are similar enough to belong to the same region.

2. Starting from the top left corner of the image, move pixel by pixel to the left and down until the bottom right corner of the image is reached.

3. For each pixel *p* check the nearest neighbor pixel above and to the left of the pixel *p*. If neither of the neighbors fulfills the defined criteria with the pixel *p*, give a new label for the pixel *p*. If only other neighbor pixel fulfills the defined criteria then give the pixel *p* the same label as the neighbor has. If both neighbors fulfill the criteria and they have same labels then give the label to the pixel *p*. If both neighbors fulfill the criteria and they have different labels then give the pixel *p* either label and add both labels to the equivalency class.

4. After the bottom right corner of the image has been reached go through the image second time. This time for each pixel *p* check if its label is in an equivalence class. If the label is in the equivalence class then common label is given to the pixel *p* with the rest of pixels which labels are in the equivalence class.

Figure 2.7. Algorithm for the connected component labeling.

There are many factors affecting the performance of the connected component labeling algorithm. Wu et al. (2005) studied ways to optimize it. I have used the well-known union-find algorithm and tree structure for updating and accessing the equivalence class. This was sufficient to achieve real-time performance in the experiments when connected component labeling was used for finding skin colored regions with a standard PC and with image size of 320*240 pixels.

2.4.2 Machine Learning and Pattern Recognition Techniques

Duda et al. (2001, p. 1) define *pattern recognition* as follows: "the act of taking raw data and making an action based on the "category" of the pattern." The definition means that pattern recognition is a rather broad field. In the case of face analysis raw data means images (not only face images) and action can be, for example, classification determining whether there is a face or not in the image or whose face is in the image. To be successful in the action a pattern recognition system has to be constructed of appropriate parts. Usually the raw data has to be changed into another form that is used in classification. Raw data can be changed into another form using various digital image processing techniques and feature extraction methods. Machine learning techniques are used for the classification. In the case of face analysis *machine learning* means that the computer adapts to the classification problem so that it can distinguish faces from non-faces, identities of the faces, genders of the faces, and so on.

There are numerous feature extraction and machine learning techniques. For example, Gabor wavelets have a biological basis and also have many other properties that make them attractive for face analysis and computer vision in general. However, it is practically not possible to describe all existing techniques in one thesis. Instead, those that were used in the experiments of the thesis are described in detail.

The techniques presented differ a lot from each other. However, one thing common to all the machine learning techniques presented is that they learn from examples. This means that a set of positive and negative data examples, for example faces and non-faces, or female and male faces, are used in the learning. The classifier learns to discriminate classes from each other (faces from non-faces, females from males, and so on) when it is trained with the examples.

Neural networks have been used in various fields including computer vision. Multi-layer perceptrons, one type of neural networks, are possibly the most used type in computer vision problems. In the experimental part of the thesis a multi-layer perceptron is applied to gender classification. An example of the multi-layer perceptron is shown in Figure 2.8.

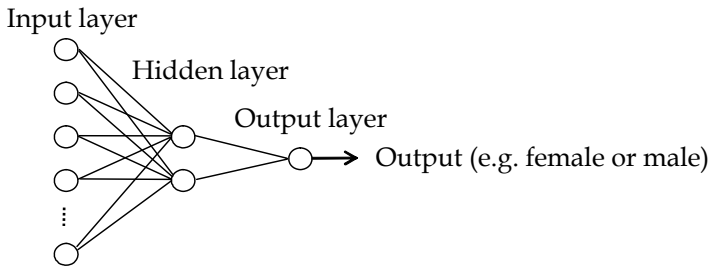


Figure 2.8. Example of a multi-layer perceptron with one hidden layer and one output node.

A multi-layer perceptron is a neural network with an input layer, an output layer, and with one or more hidden layers. Each layer has a set of nodes and there are connections between the nodes. Each connection is represented by a weight. The data to be classified is inputted to the network through the input layer. Each node at the input layer represents one data value. The input layer feeds the data forward to the nodes of the hidden layer. The hidden layer then feeds the input further to the other hidden layer or to the output layer. The output of the output layer is the classification result. When the data is fed through the network it is modified by the neuron activation functions. The functions modify the neuron inputs using the connection weights and then they output the modified input to the next layer.

The learning of the multi-layer perceptron is based on changing the connection weights between the nodes of the layers. A multi-layer

perceptron is typically trained with the back-propagation algorithm. There is a set of the training data (for example, faces and non-faces) that is used for the training. The training proceeds so that the training data examples are fed one by one into the network and the network weights are modified based on the difference between the produced output and expected output. The name back-propagation comes from the fact that weight updating takes place by updating the output layer weights first and then proceeding layer by layer towards the input layer. The training takes place in rounds so that after all training examples have been inputted to the network the new training round starts by inputting the first example to the network.

In addition to the training images there is usually a set of validation images. They are used to avoid over-fitting to the training data. Over-fitting means that the neural network fits to the training data so well that the classification of the unseen data suffers. In practice, validation takes place so that the validation images are classified with the neural network after each training round and the classification error, the difference between the expected and real outputs is calculated. Training is stopped when validation error starts to increase, which means that the classification of the validation images begins to detoriate.

Adaboost (Freund and Schapire, 1997) is also a fairly popular method among face analysis methods. The face detector by Viola and Jones (2001) had a cascade of discrete Adaboost classifiers using Haar-like features (see Figure 2.9). Different variations to the detector have been suggested. For example, Lienhart and Maydt (2002) extended the original set of Haar-like features with 45° rotated features. Huang et al. (2004) presented a nested cascade face detector and used real Adaboost (Schapire and Singer, 1999) in place of the original discrete Adaboost (Freund and Schapire, 1997). Other modifications to the Adaboost have been suggested, for example Logitboost (Friedman et al., 1998), Floatboost (Li and Zhang, 2004) and online boosting (Oza, 2001). Wu et al. (2003a) suggested a new type of Adaboost algorithm called LUT Adaboost and used it for gender classification. Later they also used it with the cascaded face detector (Wu et al., 2003b).

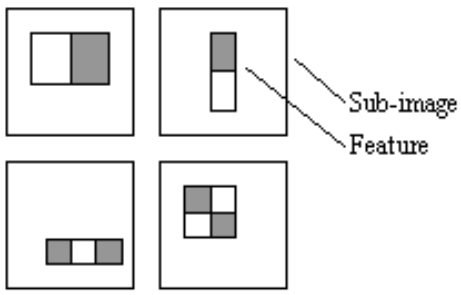


Figure 2.9. Haar-like features used with the cascaded face detector.

Haar-like features, that are often used with Adaboost and which we have also used, extract information from the raw image pixel data. Each Haar-like feature has a specific type and a specified location. Each type of the Haar-like features that we used in the experiments (and that were also used with the cascaded face detector by Viola and Jones (2001)) are shown in Figure 2.9. Each Haar-like feature has a value that is calculated as follows (see also Figure 2.9):

1. The Haar-like feature is placed on the specific location inside the face sub-image.

2. Pixel intensities inside the dark rectangle(s) are summed together. A sum is calculated similarly for the white rectangle(s).

3. The sum of the dark rectangle(s) is subtracted from the sum of the white rectangle(s).

The discrete Adaboost training algorithm is shown in Figure 2.10. The examples in the context of the thesis are face images. The training takes place in rounds. Each example face is given an equal weight at the start. This means that each face has an equal effect on feature selection. At each round each training data example is classified with each unselected weak classifier and feature. At each round a weak-classifier and the feature with the lowest classification error are selected and included in the resulting feature set. The example faces that are classified correctly with the selected feature are given lower weight for the next feature selection round so that the misclassified faces have more effect on the next selection. The number of training rounds determines the number of features that will be selected for the final classifier. The strong classifier is formed of all the selected features, and the features selected first tend to have more effect on the classification.

With the threshold weak classifiers each weak classifier has a threshold value that is determined during training. When an image is classified with the weak classifier the value calculated for the corresponding Haar-like feature is compared to the threshold. The weak classification is decided either as male or female depending on whether the calculated value is smaller or bigger than the threshold. As described above, the final classification is a result of the all weak classifications, so that the features selected earlier during training are weighted more.

Given example images (x_i,y_i); $i = 1,\ldots,n$ where $y_i = \pm 1$ for positive and negative examples respectively.

Initialize weights $w_{t,i} = 1/n$; $i = 1, ...,n$; $t = 1$

For $t = 1, \ldots, T$:

1. Normalize the weights so that the $\sum_{i=1}^{n} w_{t,i} = 1$.

2. For each feature j calculate the error $e_{t,j} = \sum_{i=1}^{n} w_{t,i} \left| h_j(x_i) - y_i \right|$, where h_j is the weak classifier for the feature j and it produces value -1 or 1.

3. Select the feature, j_t, with the lowest error $e_{t,j}$, and include it in the resulting feature set.

4. For each x_i classified correctly, $w_{t+1,i} = w_{t,i}\beta_t$, where $\beta_t = e_{t,j}/(1 - e_{t,j})$.

The strong classifier is $H(x) = \begin{cases} 1, if \sum_{t=1}^{T} \alpha_t h_t(x) \geq d \\ -1, otherwise \end{cases}$, where $\alpha_t = \log \frac{1}{\beta_t}$ and optimal d is near $\frac{1}{2} \sum_{t=1}^{T} \alpha_t$.

Figure 2.10. Training algorithm for the discrete Adaboost.

The mean Adaboost that I used in addition to the threshold Adaboost and LUT Adaboost in the experiments is our own Adaboost variant where the threshold is selected so that mean feature value is calculated separately for positive and negative face examples and the threshold is set to lie halfway between the means.

LUT Adaboost (Wu et al., 2003a) has, instead of the threshold, a lookup table (LUT) for each feature. The lookup table has a specified number of bins, for example five bins. Each bin corresponds to a certain range of possible feature values. The bin ranges are of equal size and the total range of bins includes all possible feature values. During training the feature values are calculated for each example. The counter of the bin that matches the calculated feature value is incremented by one. After training each bin contains the number of positive examples that had feature value within the range of that bin. The number of negative examples per bin is similarly stored. During classification feature value is calculated and the numbers of positive and negative examples inside the bin that correspond to the feature value are retrieved. If the number of positive examples is greater than the number of negative examples, then the classification result with the feature is positive and otherwise negative. The final classification result is calculated exactly as it is calculated with the

threshold Adaboost and with the mean Adaboost. It is the weighted classification result formed of the all weak classifications.

Support Vector Machine (SVM) is a relatively new machine learning algorithm from 1992 (Boser et al., 1992). It has been successfully applied in many face analysis problems. Many examples of gender classification studies where it has achieved the highest classification rates are given in Subsection 2.6.5. SVM was also used in the experiments described in Chapters 4 and 5.

SVM transforms the data, which may have been preprocessed, in high-dimensional feature space and classifies the data in that space to specified classes. The classes are separated from each other by maximizing the margin between the training examples of different classes in the space. The training examples that separate the classes from each other are also called support vectors. The data may be face images as in our case, or for example, hand images. In the experiments described in this thesis the face images had two classes: male and female.

The transformation to the high-dimensional space with SVM is done using a kernel function. Linear, polynomial and radial basis function (RBF) and Gaussian kernels have been widely used in the applications. I used an RBF kernel in the experiments. Each kernel has a different set of parameters and there are various algorithms for finding the optimal parameters for them. The RBF kernel has two parameters that determine the result of the training with the set of training examples used. These parameters are cost and gamma.

Local binary patterns (LBPs) (Ojala et al., 1996) are features the values of which are calculated from the image pixel intensities in a local pixel neighborhood. The basic idea is that as many binary values are created as there are pixels in the neighborhood of the center pixel and at the end these are concatenated to one binary value. The algorithm for calculating the value for the LBP-features is given in Figure 2.11.

1. For each pixel (g_p; p= 0, …, P-1; P is the size of the neighborhood) in the neighborhood of the center pixel g_c, calculate difference $x_p = g_p - g_c$.
2. The value of LBP is calculated with $LBP_{P,R} = \sum_{p=0}^{P-1} s(x_p) * 2^p$, where P is the size of the neighborhood and R is the radius from the center pixel. $s(x_p) = 1$ if $x_p \geq 0$, and otherwise $s(x_p) = 0$.

Figure 2.11. Algorithm for the calculation of LBP feature value.

An example of using the local binary pattern on a face image pixel neighborhood is shown in Figure 2.12. The example in Figure 2.12 is of the $LBP_{4,1}$-feature that has a radius of one pixel and 4 neighbor pixels next to the center pixel.

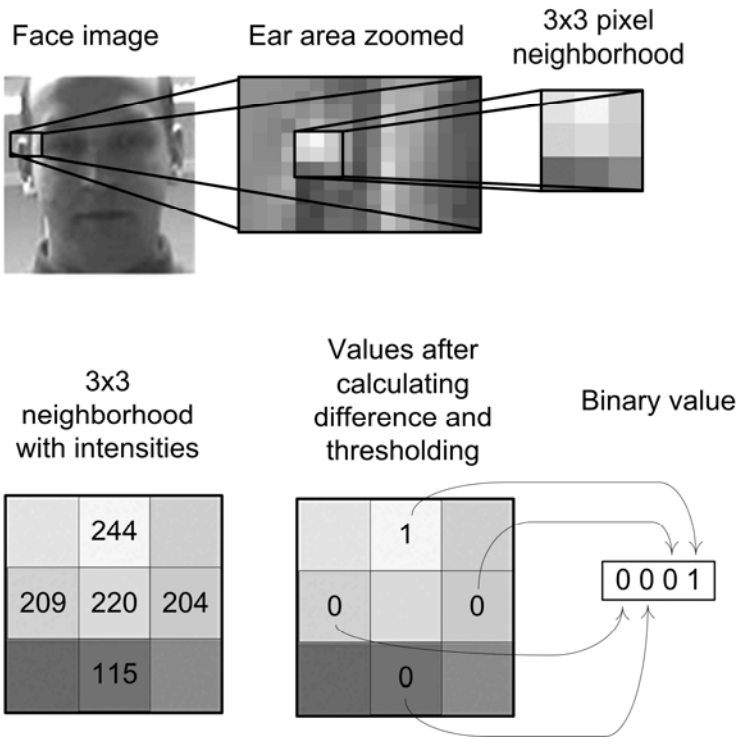


Figure 2.12. An $LBP_{4,1}$-operator in use.

Originally LBP was defined for the 3*3 pixel neighborhood but later Ojala et al. (2002) extended it to different neighborhoods and introduced rotation invariant and rotation invariant uniform extensions to it. Rotation invariant uniform patterns solve the practical problem that some patterns may occur too rarely to create reliable statistics for specific analysis problem. The calculation with these rotation invariant uniform patterns proceeds first as with regular LBP. However, after a concatenated binary value has been created, it is rotated so that as many significant bits as possible are zero before the first occurrence of binary value one. For example, 01100000 would be rotated to 00000011. In addition, if the binary value has more than two bitwise transitions (from zero to one or from one to zero) then it is changed to a predetermined value that is the same for all such binary values that have more than two bitwise transitions. For example, binary values 01010000 and 10101010 would be changed to a predetermined value while the binary value 01100000 would only be rotated. The formula and a more detailed description for the basic LBP and rotation invariant uniform LBP calculation will be found in the journal article by Ojala et al. (2002).

2.5 HUMAN ACTIVITY RECOGNITION

Human activity recognition refers to the recognition and analysis of human activities during a certain time period. Activity can be walking, sitting, throwing a ball, using hand gestures, and so on. Here the activity recognition is divided in two areas. At first person tracking and motion analysis is described. After that hand gesture recognition which is a rather specific research area is described in its own subsection.

2.5.1 Person Detection, Tracking, and Motion Analysis

Person tracking is a specific area of the more general object tracking problem. Yilmaz et al. (2006) recently reported a survey on object tracking. They defined *object tracking* as "the problem of estimating the trajectory of an object in the image plane as it moves around a scene". Naturally, in the case of person tracking the object is a person. Yilmaz et al. (2006) mentioned that robust real-time trackers that work in simple scenarios have been developed in the last few years. As usual with computer vision tasks there are numerous issues that may cause problems for tracking. Yilmaz et al. (2006) stated that the possible problems for object tracking are the following:

1. "Loss of information caused by projection of the 3D world on a 2D image
2. noise in the images
3. complex object motion
4. non-rigid and articulated nature of the objects
5. partial and full object occlusions
6. complex object shapes
7. scene illumination changes
8. real-time processing requirements."

All these problems are possible with person tracking. If only one (non-stereo) camera is used for the tracking then only the 2D image of the 3D scene is available. There is always some noise in the images. However, typically there is so little noise that it does not cause major problems. While movements of people have some general rules such as head and torso are connected to the legs and hands, and while the application may offer some context specific knowledge such as pedestrians usually continue walking in the same direction, people's movements can still be considered complex. Humans have both a non-rigid and an articulated nature (Wang and Singh, 2003) and humans also have a complex shape. In many applications, there may be several people simultaneously in the

scene and there can be, for example, furniture in addition to the people in the scene. Persons may occlude each other or they may walk behind the furniture. Especially outdoors but also indoors lighting may change and cause problems in tracking. Many person tracking applications require that tracking should take place in real-time. This holds especially in the HCI field, where feedback to the users should often be real-time. An example of such an application is the computer game QuiQui's Giant Bounce (Hämäläinen and Höysniemi, 2002) where children interact in the game by making different "flying" movements in front of the web camera.

The tracking requires that the object is detected in every image frame or before tracking begins (Yilmaz et al., 2006). Naturally, if tracking is real-time and detection happens in every frame then detection also has to happen in real-time.

Yilmaz et al. (2006) also mention assumptions that can be used to make tracking easier. According to them "almost all tracking algorithms assume that the object motion is smooth with no abrupt changes". They also mention that if objects can be assumed to have constant velocity or acceleration or if the number, appearance, shape, and size of the objects are known then tracking is easier. In addition, they mention using auditory information together with visual information, for example, to track a person's mouth.

There are also surveys specific to person tracking and human motion analysis in general (Aggarwal and Cai, 1999; Gavrila, 1999; Moeslund and Granum, 2001; Wang and Singh, 2003; Zhou and Hu, 2004).

Tracking of a human body can be done using appearance-based and model-based approaches (Wang and Singh, 2003). Appearance-based methods use color or texture information while model-based methods use a priori-knowledge of human motion. Tracking can take place in 2D or in 3D. Almost all methods that do tracking in 3D are model-based approaches (Wang and Singh, 2003).

Reflective markers are also commonly used in human motion analysis. In this case human motion is modeled by tracking the markers that are placed around the body. The very first marker based-system was the Moving Light Display (MLD) system by Johansson (1975). Nowadays there are many commercial systems (Ascension, 2007; Qualisys, 2007; Vicon, 2007) available that track humans using markers.

A well known example of a person tracking system without markers is Pfinder developed by Wren et al. (1997) in MIT. It models a person using 2D blobs, so that each differently colored region, usually head, upper body, hands, lower body, and legs has own blob. The blobs have spatial and color properties. The system is initialized so that it learns scene

appearance without people. A person entering the scene is detected by comparing pixels of the model to the pixels of the latest camera image. When sufficiently large areas are detected which do not belong to the background, the system builds the blob model using 2D contour shape analysis. Pfinder works in real-time and is robust to occlusions but will not work if there are sudden changes in the scene, such as changes in lighting, because the scene model cannot be updated accordingly.

Applications for human motion analysis range from animation and virtual reality to walking disorder diagnosis (Whittle, 1996) and person identification based on gait analysis. Gender can also be determined by analyzing human motion.

2.5.2 Hand Gesture Recognition

Hand gesture recognition is useful in HCI because it can be used in place of and together with traditional input channels (keyboard and mouse). For example, a system can be commanded with hand gestures.

Computer vision can be used for tracking hands and recognizing hand gestures. Another possibility is to use gloves that contain position and movement sensors. Computer vision may be preferred in many applications because of its naturalness for the user (Pavlović et al., 1997). However, if reflective markers are used with the computer vision approach, as has sometimes been done, then naturalness suffers.

Pavlović et al. (1997) divided computer vision based hand gesture analysis approaches into two classes: 2D appearance-based and 3D model-based approaches. They stated that 3D models, although they can model all hand gestures, are computationally expensive. Appearance-based methods are computationally less expensive, which is important in HCI applications.

The system by Shan et al. (2007) is a recent example of a real-time hand tracking and gesture recognition system. They carried out experiments to test the system. Besides measuring the tracking performance, the system was used for controlling a wheelchair. The system integrated two machine learning techniques: particle filtering (Arulampalam et al., 2002) and mean-shift tracking algorithm (Comaniciu et al., 2000).

2.6 Automatic Face Analysis

The face provides vast amount of information. We can tell a lot about the other person and his or her feelings just by seeing his or her face. Looking at another person's face and eyes is natural and usually expected when discussing face to face. On the other hand, communication between humans and computers still mostly happens using, keyboard, mouse, and a display. When more effective, versatile, and user friendly ways to use

computers are developed automatic face analysis is one promising tool to be used.

There are many ways that automatic face analysis can be used. A person can be identified from the face and facial expressions can be analyzed so that the computer can adapt to the usage situation, for example. In Chapter 7 various applications in HCI are introduced that take advantage of automatic face analysis. In this section background knowledge including existing research and technical aspects of face analysis is given.

Besides the applications, face analysis research is also interesting from other aspects. Because the face is a complex non-rigid object, the results of face analysis research can be useful in a wider range of object recognition tasks. For example, Yang et al. (2002) stated that most of the research on object recognition has been limited to rigid objects. They also stated that large sets of faces have been used in the experiments, which have been rare with other objects. Both these properties of face analysis research may help in other object recognition problems.

The relation of the automatic face analysis process to the other parts of the typical HCI system is shown in Figure 2.13 and the face analysis process is illustrated in Figure 2.14.

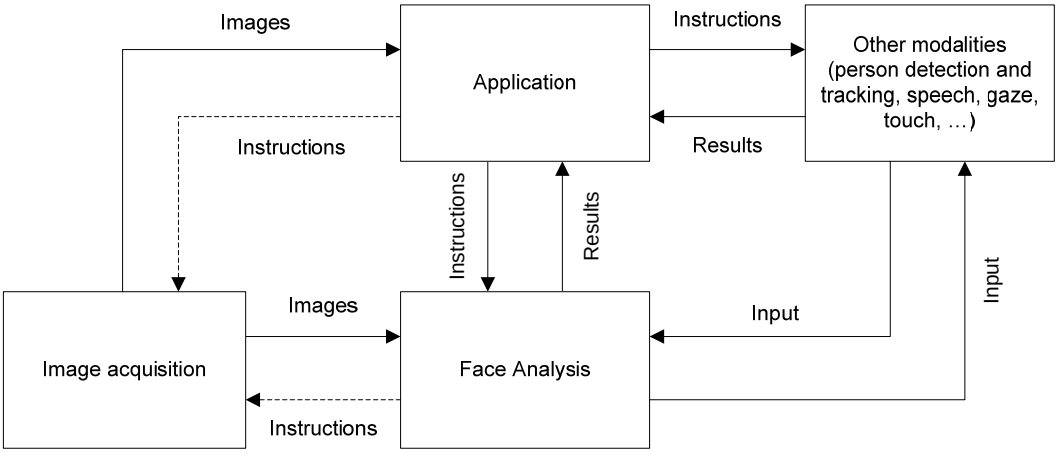


Figure 2.13. Face analysis related to the whole HCI system.

As shown in Figure 2.13 the application controls the whole system by giving instructions to the other components of the system. The other components produce results that the application utilizes. The image acquisition component provides images, for example by capturing them with a video camera and the application may choose to show them to a user if that fits the context of the application. The face analysis component analyzes images that are inputted to it by the image acquisition component and feeds the results into the application. The results may include but are not limited to the number and locations of the faces in an image at a certain moment, facial expressions, genders of the faces, and

identities of the people whose faces have been detected in the image. Both the application and face analysis component may give instructions to the image acquisition component. The direction of the camera and camera parameters can be changed, for example. There may be other components such as speech and gaze tracking components that communicate with the face analysis component, so that the application receives multimodal feedback from them. For example, person identification can be based on both face and speech data, and gaze information can be combined with the facial expression data.

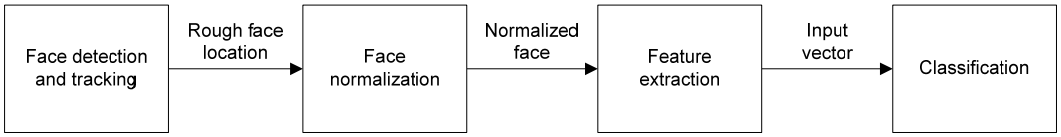


Figure 2.14. Face analysis in detail.

The first task in the face analysis process is to find the faces as shown in Figure 2.14. Depending on the application the faces may be tracked over time or detected from a single image (or from video image but without tracking). Yang et al. (2002) define *face detection* as follows: "Given an arbitrary image, the goal of face detection is to determine whether or not there are any faces in the image and, if present, return the image location and extent of each face." *Face tracking* means estimating the location of the face continuously from video image (Yang et al., 2002). In addition to face detection and tracking facial features may be detected and tracked. *Facial feature detection* means finding the locations of the features such as eyes, nose and mouth from a face.

If person detection is used in addition to face detection then person detection might be done first or it might happen at the same time with face detection but this is not a necessity. In addition, in some specific applications face images are inputted to the system and faces need not to be detected or tracked.

The results of the face detection and tracking phase may be useful as such for the application. However, usually there is need for further analysis of the faces found. Typically some kind of normalization is done for the faces found before features are extracted for the classification phase. The most typical normalizations are alignment and normalization of the illumination. Various methods can be used for both of these. After normalization the face data is transformed to the form that can be used in the classification phase. This transformation is often called feature extraction. It may be very simple such as using image pixels directly as an input vector for a classifier, or it may be a rather complicated process such as filtering the face with Gabor wavelets.

The classification phase may include several tasks. For example, the gender, identity, age, facial expressions or ethnicity of the person can be determined. One classification task may precede the other or they can be fully separate. The benefit of preceding one task with another is that the reliability of the latter classification may be enhanced (Saatci and Town, 2006). If gender, age or ethnicity classification precedes face recognition then the recognition speed may also be increased because the preceding classification reduces the set of candidate faces for face recognition.

The actual implementation of the face analysis component is affected by the application where face analysis is used. Many of the HCI applications require continuous tracking of the faces from real-time video image. This means that computationally expensive methods cannot be used unless enough efficient hardware is available. Even though the faces were tracked continuously it may be enough to do a classification of the face gender, age, or identity from only one video frame. However, the reliability can be improved if several video frames are used for the classification (Castrillón-Santana et al., 2006; Wu et al., 2003b).

Face analysis from a video image has both benefits and disadvantages. In addition to the increased number of face images on which the classificationis based, the segmentation of the moving face (or person) from the video image using motion information can be used. On the other hand, occlusions and other challenges such as varying illumination, low image resolution and changing face pose may increase the difficulty of face analysis.

Next each face analysis task is described in detail.

2.6.1 Face Detection and Tracking

As stated above, face detection means finding all the faces from an image, and face tracking means that the faces are detected over time from video image and each face is matched between the images. One should note that there could be more than one face in the image. If we can assume that the image contains only one face then the task is simpler and the term to use is *face localization*.

Face detection is a challenging task since there are many conditions that may vary. Each person has a unique face, meaning that each face looks different. Even the face of the same person looks different depending on the time when the image is taken. For example, the age of the person, eyeglasses, beard, moustache and make-up make a difference. Pose and the orientation of the faces in relation to the camera vary. Facial expressions affect the look of the face. Face may be partially occluded by some other object, also by another face. Even the imaging conditions vary. The image may be taken outdoors in daylight, indoors in fluorescent light, or in other lighting conditions. An image may be gray scale or color image

and resolution can vary. Image may be taken with a web camera, with a high quality frame grabber or with a 3D scanner. The effects of various conditions are shown in Figure 2.15.

Figure 2.15. Examples of possible causes of problems in face detection and tracking. Faces with various orientations and poses, some occluding the others.

Face detection is the first step in a fully automatic face analysis system and thus it is one reason why it has received a lot of attention from researchers. For example, it is needed in fully automatic face recognition applications.

The evaluation of the face detection methods is important (Yang et al., 2002) as it is with the other face analysis tasks. There are public databases that can be used to evaluate face detection methods. While the detector can be trained with images containing a single face, a good image database for testing contains images with many faces taken in various conditions (Yang et al., 2002) such as the combined frontal test sets of Sung and Poggio and Rowley, Baluja, and Kanade (Sung and Poggio, 1998; Rowley et al., 1998a; Rowley et al., 1998b), the CMU profile face test set (Schneiderman and Kanade, 2000), and the UCD Colour Face Image (UCFI) Database (Sharma and Reilly, 2003). However, even these sets do not usually match the actual usage situation, and when new methods are developed there is a danger that they are tweaked to work well with the specific test set (Yang et al., 2002). Yang et al. (2002) suggest that the problem could be overcome if there were a "sufficiently large and universal test set" available or if the images used for the test were randomly chosen from a smaller test set.

Different metrics have been defined for the evaluation of face detectors, for example: detection accuracy, detection speed, required training time,

the number of training samples required for the training, and the memory requirements during training and use.

The detection rate and false alarm rate determine the detection accuracy. The *detection rate* can be defined as the ratio between the number of correctly detected faces and the number of faces in the image. For example, if there are 10 faces in the image and 8 faces are correctly detected by the face detector then the detection rate is 80%. The *false alarm* rate on the other hand determines the number of detected faces that actually are not faces. It is possible to report it as a ratio between the number of false detections and the number of face locations searched for from the image. However, the ratio depends on the face locations searched for and the number of locations searched varies between methods. Usually, the interesting fact, especially from the viewpoint of the application that uses face detection, is the actual number of false detections. Therefore, when reporting experimental results the number of false detections should be used. In addition to the detection rate and false alarm rate there are two related terms: false positive and false negative. *False positive* means that a non-face object has been detected as a face and *false negative* means a face that has not been detected.

It is also good practice to report ROC (Receiver Operating Characteristics) curves for the methods. ROC curves for four different face detectors are shown in Figure 2.16. The y-axis defines the detection rate and the x-axis the number of false alarms. The points in the curve determine detection rate with a certain number of false alarms. The bigger the area under the ROC curve is the better, because the closer the curve is to the upper left corner the higher the detection rate is while number of the false alarms is not increased. The perfect curve would be such that it goes from the lower left corner to the upper left corner and from there to the upper right corner. As can be seen in Figure 2.16 the best performing face detector in this example case is the WFS tree with sparse features by Huang et al. (2007).

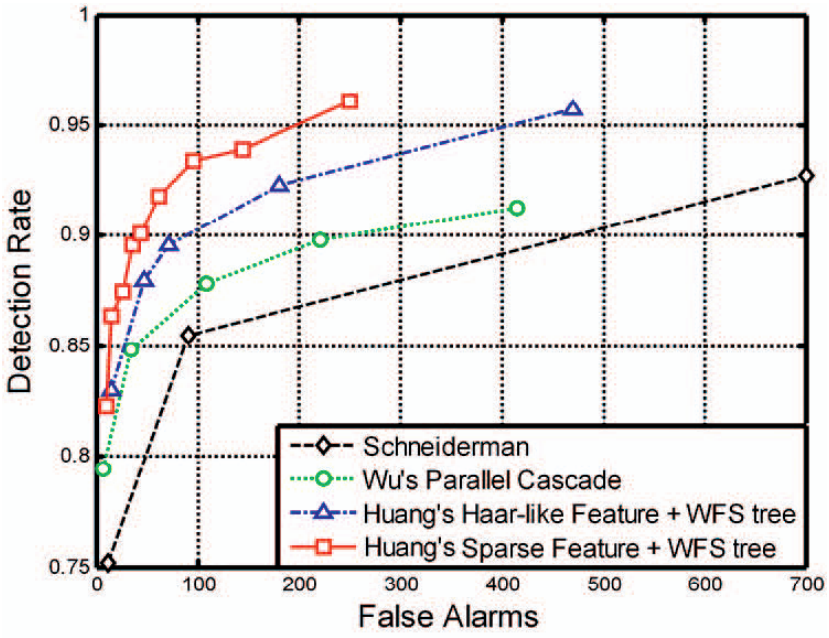


Figure 2.16. ROC curves for four face detectors. The image has been modified from the original image in the article by Huang et al. (2007).

A high detection rate and only few or no false detections are often preferable, but some applications may have other criteria. Yang et al. (2002) mention face validation application. In that case the detected face is used for identity verification. The detected face is matched to the face of the person in the database that the person being verified claims to be. The false detections will be rejected by the face verifier because they hardly match the face in the database. Therefore, false detections are not that big a problem in this case.

The detection speed is usually an important factor. There are great differences in detection speeds between detection methods. There are also some other factors than the method which determine the detection speed. For example, the image size affects the speed. Naturally, the larger the image is the longer time is required to detect faces from it. Hardware is another important affecting factor. The methods that work in "real-time" with a standard PC are the most useful in typical HCI applications since users usually expect immediate feedback on their actions. Such face detection methods are, for example, the cascaded face detector by Viola and Jones (2001) and the rotation invariant multi-view face detector by Huang et al. (2007). Naturally, as hardware is constantly becoming more powerful even such methods that have been computationally too expensive to use become usable at some point.

The number of face samples needed in training is important when considering the memory requirements for the system where training takes place. Memory may also be a problem when the trained detector is in use. The required training time and work needed to train the detector may be

an issue, for example, when selecting a detection method for a commercial product or even when doing academic research. A long training time may also be a problem with an application that uses real-time detection and on-line training would be needed (Yang et al., 2002).

One thing to consider is how precisely the face has to be located that the detection is considered correct. This issue is illustrated in Figure 2.17. If the application is interested only of the number of faces in the image or just the rough location is needed, then all the detections for the face can be considered correct. However, if further classification is done for the detected face then badly located faces may become a problem even if face alignment is used. As good data as possible should be provided to the classifiers and alignment does not necessarily work if initial face location is very inaccurate.

Figure 2.17. Image where faces are detected shown on the left and possible detections for a face shown on the right. Which detections are correct? Original image from the article by Yang et al. (2002).

Two extensive surveys have been published on face detection (Hjelmås and Low, 2001; Yang et al., 2002). These surveys also handled facial feature detection. Face detection methods can be categorized in several ways. Yang et al. (2002) divided face detection approaches into four categories: knowledge-based, feature invariant, template matching and appearance-based. Hjelmås and Low (2001) had two main categories: feature-based approaches and image-based approaches. These two categories were further subdivided into different categories.

Knowledge-based approaches use rules that are defined by a human. These rules can be based on the knowledge of face geometry (Sobottka and Pitas, 1996; Mäkinen and Raisamo, 2002; Castrillón-Santana, 2003). For example, Sobottka and Pitas (1996) used among some other rules the following two: "the eyes are located in the upper or middle part of the head" and "the ratio of the eye distance to head width is within a certain range." The rules can also be based on something else, for example on the

typical intensity distributions in a face image: "there are significant [intensity] minima for the left and right eye" (Sobottka and Pitas, 1996).

Feature invariant methods are based on the idea that facial features are detected from the image before the face. After the features have been detected they are grouped together to form a face. The facial features can be low-level or high-level. Examples of low-level features are points, edges, color, and intensity. High-level features are for example eyes, nose, and mouth. As an example of the feature-invariant method Yang et al. (2002) give a method by Yow and Cipolla (1997) that searches points and edges from an image and then groups the edges together.

In the template matching methods some sort of template is compared to image regions and the regions that correlate well with the template are considered as a face. The template can be average face calculated from thousands of frontal face examples or it can be formed from the edges of the eyebrows, eyes, nose and mouth as was done by Miao et al. (1999).

The appearance-based methods have been under intense research recently and excellent detection rates have been achieved with them (Yang et al., 2002). In appearance-based methods a face model is learned by the face detector so that it is trained with a large set of face and often non-face examples. The detector can use a neural network or several neural networks, SVM, Adaboost (Freund and Schapire, 1997) or some other machine learning method.

The cascaded face detector by Viola and Jones (2001) is a very well known appearance-based method. As its name suggests, the detector is constructed of a cascade of classifier layers. Each layer has a set of simple classifiers trained with the Adaboost algorithm. Next the detector is described in detail because it was used in the experiments reported in Chapters 4 and 5.

The detector searches faces from an image by starting from the top left corner of the image and ending to the bottom right corner of the image (see Figure 2.18.) The image where faces are detected is searched through several times with a different sub-image size each time. However, the image does not have to be resized when different size faces are searched for from the image. Instead, Haar-like features (see Figure 2.9) and integral images that are used with the detector make it possible to resize the features instead and still use the same number of calculations with all feature sizes.

When the image is scanned through, the sub-images are passed to the first layer of the face detector cascade that contains a set of Haar-like features to determine whether the sub-image contains a face or not (see Figure 2.19). If the first layer classifies the sub-image as a face then the sub-image

is passed to the next layer. This is continued until the sub-image is passed through all layers or discarded in a layer. If it is passed successfully through all layers then the final classification is a face and otherwise a non-face.

Figure 2.18. Each image is scanned from top left corner to bottom right corner using sub-images.

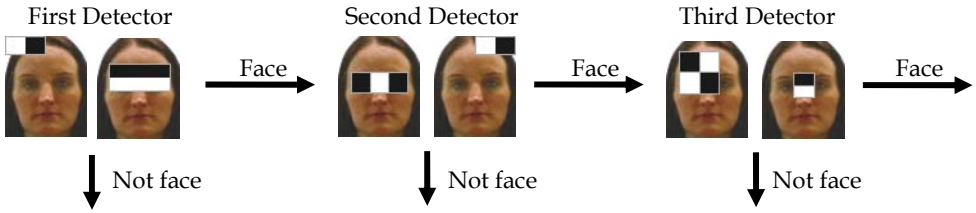


Figure 2.19. Face detector cascade. Two features are shown for each layer.

Each layer is an Adaboost classifier that makes the detection decision based on the set of Haar-like features on that layer (see Figure 2.19 and Figure 2.9). The original cascaded detector used four types of rectangular features and each type is shown in Figure 2.9. Later more feature types have been proposed (Lienhart and Maydt, 2002). The main idea of the cascade is that easily recognizable non-face sub-images can be rejected at the earlier layers which have fewer features than the later layers. The original face detector by Viola and Jones (2001) had 32 layers and the first layer had 2 features while the last 20 layers had 200 features. This cascaded structure brings efficiency because it takes less time to process a sub-image in the earlier layers and most of the sub-images are rejected at the first layers. The detector can reliably detect frontal faces in gray scale images with over 90% detection rate at the frequency of 15 frames per second with a typical PC.

Face tracking means following a face or faces for a certain period of time. Many of the algorithms that can be used for object and person tracking can also be used for face tracking. Before a face can be tracked it needs to be detected at least once. However, after the initial detection, detection and tracking can also take place simultaneously. For example, the system by

Yang et al. (2006a) had separate face detection and tracking modules. The system switched between the modules when needed to improve robustness of tracking.

After the face has been detected it can be tracked by searching the face, for example, from the locations close to the previous location. If motion estimation is used, then this can be taken into account when deciding the locations where the face most probably is. A model of the detected face can be used when tracking the face. The model can be a template image of the face or template images of some facial features (Colmenarez et al., 1999; Nakanishi et al., 2002) in the previous face location, color distribution of the detected face (Yang and Waibel, 1996; Bradski, 1998; Yang et al., 2006a), intensity histogram of the detected face, active contour model (Sobottka and Pitas, 1996), or something else. The searched location that gives the highest probability to the model is then selected as the new face location. Naturally, if the probability is very low then the system may decide that the tracked face cannot be found or some other means may be used to find the face.

2.6.2 Facial Feature Detection and Tracking

Facial feature detection means the detection of such high-level features as eyes, nose, chin, mouth, mouth corners, and lips or low-level features such as facial edges or points, for instance.

Facial feature detection is often an integral part of face detection. Those knowledge-based face detection methods that use rules based on facial feature locations require that facial features are detected before the face location is determined. However, facial feature detection can also be separate from face detection.

There are variations in facial feature appearances between people, which cause challenges for facial feature detection and tracking. In addition, especially mouth appearance changes a lot when a person speaks and shows expressions. Eye appearance is also changed within expressions but also when we open and close the eyelids.

The same approaches can be used to detect and track facial features as can be used to detect and track faces. Some of the knowledge-based face detection methods that use locations of the facial features can also be thought of as knowledge-based facial feature detection methods. The method by Mäkinen and Raisamo (2002) is one such example. An example of the appearance-based method is the system presented by Niu et al. (2006) that had a cascade for eye detection. The cascade was somewhat different from the cascade by Viola and Jones (2001) used to detect faces but both systems used the Adaboost algorithm. Templates have also been experimented with. Rurainsky and Eisert (2003) used deformable templates to detect pupils, mouth corners and inner mouth lip lines from

face images. Gorodnichy and Roth (2004) tracked nose with a method that used 3D-templates. The 3D-templates were created from two images captured by two web cameras positioned next to each other.

Facial feature detection can be used in face normalization, as will be described shortly, and facial feature tracking can be used in gesture recognition for instance. Changes in head pose can be determined and head nodding and shaking can be recognized by tracking eyes. Gaze tracking is described in Subsection 2.7.1 because it is a special case of facial analysis, it has been under intense research, and there is specific hardware available for gaze tracking.

2.6.3 Face Normalization and Alignment

After or even before the face and possibly facial features have been detected some kind face normalization is usually needed. Most systems include some sort of image intensity normalization. The face images may have been captured in different lighting conditions and with different camera parameters and normalization improves the robustness of the system. Histogram equalization described in Subsection 2.4.1 can be used for minimizing the differences between the imaging conditions. This is a common procedure especially with neural networks. It has been used with neural network based face detection by Sung and Poggio (1998), Roth et al. (2000) and by Rowley et al. (1998), for instance. However, histogram equalization is not restricted to neural networks. For example, the SVM based face detection system by Heisele et al. (2000) and the Adaboost based face detection and gender classification system by Wu et al. (2003b) included histogram equalization.

In addition to histogram equalization there are other methods that can be used for intensity normalization. For example, the cascaded face detector by Viola and Jones (2001) used variance normalization to create unit variance for the sub-window intensities during image scanning. To make the normalization faster it was done for the feature values instead of the pixels and the variance was calculated using the integral images. In addition to histogram equalization Sung and Poggio (1998) used illumination gradient correction to reduce the effect of heavy shadows. The illumination gradient correction was done so that the intensity plane that best fitted the face image was subtracted from the image. Choi et al. (2007) presented a method for shadow compensation that was specifically designed for faces. It used knowledge of how shadows appear on faces when the direction of the light changes. Other methods have also been proposed for shadow compensation.

In addition to intensity normalization, the location of the face can be normalized so that the face sub-images that are used as input to the following face classification tasks are more consistent in location and face shape. Most of the face detection methods find the rough location of the

face and face alignment can be useful in these cases. One possibility is to use located facial features. For example, faces can be rotated so that the eyes are vertically aligned. Furthermore, faces can be resized and the place of the sub-images translated so that eyes are at the same location in all sub-images. The shape of the face can also be modified, for example, by stretching or squeezing the face so that, in addition to the eyes, the mouth is also at the same location in the sub-images. Naturally, for the alignment to be useful, it has to be exact. Otherwise it just makes the face analysis slower while the classification rates remain unchanged or even get worse as the experiments described in Chapter 5 show.

For alignment facial features detected by the face detection method can be used if the method locates them. Another possibility is to use facial feature detection methods. For example, Huang and Wechsler (1999) and Niu et al. (2006) presented methods for eye detection. A third possibility is to use shape models that are described next.

Cootes and Taylor (2001) proposed two different shape models: Active Shape Models (ASM) and Active Appearance Models (AAM) that can be used for modeling the face shape (or some other object shape). With both of the models face model is learned from example faces. The example faces are annotated by defining some locations of facial features and edges on them manually. An example of an annotated face image is shown in Figure 2.20. After the faces have been annotated the model is trained to model variations between the faces based on the edges defined. In addition to the edges, both models learn the appearance (texture) of the object. However, ASM uses only the appearance around each defined point while AAM uses the appearance of the whole face. After the model has been created it can be fitted to other faces. An example of fitting an AAM model to a face is shown in Figure 2.21.

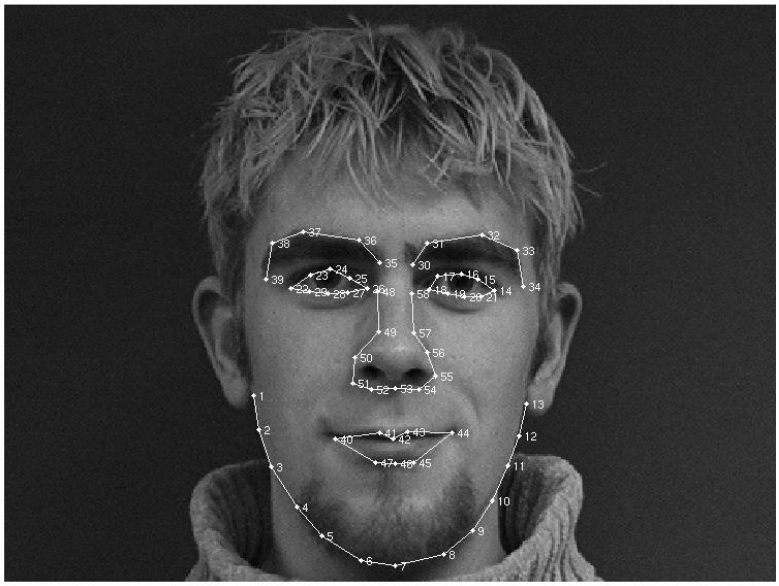


Figure 2.20. Example of the annotated face image. The face is from the IMM database (Stegmann et al., 2003).

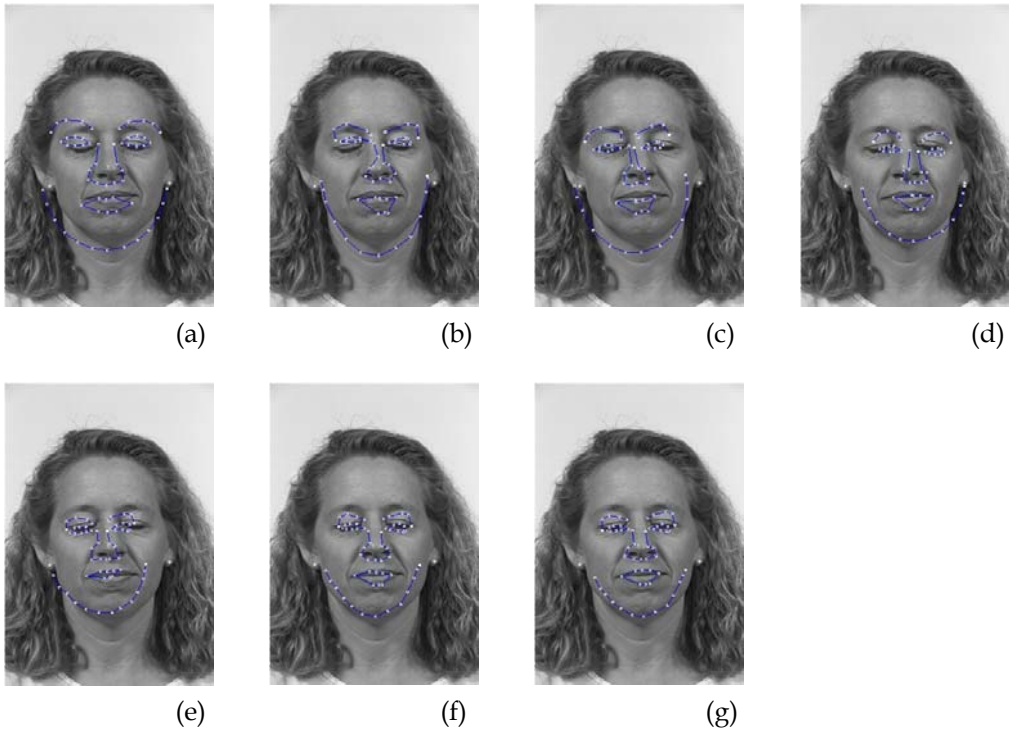


Figure 2.21. Example of fitting an AAM model to a face not used in model training. Shapes model (a) initially, (b) after 1 round, (c) 2 rounds, (d) 3 rounds, (e) 4 rounds, (f) 5 rounds, and (g) after 100 rounds. The face is from the FERET database (Phillips et al., 1998).

Other shape models have been proposed, too. Li et al. (2002) presented the Direct Appearance Model (DAM) that they claimed to have two advantages over AAM. They stated that DAM improves the convergence and accuracy of the model fitting because in AAM shape and texture are used together while in DAM face texture is used directly to estimate shape of the face. The other improvement was remarkably smaller memory requirement during model learning.

Simple Direct Appearance Model (SDAM) (Xiao, 2002; Wang et al., 2003) was suggested for approximate face alignment before finer alignment is done by a more complex method such as AAM. The SDAM algorithm extracts the centers of the eyes and mouth and initializes locations of other feature points based on the three extracted features (Wang et al., 2003).

Tamminen and Lampinen (2006) presented a sequential Monte Carlo based method that used shape and appearance together and separately in the model. They did experiments with unoccluded and partially occluded faces and showed that their method is very promising, especially when used with partially occluded faces because it could converge very accurately even in these cases.

2.6.4 Face Recognition and Verification

In addition to face detection the other face analysis topic that has received a lot of attention from researchers is face recognition. Zhao et al. (2003) defined *face recognition* in their survey as follows: "given still or video images of a scene, identify or verify one or more persons in the scene using a stored database of faces."

Although even commercial systems on face recognition exist, the face recognition field still faces many challenges. General face analysis challenges that also hold for face recognition were presented in subsection 2.6.1. Zhao et al. (2003) mentioned that major challenges with face recognition are "illumination, pose and recognition in outdoor imagery." These major challenges are illustrated in Figure 2.22.

Figure 2.22. Photos of the same person's face taken at different times, in different lighting conditions, and with different facial expressions and in various poses.

The history of computer based face recognition goes back to the 1970s (Kelly, 1970; Kanade, 1977). Since then numerous studies have been

carried out and several face recognition literature surveys have been written (Samal and Iyengar, 1992; Chellappa et al., 1995; Zhao et al., 2003; Scheenstra et al., 2005; Tan et al., 2006).

The reason why face recognition has been under intense research in recent years is partly in the amount of commercial applications. One application area for face recognition is security and surveillance and some commercial systems already exist. The other reasons for popularity are that hardware has become efficient enough and the importance of security related applications has increased (Zhao et al., 2003). The complexity of the face as an object to be analyzed makes face analysis, including face recognition, interesting from the research point of view, as stated above.

In the face recognition field, there have been efforts to achieve comparable results of different methods. Probably the best known public database for face recognition is the FERET database (Phillips et al., 1998). It contains 14,051 face images of 1,199 individuals. Depending on the person there are images with different facial expressions and poses. In addition, there are images where lighting conditions have been changed or images were taken at different dates. The FERET database has later been updated with a color version containing 11,338 images of 994 individuals and is largely same as the original FERET database but for color format.

There have also been several Face Recognition Vendor Tests (FRVTs) in years 2000, 2002, and 2006 (Blackburn et al., 2001; Phillips et al., 2003; Phillips et al., 2007) carried out by the US government organization National Institute of Standards and Technology (NIST) that evaluated commercial and prototype face recognition technologies. As a part of the latest evaluation, the US government provided Biometric Experimentation Environment (BEE) that made it easier for an experimenter to evaluate the methods and for researchers to prepare their methods for the evaluation.

As another example, The CSU Face Identification Evaluation System (CSU, 2003) provides implementations of some well known algorithms (PCA, PCA combined with LDA, Bayes, and Elastic Bunch Graph Matching with Gabor jets) and protocols to carry out face recognition experiments.

Although there are challenges in face recognition the technology is constantly improving. The face recognition accuracy in the last FRVT 2006 evaluation was ten times better than in the FRVT 2002 evaluation four years earlier (Phillips et al., 2007).

2.6.5 Gender Classification

Next a comprehensive description of the gender classification works is given because gender classification has a core role in the experimental part of the thesis.

Gender classification research can be categorized in several ways. One possibility is to divide the works based on the viewpoint that gender classification is approached from. For example, BenAbdelkader and Griffin (2005) divided works into those that approach the topic from a psychological and neurophysiological viewpoint and to those that approach the topic from a computer vision viewpoint.

A second possibility would be to categorize works based on the machine learning techniques used in the work. In this case, one category could be for the works that use SVM for gender classification, the other could be for the works that use a neural network and so on. However, in many works several different machine learning techniques have been used (see Appendix 1), so categorization based on one technique would be difficult in these cases.

A third possibility would be to divide the works on the basis of characteristics of the gender classification system used. For example, a division could be made into two categories: gender classification of faces manually extracted from the images and gender classification with automatic face detection. The classification task in the latter category is more challenging because some non-faces may be detected and correctly detected faces may be badly aligned. On the positive side the results of the latter category are closer to the real applications that include automatic face detection.

In the following descriptions of the previous gender classification studies no specific categorizing has been used. The summary of the existing gender classification works is given in Appendix 1 where the above categorization has also been taken into account.

The neurophysiological and psychological research on gender classification started before the computer vision research on the topic. In neurophysiology and psychology the interesting topics have been what facial information humans use to perform the gender classification task (Abdi et al., 1995; Baudouin and Humphreys, 2006; Bruce et al., 1993; Buchala et al., 2005; Burton et al., 1993; Cheng et al., 2001; Edelman et al., 1998; Fellous, 1997; O'Toole et al., 1998;) and what the connection of the gender classification is to identity recognition and other face analysis tasks (Baudouin and Humphreys, 2006; Cheng et al., 2001; Clutterbuck and Johnston, 2004; Tranel et al., 1988; O'Toole et al., 1998).

The computer vision research on the topic started at the beginning of the 1990's. The research has been partly motivated by psychology but it has applications in other fields including HCI.

The very first computer vision studies on gender classification were reported simultaneously by Cottrell and Metcalfe (1990) and by Golomb et

al. (1990). The studies were quite similar. Auto-associative networks and perceptrons were used in both and the aim was to have a working gender classifier. In addition, Cottrell and Metcalfe (1990) also considered face and facial expression classification.

Like the first two studies, most of the studies after that have presented novel algorithms or algorithm combinations for gender classification (BenAbdelkader and Griffin, 2005; Brunelli and Poggio, 1995; Buchala et al., 2004; Castrillón et al., 2003; Costen et al., 2004; Graf and Wichmann, 2002; Gutta et al., 1998; Iga et al., 2003; Jain and Huang, 2004a, 2004b; Kim et al., 2006; Lian and Lu, 2006; Lyons et al., 2000; Moghaddam and Yang, 2000; Shakhnarovich et al., 2002; Sun et al., 2002a, 2002b, 2006; Tivive and Bouzerdoum, 2006; Walavalkar et al., 2003; Wilhelm et al., 2005; Wiskott et al., 1995; Wu et al., 2003a, 2003b). The studies proposing new algorithms aim at better classification rates.

With the exception of three studies (Baluja and Rowley, 2007; Kim et al., 2006; Shakhnarovich et al., 2002), the best classification rates have been achieved with support vector machines (SVMs) when SVM and other machine learning methods have been compared (BenAbdelkader and Griffin, 2005; Buchala et al., 2004; Castrillón et al., 2003; Costen et al., 2004; Jain and Huang, 2004b; Moghaddam and Yang, 2000; Sun et al., 2002b, 2006; Wu et al., 2003a). The main contribution of these studies has often been a novel feature extraction method.

However, numerous methods have also been suggested for classification with or without comparison to SVM, such as, HyperBF networks (Brunelli and Poggio, 1995), elastic graph matching (Lyons et al., 2000; Wiskott et al., 1995), RBF network (Gutta et al., 1998; Moghaddam and Yang, 2000), LDA (BenAbdelkader and Griffin, 2005; Costen et al., 2004; Jain and Huang, 2004a, 2004b; Kim et al., 2006; Moghaddam and Yang, 2000; Sun et al., 2006), Adaboost (Shakhnarovich et al., 2002; Sun et al., 2006; Wu et al., 2003a, 2003b;), Gaussian process classifiers (Kim et al., 2006), and so on. For example, Kim et al. (2006) found that their method outperformed SVM when cross-validation was used. Their method was also useful in improving the SVM performance.

The classification rate depends heavily on the face data quality. Over 90% or even near 100% classification rates have been achieved with high quality data when faces have been manually extracted and aligned. However, around 80% classification rates have been achieved with lower quality images when automatic face detection has been used (Shakhnarovich et al., 2002; Wu et al., 2003b; Castrillón et al., 2006).

In addition to research on finding the best machine learning method for gender classification, there have been other types of studies. Yang et al. (2006b) did a comparative study by combining various face normalization

methods with various gender classifying algorithms. They also used automatic face detection in the study. The best face image resolution to be used has also been studied. Gray et al. (1995) performed experiments with neural network classifier for 10*10, 15*15, 22*22, 30*30, and 60*60 size face images. Only the 10*10 size gave clearly poorer results than the other resolutions. Tamura et al. (1996) showed that gender classification is also possible for very low resolution face images. Some other studies (Wu et al., 2003a; Wu et al., 2003b; Buchala et al., 2004; Buchala et al., 2005; Kim et al., 2006) have also included experiments with various image resolutions but the focus has been on something else, for example, on the classification algorithm.

The usefulness of various facial features and face parts has also been studied. Hayashi et al. (2002) used face color and wrinkles to classify gender and age. Kawano et al. (2004) compared gender classification performances for the whole face, jaw, lips, nose, and eyes. Buchala et al. (2005) published a similar study. They used whole face, eye area, and mouth area separately and together for the classification. Lapedriza et al. (2006) presented a system to extract facial features in uncontrolled environments and did gender classification experiments with the system. Lu et al. (2006) combined facial intensity and range data and did experiments both for gender and ethnicity classification. They found that intensity and range data together resulted in better classification rate than either alone.

Nishino et al. (2004) used face thermography for the gender classification. Although the classification rate of 77.5% that they achieved is not as good as can be obtained with methods using photographs for the classification, it is still clearly better than the rate achieved by chance and the method could be useful in certain situations. The authors mention a biometric authentication situation where a person whose gender is to be recognized has intentionally changed his or her look to the other gender. Thermal infrared imagery has also been used for face recognition, for example very recently by Buddharaju et al. (2007).

Ueki et al. (2004) examined the usefulness of clothing in addition to the face and hair for gender classification and were able to improve the performance of the classifier by using clothing images as well. The clothing images were taken from the neck area and tie and décolleté were looked for. Saatci and Town (2006) studied if gender classification performance could be improved by classifying expression before gender so that each expression has its own gender classifier. The classification rates were improved for facial expressions when gender classification preceded the facial expression classification, but the gender classification rate decreased if a gender classifier specific to the facial expression was used. However, Saatci and Town (2006) hypothesized that this decrease in

the gender classification rate was due to the small size of the training image set. Naturally, it might be useful to use other kinds of combinations, too. For example, if the approximate age of the person was classified first and there was a separate gender classifier for each age group, then it might be that the classification rate would increase. Anyhow, there is proof that humans classify the gender of adult and child faces using features specific to that age group (Cheng et al., 2001).

In a fairly recent study Castrillón et al. (2006) experimented with an evolving gender classifier. Ultimately the system learned while it was on-line and novel faces were presented to it. At the end the system did not perform as well as their off-line classifier, but Castrillón et al. (2006) evinced several possible reasons for this. One of the reasons was that faces were aligned using eye locations and eyes were located automatically with the on-line classifier. As shown in Chapter 5 of this thesis, the automatic alignment is indeed one likely reason why the off-line classifier outperformed the on-line classifier.

2.6.6 Facial Expression and Gesture Classification

Emotions greatly affect human behavior and it has been shown that emotions affect our memory and other mental processes (Lewis and Critchley, 2003). For example, humans cannot make decisions if their emotional functions have been damaged (Damasio, 1994). Emotions and facial expressions also have an extremely important role in communication between humans. Facial expressions mirror our emotions, mental activities, and physiological activities (Fasel and Luetin, 2003), and help other people to understand us.

When humans interact with computers emotions are also present. When there are some problems with the application we are using we may become annoyed and eventually even angry. There are even extreme cases where users have broken their computers after getting angry. Naturally, when humans are communicating with each other through computers emotions and facial expressions are also involved.

There have been efforts to define facial expressions precisely. Ekman and Friesen (1978) developed the Facial Action Coding System (FACS) that has become the most well known system for describing facial expressions. It defines 46 facial action units (AUs) that are based on facial muscle movements. All facial expressions can be defined using this system.

Ekman and Friesen (1971) defined six basic emotions and their corresponding facial expressions that seem to be universal among all cultures in the world. These emotions are: happiness, sadness, fear, disgust, surprise, and anger. However, in addition to these basic expressions there are a lot of expressions that do not belong to the basic ones (Tian et al., 2001) and there are differences among cultures in

showing and interpreting facial expressions (Fasel and Luetin, 2003; Matsumoto, 1993). Facial expressions can also be caused on purpose, in which case they can be considered gestures.

Automatic facial expression analysis has attracted a lot of attention among researchers although less than face detection and recognition. Pantic and Rothkrantz (2000) and Fasel and Luetin (2003) have reported surveys on facial expression analysis. In addition to the general challenges in face analysis including lighting conditions, occlusions and so on, there are challenges specific to automatic facial expression classification. In the case of facial expressions we are measuring intra-personal variations in face. However, there are still differences in facial expressions and their intensities between different individuals. Because there are about 7,000 combinations of AUs (Ekman, 1982) it is also hard to develop a system that can take all these combinations into account. Since facial expressions are dynamic they have temporal characteristics. Each expression has onset (attack), apex (sustain), and offset (relaxation). These can be analyzed from an image sequence but not from a single image.

Fasel and Luettin (2003) pointed out that the automatic facial expression classification field has still a lot of research to do. Most of the existing systems classify expressions directly to the six basic emotions (Fasel and Luettin, 2003). However, in practice expressions are almost always more complex and many expressions are caused by physiological activities or are used as gestures. It is also possible to classify expressions using the AUs. In this case the classified AUs are interpreted using a facial expression dictionary. One example of such a dictionary is the Facial Action Coding System Affect Interpretation Dictionary (FACSAID) by Ekman et al. (1998). Only a few existing systems use these dictionaries in the classification.

Most of the systems assume frontal faces and allow only small head movements between the video frames, and manual initialization is often needed. In addition, there are only a few systems (Lien, 1998; Lisetti and Rumelhart, 1998; Fasel and Luettin, 2000) that classify intensities of facial expressions.

2.6.7 Age Classification

The age of the person can also be approximated from the face and there has been some computer vision research on the topic (Kwon and Lobo, 1999; Iga et al., 2003; Lanitis, 2002; Lanitis et al., 2002, 2004; Ueki et al., 2006; Wilhelm et al., 2005).

Age classification can be done based on the ratios between facial features and on wrinkles that appear in the face as a person ages (Kwon and Lobo, 1999). Babies and young children have different bone structure than adults. For example, the eyes are located in a higher position on adult faces and

the nose moves towards the mouth away from eyes when a person gains age (Kwon and Lobo, 1999). After person reaches adulthood the bone structure does not change any more.

The specific challenges with age classification are that the age of the person is hard to predict exactly because facial appearance changes slowly when a person is aging and this change in appearance is somewhat person dependent. Further, bone structure and wrinkles are also affected by other factors than age alone. For example, identity and ethnicity affect the bone structure and wrinkles that appear within aging can also be caused by facial expressions.

Kwon and Lobo (1999) used facial feature detection and wrinkle detection to classify age to the three age groups: babies, young adults and seniors. They carried out experiments with the faces of 5 babies, 5 young adults, and 5 seniors. Using the locations between detected facial features and the number of wrinkles on the face they determined the age group of the face. Classification was successful for all 15 faces.

Iga et al. (2003) classified faces into five age groups: 15-24, 25-34, 35-44, 45-54, and 55-64 years. The classification was based on face texture, on facial feature locations, on Gabor features, on skin color, and on hair. SVM classifiers were trained for these face properties and age classification was decided by votes from the classifiers. In addition, the gender of the face was recognized before age classification and there were separate SVM classifiers for both genders.

Wilhelm et al. (2005) investigated the classification of age, gender, facial expressions, and identity. They classified age to five age categories between 10 and 60 years. The best performing classifier they experimented with was a multi-layer perceptron when AAM was used to model the shape and texture of the faces. The classification rate was slightly over 40%. However, they noted that if they had used only two categories then young and senior people would have been distinguished quite successfully.

Ueki et al. (2006) presented a classifier based on two phases using 2D-LDA and LDA to classify age. The benefit of their classifier is that it is robust under various lighting conditions. Ueki et al. (2006) experimented by using age ranges of 5 years, 10 years, and 15 years. The respective classification rates for each range were 46.3%, 67.8%, and 78.1%.

Besides classification to age groups it is also possible to estimate the exact age of the person. The studies by Lanitis (2002) and by Lanitis et al. (2004) achieved roughly a 5-year mean error in the experiments where they used face images of people aged between 0 and 35 years. Lanitis (2002) investigated the usefulness of the whole face including hair, internal face, lower face including mouth, and upper face including eyes to the

classification. He created a statistical model to describe shape and grey-level variations in the face images of different ages. The upper part of the face gave the best age estimation with 3.83 years mean error for the 80 test face images.

Lanitis et al. (2004) experimented with four different age classifiers: quadratic classifier, shortest distance classifier, multi-layer perceptron, and self-organizing map. They also experimented with four setups: single step, age specific, appearance specific, and appearance specific combined with age specific setup. In the single step setup the age of the face image was classified directly. In the age specific setup there was a classifier that classified the face to the one of the three age ranges: 0-10, 11-20, and 21-35 years. After classifying the face to an age group it was classified more exactly with the classifier specific to the age group. In the appearance specific setup the faces were clustered so that similar looking faces were in the same clusters. When a novel face was classified, it was first classified to a cluster and after that the age was classified with the classifier specific to that cluster. The final setup combined age specific and appearance specific setups, so that the face was first classified to a cluster, then to the specific age group, and finally to the specific age within the selected appearance cluster and age group. The best performance, 3.82 years mean error, was achieved with the combined setup when a quadratic classifier was used.

There is also a database specifically intended for research on face based age classification: FG-NET Aging database (FG-NET, 2007).

2.6.8 Ethnicity Classification

Ethnicity classification has also received some attention from researchers. The focus has been on experimenting with different machine learning techniques on ethnicity classification. For example, RBF networks (Gutta et al., 1998), Gabor wavelets and LDA (Lyons et al., 2000), Adaboost classifier with Haar-like features (Shakhnarovich et al., 2002), Gabor wavelets and SVM (Hosoi et al., 2004), LDA (Lu and Jain, 2004), and SVM (Lu et al., 2006) have been used. Over 90% classification rates have been achieved in the most of the above studies.

Defining ethnic groups is somewhat artificial, since many people have backgrounds in various ethnic areas and various groupings can be used. It can also be seen in the groupings used in the studies on ethnic classification. Gutta et al. (1998) used Caucasian, Asian, African, and Oriental groups. Lyons et al. (2000) used East Asian and non-East Asian groups. Shakhnarovich et al. (2002), Lu and Jain (2004), and Lu et al. (2006) used Asian and non-Asian groups. Finally, Hosoi et al. (2004) used three groups namely Asian, Caucasian, and Negroid. Nevertheless, although the definition of groups is somewhat challenging and perfect classification results cannot be expected, it can still be done and classification can be

useful in certain applications. The application examples are described in Chapter 7.

2.7 MULTIMODAL INTERACTION

Modality is a sense that is used in human-computer interaction (or in human-human interaction) while *communication channel* conveys information using a modality in a specific way. A channel has direction from human to computer or from computer to human. The former direction is denoted as input and the latter as output. Besides vision there are other modalities such as audio and haptic modalities. Examples of communication channels are face analysis, gaze tracking, speech recognition, display, auditory information channel, and keyboard. Although computer vision and face analysis have the main focus in the thesis other modalities and communication channels are equally important in HCI. Each channel has strengths and weaknesses. Since face analysis is an input communication channel, comparison to the other input channels is useful. The strengths and weaknesses of various input channels are shown in Table 2.1.

It is often beneficial to use several modalities and channels in parallel. Sound can be used in awareness systems to provide information on the user's surroundings and to attract the user's attention. Face analysis and gaze tracking can be used to observe the user's identity, attention and mental processes. The user can feel and examine the objects using the haptic modality. The user can command the system with speech and the system can perceive the user's location and emotions from speech. There are also less commonly thought modalities such as smell and taste, and biosignals including EMG, ECG, and EEG that fit specific applications.

Input modalities can be used in the same system for separate tasks or for the same task. In the latter case they may be alternative to each other, in which case the user can choose which modality to use or they may be interleaved, in which case the task is accomplished using many modalities. Disabled people especially can benefit of the alternative modalities. For example, visually impaired people can use haptic and auditory modalities in place of vision in many tasks. A concrete example of the task where multiple modalities or rather multiple input channels can be interleaved is movie rating. After a user has watched a movie he can use a hand gesture to point to the screen and say "rate this" with an expressive gesture on his face. The movie is rated on the basis of the expression shown on the face: a strong smile meaning a great movie, a neutral face meaning an ok movie, and disgust meaning a bad movie.

Table 2.1. Strengths and weaknesses of various input communication channels.

Input channel	Strengths	Weaknesses
Face analysis	Natural, face reveals identity, gender, age, expressions, ethnicity, health information, and general direction of gaze	Requires camera, user has to face the camera, limited reliability in uncontrolled environments, privacy, very limited usage in command-and-control tasks
Gaze tracking	Target of attention easily acquirable, fairly natural with non-wearable trackers, possibly only communication channel for severely paralyzed person	Requires special hardware, requires calibration, limited head movement, limited usage in command-and-control tasks
Speech recognition	Natural, efficient for communicating explicit information, expressive, users often prefer speech to writing	Privacy, not reliable in noisy environments, serial, may require user to memorize commands, limited grammars and vocabularies
Keyboard	Accurate, usually the best choice for writing, useful in command-and-control applications with mouse	Non-natural, perceived information of the user is very limited, hard and slow to convey emotions
Mouse	Accurate, designed for desktop paradigm, very useful in command-and-control applications	Non-natural, perceived information of the user is very limited, very hard to convey emotions
Game controllers	Suitable for various types of game applications, option for mouse and keyboard, often possibility to improve user's immersion and motivation	Use may require learning, use often requires both hands, somewhat non-natural
Bio-electrical signals	Provide information on the user that might otherwise be unavailable	Requires special equipment, often non-natural, data may be inaccurate, interpretation of the signal may be hard

Clearly it is often most beneficial to use modalities together. Next an overview of some modalities and communication channels is given and their contribution to face analysis is considered.

2.7.1 Eye Tracking

Face detection and head pose recognition reveal the direction in which a person is facing. For example, when a person sits on a chair in her office this can be detected with a camera above the computer display. When face detection is combined with face recognition a computer can be unlocked and a user's desktop loaded. However, after this eye tracking can be used to observe the user's focus of attention.

Eye tracking currently usually happens with a specific eye tracker device, such as Tobii eye trackers (2007). Tobii eye trackers track eyes using infrared light emitted from infrared illuminators and reflected from the pupils to the infrared sensors. The eye tracker can be integrated on the display or it can be a separate box. There are also trackers that require head worn equipment but because their usage is restrictive such trackers are being replaced with other types of trackers.

Morimoto and Mimica (2005) reviewed recent eye tracking technology. Some of the techniques reviewed worked without calibration and allowed fairly free head movements. Most systems still used infrared light but, for example, Beymer and Flickner (2003) combined 3D face detection with eye tracking. The eye tracking was done with two narrow field of view (FOV) cameras steered according to face detection. The system required one time calibration for each user. They achieved 0.6° gaze direction accuracy when a person was at a distance of 0.6 meters from the display.

Besides determining focus of attention, eye tracking can be used for detecting blinks, winks, staring, and other natural eye behavior (Selker et al., 2001; Hyrskykari et al., 2005). Changes in pupil size may indicate that a user has experienced a strong emotion (Partala and Surakka, 2003). The distance of the user from the eye tracker that is usually at the same location as the display can also be measured.

2.7.2 Haptics

Humans perceive haptic sensory information using skin, muscles, tendons, joints, and mucosae (Klatzky and Lederman, 2002). Humans use haptic perception passively and actively. They can examine objects by touching them. For example, texture, temperature, hardness, and weight can be examined. Weight can also be measured by lifting or by otherwise moving the object. These are properties that are hard or even impossible to perceive by vision or other senses. On the other hand vision is more useful in perceiving the geometry of an object (Klatzky and Lederman, 2002). There are also some common features with visual and haptic perception:

spatial representation may be same and visual and haptic memories are connected (Klatzky and Lederman, 2002).

Currently haptic modality is mainly an input modality in the current desktop computer systems. Haptic interaction is present when we use a keyboard or mouse since we use them by touch. However, haptic interaction can be enhanced by different means. There are mice, game pads, and joysticks with tactile feedback and with force feedback. Mobile phones may vibrate when someone is calling. Tactors can be attached to skin giving haptic feedback on specific situations. The Phantom device is an example of a 3D force feedback device (see Figure 2.23).

Figure 2.23. Example of haptic interaction. A user uses a Phantom device (SensAble, 2007) with a Reachin display (Reachin, 2007) to navigate in 3D space.

It is also possible to use haptics as a more expressive input modality by equipping a computer with pressure sensors, force sensors, motion sensors, and thermal sensors. Nintendo Wii (Nintendo, 2006) is a very well known game console that includes controllers with motion sensors. The controller is also equipped with tactile feedback. The gestures that a user makes while holding the controller are recognized and, for example, the swing of the tennis racket can be imitated. An alternative way to recognize gestures would be to use computer vision. However, the application determines which technique should be chosen and it may be useful to use both simultaneously in some situations.

2.7.3 Speech and Non-speech Audio

Sound is an important input and output modality for humans. Speech and vision are the main communication channels when humans communicate face to face. It has been claimed that speech is the dominant way to communicate and exchange information (Huang et al., 2001, pp. 1). While the attention of the other person and most of the expressions are

perceived through vision, almost all explicit information and a part of the emotional information and expressions are conveyed through speech.

Since the aim in perceptual user interfaces is to make communication natural, both speech recognition and speech synthesis are needed. Current speech recognition technology is fairly reliable when vocabulary is limited and the environment is not noisy. For example, it is already used in many mobile phones and there is commercial speech recognition software available. Speech synthesis is even more mature technology. However, there are still some problems with it: synthetic speech may sound unnatural and, for example, long series of numbers may be problematic for the synthesizer. The option for synthetic speech is recorded speech but it can only be used if all possible phrases are known beforehand that is not the case in completely natural interfaces.

Speech recognition technology uses partly the same machine learning and pattern recognition algorithms as computer vision. For example, hidden Markov models (HMMs) that are commonly used in speech recognition (Duda et al., 2001, pp. 128) are also used in human activity analysis (Brashear et al., 2006; Rigoll et al., 2000; Starner and Pentland, 1995) and face analysis (Lien, 1998). Other common techniques are neural networks and linear discriminant analysis (LDA).

Non-speech audio is also a useful way of conveying information. Humans perceive sounds often without even noticing it. We can tell that there is someone moving in the corridor when we hear his footsteps and sometimes we may know who the person is without seeing him or her. On the other hand, sound can also alert us. For example, we focus our attention immediately to the direction of a crashing sound.

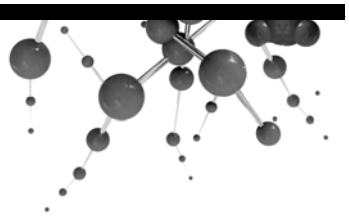
Non-speech audio has been used as an output modality with computers. For example, in desktop environments there are often sounds defined for different events: arriving email, error message, and so on. These event sounds sometimes do not have a natural counterpart, in which case a user has to learn the meaning of the sound. Such sounds are called earcons (Brewster, 1993). If the sound has a natural counterpart it is called an auditory icon.

There are also ways to convey information using sounds that have been used less with computers: music, soundscapes, and sonifications. Music is commonly listened to while using computer but it is usually listened to only to entertain. However, information specific to the computer usage situation can be presented by varying musical properties such as pitch, timbre, and rhythm. Soundscape creates an auditory picture of the environment using background and foreground sounds. It can be used to "encompass environments and create atmospheres and identities to locations, and tie different interaction elements together" (Kainulainen et

al., 2007). Sonifications represent data and data relations as sounds similarly to the visualizations.

2.8 SUMMARY

The purpose of this chapter was to give sufficient background knowledge to understand the following chapters. Knowledge of machine learning and pattern recognition techniques helps to understand the experiments described in the next three chapters. This knowledge is also useful for becoming familiar with of the tools in Chapter 6 that were developed to carry out the experiments. Knowledge of the state-of-the art in various face analysis areas is needed when focusing on the existing and possible future applications described in Chapter 7. Since many of the applications can benefit when other modalities are combined with vision the background on multimodal interaction is then helpful.



3 Face and Facial Feature Detection

3.1 INTRODUCTION

As described in Chapter 2, face detection is often the first task in automatic face analysis. Many rather reliable methods have recently been proposed for face detection (Phimoltares et al., 2007; Schneiderman and Kanade, 2000; Viola and Jones, 2001). However, there is still room for improvement in terms of detection reliability and efficiency.

While humans can tolerate delays to some extent when they use computers, in natural interaction responses should be fast. Therefore it is important that face analysis techniques to be used in perceptual user interfaces are also as fast as possible.

I have developed a novel face detection method and carried out experiments with it (Mäkinen and Raisamo, 2002). It can analyze images sized 320*240 pixels with over 20 Hz rate on a Microsoft Windows system with 1.14 GHz AMD Athlon CPU and with 256 MB of memory. The advantage of this method is speed when compared to many other face detection systems and 20 Hz is enough for the method to be used in perceptual user interfaces (Turk and Kölsch, 2004).

The method by Sobottka and Pitas (1996a) is somewhat similar to the method presented. For example, they used best fitting ellipse, intensity profiles and a set of rules to determine the facial feature locations. The biggest differences are that they also used watershed algorithm and the rule set for locating the facial features was different from the one presented here.

This chapter is organized so that first the technical aspects of the face detection method are described. Then the experiments carried out will be explained and the results of the experiments presented. In the discussion section the results are related to the wider context and to the other studies. At the end of the chapter there is a brief summary.

3.2 TECHNICAL BACKGROUND

The method performs face detection in several phases that are shown in Figure 3.1. Before the actual detection starts there is an initialization phase. Face detection takes place after initialization. The detection starts with blob detection when an image is received from a video camera. Feature selection and face probability calculation is the last phase of the detection. When an image has been processed the processing of the next image starts with the blob detection and this loop continues until face detection is stopped. Next each detection phase is described in detail.

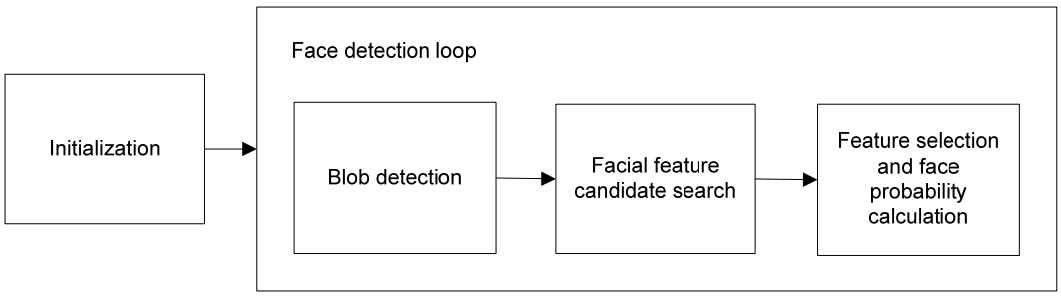


Figure 3.1. Face detection phases.

3.2.1 Initialization

During the initialization phase images of the scene where detecting takes place are captured with a video camera and a background model of the scene is created. The background model is an average image of the captured images. During the blob detection phase only those image regions that are sufficiently different from the background model are analyzed.

Successful initialization requires that there are only static objects in the scene. It is particularly important that there are no faces or other skin colored regions in the scene during the initialization. Otherwise face detection for those regions will very likely fail during the blob detection phase.

Initialization is not absolutely necessary, but it increases the speed and robustness of the detection loop. The speed up is possible, because when a person enters the scene and an image is captured with a video camera background regions can be determined rapidly and only non-background regions need further analysis.

The initialization requires a few seconds. It is also possible to update the background model during detection using non-face regions for the updating. This way it is possible to adjust to changes in lighting, for example.

3.2.2 Blob Detection

Promising regions that could be a face are searched during this phase. The captured image is scanned through pixel-by-pixel. Each pixel is compared to the corresponding background model pixel.

Those pixels that differ sufficiently from the background are compared to a skin color model. Hue and saturation are calculated from the RGB components and it is confirmed that the hue and the saturation are within the skin color ranges defined.

Skin colored pixels are combined into blobs (see Figure Figure 3.2a) using the connected component labeling algorithm described in Subsection 2.4.1. After skin colored blobs have been found such blobs are removed that are smaller than the defined minimum size.

Then the first-order and second-order central moments are calculated for each remaining blob. The best fitting ellipse is then calculated for each blob based on the moments. Finally, the blobs are rotated to the upright position. The major and minor axis and orientation of the ellipse determine how much and to which direction the blob is rotated to (see Figure 3.2b).

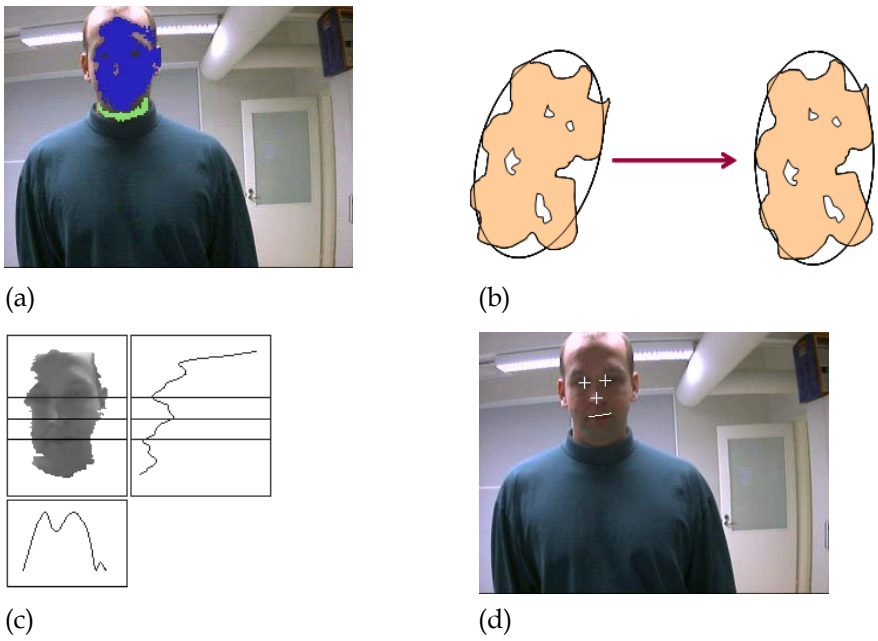


Figure 3.2. (a) Skin colored blobs are detected. (b) Blobs are rotated to the upright position. (c) A vertical intensity profile is created for the blob. Horizontal intensity profile (brighter intensities are shown lower) from the eye row also visualized. (d) The best feature candidate combination was chosen.

3.2.3 Facial Feature Candidate Search

After rotation facial feature candidates are searched for each blob. I used several rules in the search. The rules were experimentally selected so that they were enough strict to eliminate most of the non-faces while loose enough to find as many real frontal faces as possible.

The facial feature candidate search starts by creating a vertical intensity profile for the detected blob(s) (see Figure 3.2c). The average intensity value is calculated for each row of the blob. The profile is further smoothed by an average filter that has size of three rows. In other words, the intensity value of each row is the average of the row and the rows immediately above and below it.

After the vertical profile has been created eye candidates, nose tip candidates, and mouth candidates are searched and selected based on a set of rules. The following rules are used for eye candidate searching:

1. The eye row has to be above the middle point in the vertical direction.

2. The row has to be local intensity minimum in the vertical profile and the intensity has to be smaller than the average intensity of the whole profile. A row next to the row that fulfills the criteria is also further examined.

3. The horizontal intensity profile of the row (not of the whole blob) that has been filtered with 3x1 size average filter has to contain two local minima and there has to be a local maximum in between the minima.

4. The best local maximum of the row is such that it is located horizontally closest to the center.

5. The two best local minima of the row (one for each eye) are the minima that are at the eye width distance from the border of the blob and eye width distance from the best local maximum on the row.

Nose tip candidates are searched with the following rules:

1. The nose row has to be located in the range between ¼ and ¾ of the vertical profile.

2. The row has to be the local maximum in the vertical profile and the intensity of the row has to be higher than the average intensity of the vertical profile. A row next to the row that fulfills the criteria is also further examined.

3. The nose is assumed to be horizontally in the middle of the row and the middle point is used as a nose candidate location.

Mouth candidates are searched with the following rules:

1. The mouth row has to be located below the middle point of the vertical profile.

2. The row has to be the local intensity minimum in the vertical profile and the intensity has to be smaller than the average intensity of the whole profile. A row next to the row that fulfills the criteria is also further examined.

3. There have to be at least two local maxima and at least one local minimum in between the maxima in the horizontal profile of the row examined. The intensity of the minimum has to be below the average intensity of the row.

4. The two maxima that are closest to the center of the row and fulfill the previous rule define the horizontal start and end of the mouth candidate. However, if there is an intensity difference greater than 10 units (in the value range 0-255) between the minimum and either maximum then the start (or end) of the mouth is set at the location at which the difference was measured.

The above rules usually produce many candidates and the best candidates have to be selected for the final detection. The next phase performs the final selection.

3.2.4 Feature Selection and Face Probability Calculation

Although there are general rules for the locations of the eyes, nose, and mouth (such as the eyes are above the nose and the mouth in the upright face) there are also small variations between different people in these locations (see, for example, Farkas (1994)). With the face detection method, the best facial feature candidates are selected using probabilities that are determined from the relative locations between the candidates.

A set of probability rules based on facial feature positions and distances was created experimentally. Probabilities were defined to be between 0 and 1 (and corresponding probabilities between 0% and 100%). In some cases probability was 1 for a range of values while in some cases there was just one value that had a probability of 1. The probability was linearly reduced to 0 when the value moved away from the value that produced the maximum probability. The distances and probability rules that were used in the experiment are given in Table 3.1.

Table 3.1. Probability rules used for selecting the best facial feature candidate combination.

Rule	**Value giving maximum probability**	**Value giving zero probability, 1st direction**	**Value giving zero probability, 2nd direction**
Distance of the eyes (two local minima in the horizontal profile) from each other, d_e, related to the row width, w_r, where eyes are located on	$d_e = w_r * 2/5$	$d_e \geq w_r * 4/5$	$d_e = 0$
Distance of the eye horizontally from the border of the blob, d_b, related to the row width, w_r	$d_b = w_r / 5$	$d_b \geq w_r * 2/5$	$d_b = 0$
Distance of the eye to the local maximum between the eyes, d_m, related to the row width, w_r	$d_m = w_r / 5$	$d_m \geq w_r * 2/5$	$d_m = 0$
Vertical position of the eyes, p_e, relative to the blob height, h_b	$p_e / h_b = 0.45$	$p_e / h_b \geq 0.65$	$p_e / h_b \leq 0.25$
Vertical position of the nose, p_n, relative to the blob height, h_b	$p_n / h_b = 0.73$	$p_n / h_b = 1.0$	$p_n / h_b \leq 0.46$
Vertical position of the mouth, p_m, relative to the blob height, h_b	$p_m / h_b = 0.86$	$p_m / h_b = 1.0$	$p_m / h_b \leq 0.66$
Width of the mouth, w_m, (calculated from the two maxima on the horizontal profile) relative to the blob width, w_b (at the widest row)	$0.2 \leq w_m/w_b \leq 0.65$	$w_m = w_b$	$w_m = 0$
Vertical position of the eyes, p_e, and nose, p_n, related to each other and to the blob height, h_b	$(p_n - p_e) / h_b = 0.27$	$(p_n - p_e) / h_b \geq 0.54$	$(p_n - p_e) / h_b = 0$
Vertical position of the nose, p_n, and mouth, p_m, related to each other and to the blob height, h_b	$(p_m - p_n) / h_b = 0.27$	$(p_m - p_n) / h_b \geq 0.54$	$(p_m - p_n) / h_b = 0$
Vertical position of the eyes, p_e, nose, p_n, and mouth, p_m, related to each other	$(p_n - p_e) / (p_m - p_n) = 1.0$	$(p_n - p_e) / (p_m - p_n) \geq 2.0$	$(p_n - p_e) / (p_m - p_n) = 0$
Horizontal position of the left eye, l_e, right eye, r_e, and center of the eyes, c_e, and center of the mouth, c_m, related to each other	$c_e = c_m$	$c_m \geq r_e$	$c_m \leq l_e$

The distances and probabilities were calculated for all facial candidate combinations and the rules were given weights. In the experiment, the rule that related eyes, nose, and mouth vertically to each other was given twice as much weight as the other rules. The combination with the best weighted probability was selected (see Figure 3.2d). The probability also

stated the certainty that a face was detected. The face probability was set at zero if eye, nose, or mouth candidates were not found but otherwise it was something from 0% to 100%.

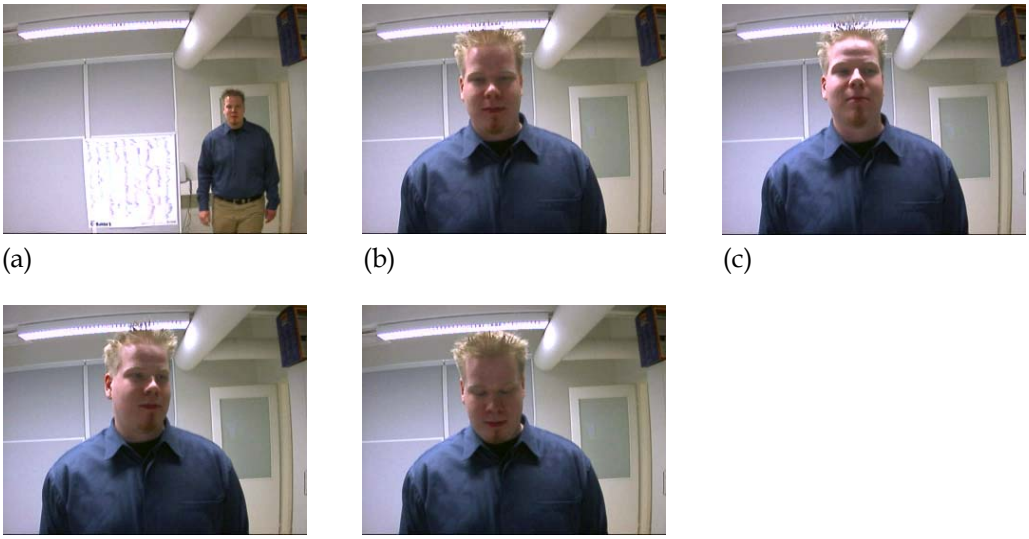
3.3 EXPERIMENT

To find out how reliable and fast the described rule-based face detection method is I carried out an experiment. The experimental procedure is described next. After that the results are reported and discussed.

3.3.1 Experimental Setup and Data

The data was collected with a web camera in an office environment. Eight people participated in the experiment. Five of the participants were female and three were male. One participant wore eyeglasses. All had white skin pigment.

The procedure was such that each participant walked in front of the camera and looked straight at the display below the camera. Then they turned their heads to the left, upwards, and downwards (see Figure 3.3). The photos taken with the camera were automatically analyzed by the face detection method and a log was created and stored along with the photos captured during the experiment.



(a) (b) (c)

(d) (e)

Figure 3.3. Photos taken by the web camera during (a) phase 1, (b) phase 2, (c) phase 3, (d) phase 4, and (e) phase 5.

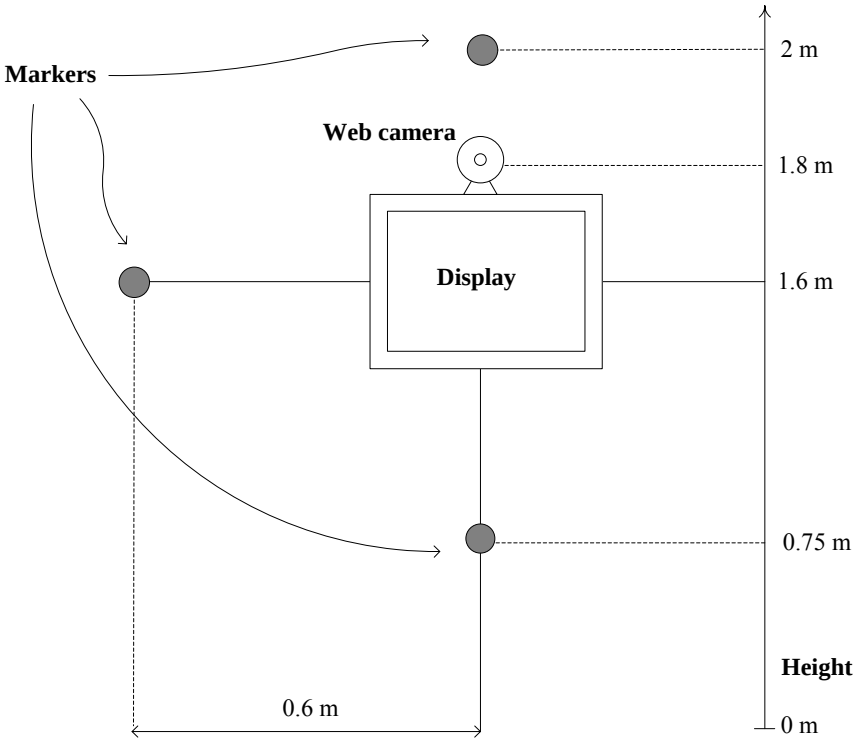


Figure 3.4. Experimental setup.

The experiment conditions were controlled so that 8 seconds was reserved for walking from behind the corner to the front of the display and the web camera (phase 1), three seconds for looking straight at the display (phase 2) and three seconds for each head turn: when the head was turned upwards (phase 3), to the left (phase 4), or downwards (phase 5). An example photo from each phase is shown in Figure 3.3. The place to stand was marked on the floor about 1 meter away from the display and the directions where to turn the head were also marked with red markers on the wall and on the table (see Figure 3.4).

Altogether 2,631 images were analyzed automatically during the experiment. The image size was 320*240 pixels. The experiments were carried out on an AMD Athlon 1.14 GHz CPU and with 256 MB of memory.

3.3.2 Detection Reliability of the Face

Face detection rate was measured for each phase. In the experiment, the face detection was accepted if the face probability calculated by the detector was over 0%. Average face detection rates and their standard deviations for each phase are shown in Figure 3.5. Average detection rates for each participant in phase 2 (looking straight at the display) are shown in Figure 3.6.

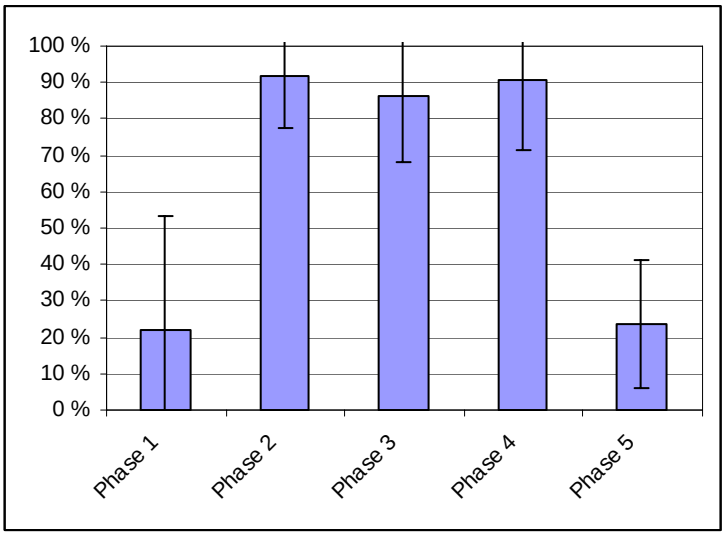


Figure 3.5. Face detection rates for each phase.

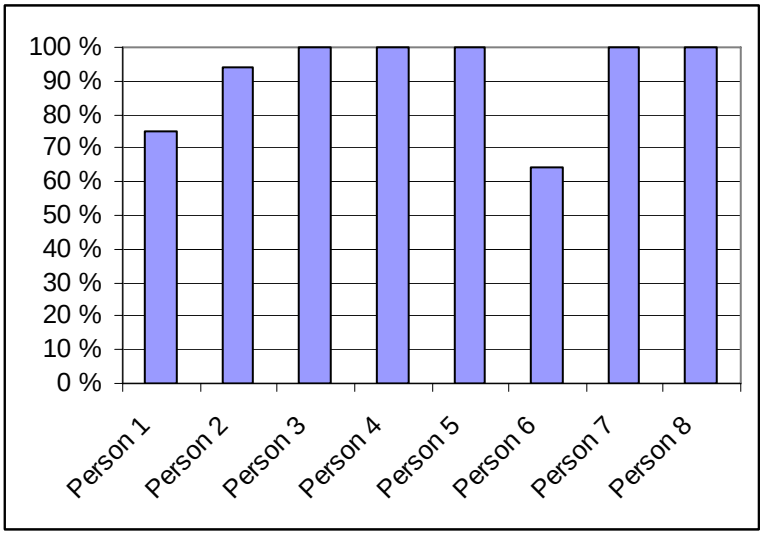


Figure 3.6. Face detection rates for the participants in phase 2 (looking straight at the display).

As can be seen from Figure 3.5, there are great differences in the rates between the phases and inside the phases between the participants. In phase 2 (looking straight at the display) and 4 (head turned to the left) the detection rate was slightly over 90% but in phase 1 (walking) and phase 5 (head turned downwards) it was slightly over 20%. One obvious reason for the low detection rates in phases 1 and 5 is that it was necessary to find candidates for all facial features to have face probability above 0%. However, it was hard to find the facial features in phase 1 because the person was a long way from the camera. It was also required that eye candidates were above the middle point of the blob in a vertical direction. The eyes could be located below the middle point in the phase 5 and because of this facial feature combinations were not necessarily found.

As can be seen in Figure 3.6, there are also differences in the detection rates between the participants in the phase 2 (looking straight at the display). Nevertheless, the face was detected in over 50% of the images for all subjects and for five subjects the face was detected in every image.

The face detection rate as such does not tell much about the goodness of the detection because the detection was always counted when the face probability calculated by the method was over 0%. The average face probabilities and their standard deviations for each phase are shown in Figure 3.7. As can be seen, the probabilities were rather low in all the phases. This could be due to two reasons: either the rules were very strict or the facial features were not successfully located. In fact, it was possible that facial features were not correctly detected even if the face probability was high. The successfulness of the facial feature detection is examined next.

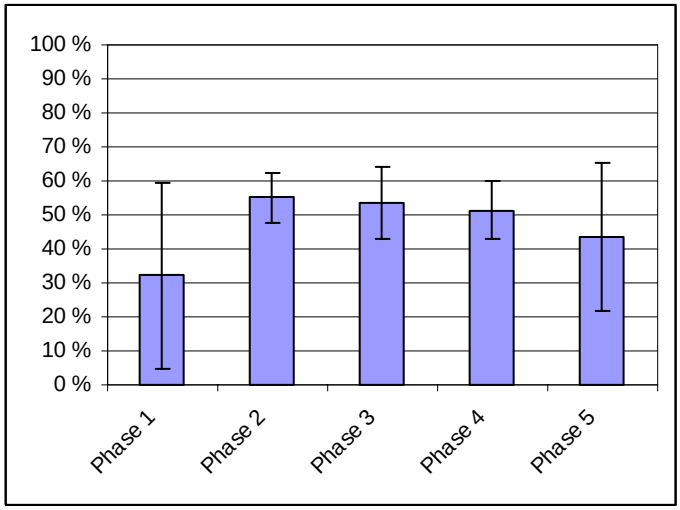


Figure 3.7. Average face probabilities in each phase.

3.3.3 Detection Reliability of the Facial Features

The reliability of the automatic facial feature detection was determined manually. The percentages for each person in phase 2 (looking straight at the display) are shown in Figure 3.8. The eyes were usually found without problems. The eye candidates were correctly chosen in 83.15% of detections. Noses were also fairly successfully found. The nose selection was successful in 71.56% of the detections. However, mouth detection was problematic. Only 47.19% of the mouth detections were correct.

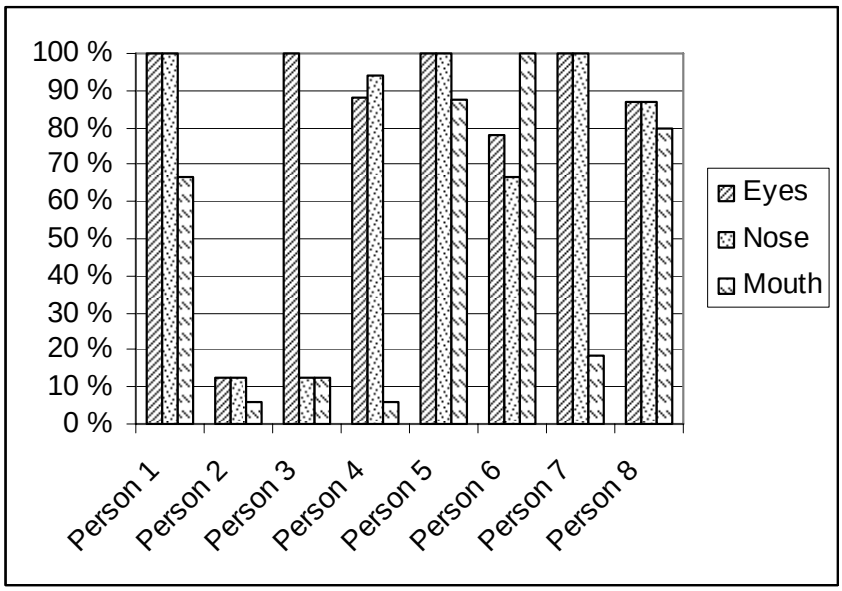


Figure 3.8. Percentages of the successfully detected facial features in the phase 2 (looking straight at the display).

For person 2 the skin color model could not separate hair and skin which explains the low success in this case. There was the same problem with the person 3. Even so, the eyes were always correctly found for her. Hair does not explain the low mouth detection rate for the person 4. In this case the neck skin was included in the face blob. As a result, the mouth was almost always located below the actual mouth location.

Clearly the low mouth detection rate had an effect on the face probabilities calculated by the method. However, it seems that the low face probabilities were also partly explained by the strict probability rules.

3.3.4 Detection Speed

The detection speed of the method described here depends heavily on the amount and size of the skin colored regions in the image. The number of the faces in the image and the people's closeness to the camera determine the number and size of the skin colored regions. Furthermore, the image size used affects the size of the regions. The number of facial feature candidates found also has an effect on the speed but the blob detection speed is the dominant factor.

The detection speed was measured with a profiler in the experiment. As described in Subsection 3.3.1 one person at the time walked in the front of the camera and then stood next to it. The image size was 320*240 pixels and the system had an AMD Athlon 1.14 GHz CPU and 256 MB of memory. The overall detection frequency, calculated from the time it took to analyze all the images of all the participants, was 23 Hz (images/second). However, when a person was walking towards the camera the skin colored regions were very small and the face detection

took very little time. When considering only the images where the participants were standing next to the web camera the detection frequency was between 18 and 23 Hz.

3.4 DISCUSSION

The detection reliability for the frontal faces, 91.68% correctly detected faces, is comparable to the other face detectors. For example, Viola and Jones (2001) achieved a detection rate between 93.7% and 78.3%. The rate depended on the number of false positives. One way to control the detection rate and the number of false positives with the method presented here is by using face probability. There are fewer false detections but also fewer correct face detections when the face probability required for an accepted detection is increased. This also holds for the other face detection methods.

The detection speed of the method, around 20 images per second for 320*240 pixel size image on an AMD Athlon 1.14 GHz CPU and 256 MB of memory, is enough to be used in perceptual user interfaces requiring real-time performance from the detector (Turk and Kölsch, 2004). As a comparison the cascaded face detector by Viola and Jones (2001) processed 15 images per second for 384*288 size images on a Pentium III 700 MHz CPU.

The greatest limitation for the method presented is that it uses skin color for detection. Color images are required by the method. However, since video cameras including web cameras capture color images there is a vast amount of HCI applications where the method can be used.

A more difficult problem is how to create a good skin color model that can be used in varying conditions. The model used worked well in the experiments because the lighting was stable and all the participants had white skin pigment. However, illumination and the ethnicity of the person affect the color of the skin and this affects the reliability of the method in unconstrained environments. Although there have been efforts to create models and methods that can be used in various lighting conditions, with cluttered backgrounds, and with different skin types and makeup the method that would work well in all the cases is still to be found (Kakumanu et al., 2007).

The presented facial feature detection method can also be used separately from the skin color detection. For example, rough face location could be determined by another face detector and the face alignment could then be done by using the facial feature locations detected by the proposed method.

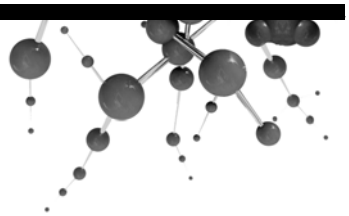
One possibility would be to combine different face detectors. When the detectors use different image information or learn the face model differently it is possible to improve face detection accuracy.

Face anthropometry is a research area that provides information on face dimensions. For example, Farkas (1994) presented precise face measurement data for North American Caucasians, for Chinese, and for African-Americans. This information is useful when developing rules for face detection. However, when doing face detection the conditions are usually challenging. For example, with the face detection method proposed hair could be included for the skin colored area, which could affect automatic face distance measurements. The probability rules used with the method were determined experimentally. This way the phases preceding facial feature selection were taken into account. Nevertheless, as the experiment shows, there is room for improvement especially when determining mouth location. After the experiment some modifications have been made in the rules. These modifications are described in Chapter 5 because the modified rules were in use when the facial feature detection was applied to face alignment.

3.5 SUMMARY

A rule-based face detection method was presented in this chapter. The experiments showed that the method can detect frontal faces with over 90% detection rate in real-time. The skin color model is the most problematic issue with the method because it imposes restrictions on the lighting conditions and requires color images. However, the probability rules used for the facial feature detection are the novelty of the method and can also be used without skin colored blob detection. In fact, using face alignment after face detection and before gender classification is one topic in Chapter 5. The facial feature detection method proposed was one of the two methods used for alignment.

The face detection method proposed was combined with a neural network gender classifier and the experiments with this combination are presented in Subsection 5.3.1. The cascaded face detector by Viola and Jones (2001) was also combined with a neural network gender classifier and the experiments with the combination are presented in Subsection 5.3.2. The same test images were used in both experiments. The results of the experiments can be compared and considered when creating new method combinations.



4 Gender Classification

4.1 Introduction

One could say that gender classification is vital for humans. However, there are also many applications for gender classification performed by computers. Many of these were mentioned in the preceding chapters and more will be presented in Chapter 7.

Earlier work on gender classification was described in Subsection 2.6.5. Although experimental studies on gender classification with different machine learning methods exist, none of the studies have been very thorough in comparing methods. Many times a novel method was compared to only one other method. Moreover, usually a non-public face database was used, making comparison of the methods of the studies virtually impossible. I compared a large set of gender classification methods to find differences between the methods. I made the comparison experimentally using six gender classification methods: multi-layer perceptron with pixel based input, SVM with pixel based input, SVM with LBP features, threshold Adaboost, mean Adaboost and LUT Adaboost.

The experiments were carried out in two parts. In the first part the classification accuracies of the methods were compared with face images with and without hair. The second part of the experiments (Mäkinen and Raisamo, accepted) was carried out to find out how sensitive the methods are to variations in face orientation, size, and offset from the image center.

4.2 TECHNICAL BACKGROUND

A general overview of the classification methods was given in Subsection 2.4.2. Here some aspects of the methods are described in more detail from the viewpoint of the experiments.

In the experiments the multi-layer perceptron took histogram equalized face images as input. The image pixel values were transformed from gray scale values (0-255) to the range from -0.5 to 0.5. Each node in the input layer thus received one transformed pixel value. In this case there was always one output node and the output values were between -0.5 and 0.5. The output scale was defined so that the closer the output was to -0.5 the more feminine the face was and the closer the output value was to 0.5 the more masculine.

The standard back-propagation algorithm was used to train the network. In this case a set of male and female face images was used in the training. The training took place so that each face example was inputted to the network one by one and this was repeated in rounds. The output of the network was compared to the expected output, -0.5 for the female and 0.5 for the male face. Then the difference between the expected and real output was calculated and the weights of the network connections updated. The weights were changed so that if the same face example was inputted again to the network the network output would be closer to the expected one. Some of the training images were separated and used as validation images.

For the SVM the optimal values for cost (C) and gamma (Γ) parameters were searched using grid search and cross-validation. When using grid search both parameter values were varied within a specified range of values. The value pair that produced the best average classification rate for the training examples was selected.

I used two kinds of data representations with SVMs: image pixel data and data obtained from the images filtered with local binary pattern (LBP) operators. In both cases the face images were histogram equalized and pixel data was scaled to range from -1 to 1 before being inputted to the SVM. With the second data representation the LBP-operator was used in between equalization and scaling and its use is described next.

I used the basic LBP operator with four neighbors at a radius of one ($LBP_{4,1}$) and the uniform LBP with eight neighbors at a radius of one ($LBP^{u2}_{8,1}$) somewhat similarly to the experiments by Hadid et al. (2004) and Lian and Lu (2006). I divided each face image into 8*8-blocks and filtered each block with the basic LBP operator with four neighbors at a radius of one ($LBP_{4,1}$). However, with face images of size 36*36 it is not possible to divide images evenly into 8*8 blocks. In this case I used smaller 4*8 blocks

at the right side of the images, 8*4 blocks at the bottom of the images, and one 4*4 block at the bottom right corner of the images. Then I created a histogram of each block. Since there are 16 distinct values that can be produced by the $LBP_{4,1}$-operator, each histogram had 16 bins and each bin contained the amount of each value in the filtered block. I also filtered the whole face image with the uniform LBP with eight neighbors at a radius of one ($LBP_{8,1}^{u2}$) and created a 59-bin histogram for it. Finally, all the histograms were concatenated. For example, with a 24*24 image there were nine 8*8-blocks. The total number of bins in the concatenated histogram vector was therefore 9*16+59=203. The vector was used as an input to the SVM.

4.3 EXPERIMENTS

In the first part of the experiments multi-layer perceptron with pixel based input, SVM with pixel based input, SVM with LBP features, threshold Adaboost with Haar-like features, mean Adaboost with Haar-like features, and LUT Adaboost with Haar-like features were used. In the second part of the experiments multi-layer perceptron with pixel based input, SVM with pixel based input, SVM with LBP features, and threshold Adaboost with Haar-like features were used. The threshold Adaboost variant was chosen instead of the mean Adaboost and LUT Adaboost because it is probably the most well known of the three Adaboost variants. The decision to omit the other Adaboost variants from the second part is also justified because all the Adaboost variants produced very similar results in the first part of the experiments.

4.3.1 Data

Images from the FERET database (Phillips et al., 1998) were used in the experiments. The database contains face images of people of varying ages, with eyeglasses and without, with facial hair and without and of different ethnic backgrounds. The fa- and fb-subsets containing frontal faces of 1,196 people were used in the experiments. Duplicate images of the same person were removed, so that only one image per person was left. The eye locations were manually determined and genders annotated.

For the first experiment 900 face images were randomly selected (450 female and 450 male images). Then they were aligned using the eye locations, the faces were cropped and after cropping the face images were resized to the size of 24*24 pixels or to the size of 32*40 pixels. The 24*24 size images were cropped so that only little or no hair was left in the image. With the 32*40 image size the number of images was reduced to the 754 images from 900 (still equal number of both genders) because including hair in the face area could extend the face area beyond the original image borders and such images were removed.

In the second experiment, with 24*24, 36*36, and 48*48 face image sizes, 304 randomly selected FERET face images (equal number of both genders) were used as training images and 107 randomly selected face images (60 male faces and 47 female faces) were used as test images.

4.3.2 Procedure

The faces were aligned and the face areas used for the input were determined using the algorithm shown in Figure 4.1.

1. The image is rotated so that eyes are vertically aligned.
2. The Euclidean distance d_o between the eyes is calculated in the rotated image.
3. The ratio r is calculated by $r = d_o / d_t$, where d_t is the defined distance of the eyes in the resized image.
4. The width w_o and the height h_o of the of the area around the eyes are calculated by $w_o = r * w_t$ and $h_o = r * h_t$, where w_t and h_t are the width and height of the resized image (e.g. $w_t = h_t = 24$ or $w_t = 32$ and $h_t = 40$).
5. The coordinates for the corners of the face area in the rotated image are calculated by $x_l = x_e - w_o / 2$, $y_t = y_e - h_o / r_h$, $x_r = x_l + w_o$ and $y_b = y_t + h_o$, where x_l is the x-coordinate of the left border, x_e is the x-coordinate of the point halfway between the eyes, y_t is the y-coordinate of the upper edge, y_e is the y-coordinate of the eyes, x_r is the x-coordinate of the right border, y_b is the y-coordinate of the lower edge and the ratio of the height above and below eyes r_h was 3.5.

Figure 4.1. Face alignment and face area calculation algorithm.

After determining the face areas with the above algorithm faces were resized to specified sizes. To the 32*40 size images 10% was added to width on both sides (20% in total), 40% to the top and 12% to the bottom of the area before scaling to include hair in the face images.

In the first experiment, 80% of the FERET images were put in the training set and 20% in the test set images. Because with the neural network and SVM there is a risk of over-fitting to the training data, a part of the training images was used for validation when selecting optimal parameters for the methods. For the neural network 2% of the training images were separated in the validation set and several neural networks were trained using different numbers of hidden neurons and learning rates. Each neural network was then tested with the test images. For the SVM the best parameters were searched with the five-fold cross-validation, so that 20% of the training images were in the validation set at a time. After the optimal parameters had been selected the final SVM classifiers were trained with the whole training set of the images. Since Adaboost is resistant to over-fitting (Freund and Schapire, 1997) we trained classifiers

using Adaboost directly with the whole training set. For the LUT Adaboost one can use a different number of bins and 4, 6, 8 and 12 bins were tried. 500 features were selected for each Adaboost classifier.

The best parameters found for the face images with and without hair are shown in Table 4.1. The results presented for the first experiment were achieved with the parameters presented.

Table 4.1. Best parameters for the methods with face images with and without hair.

	Face images without hair (24*24)	**Face images with hair (32*40)**
Neural Network		
Number of hidden nodes	2	2
Input-hidden layer learning rate	0.0007211	0.0279399
Hidden-output layer learning rate	0.01	0.57735
SVM (RBF Kernel)		
C	32.0	2.0
Γ	0.0001221	0.0078125
LBP + SVM (RBF Kernel)		
C	2.0	2.0
Γ	0.03125	0.03125
LUT Adaboost		
Number of bins	4	8

In the second experiment, the training took place as in the first experiment but with a smaller number of images. For each method the classifier that performed best for the properly aligned faces was used. The best parameters are shown in Table 4.2. For the threshold Adaboost 500 features were used as in the first part of the experiments.

Table 4.2. Best parameters for the methods in the second experiment.

	24*24 size images	**36*36 size images**	**48*48 size images**
Neural Network			
Number of hidden nodes	2	2	2
Input-hidden layer learning rate	0.0416305	0.0277671	0.0208288
Hidden-output layer learning rate	0.577350	0.577350	0.577350
SVM (RBF Kernel)			
C	128	32	512
Γ	0.0004882812500	0.0001220703125	0.000030517578125
LBP + SVM (RBF Kernel)			
C	2	8	32
Γ	0.03125	0.0078125	0.0001220703125

The best classifiers were tested by rotating the face images from -45 degrees to +45 degrees, scaling the face images with factors from 0.2 to 5, and by translating the bounding box vertically and horizontally from -3 to +3 pixels. Note that the translation was done after the face images were resized because otherwise the translation would have been different for different faces. This follows from the fact that the faces had various sizes in the original images (photos taken from various distances). Example results of performing each type of transformation are shown in Figure 4.2.

(a)  (b)  (c)  (d) 

Figure 4.2. Examples of the face transformations for the sensitivity tests. (a) Original (resized) face image. Face after (b) rotation, (c) scaling, and (d) translation.

It is possible that a part of the face image would have gone partially beyond the image bounds after transformation, so this was checked and image pixels that were outside the image were given an intensity value of 127 (intensity range between 0 and 255). A good example of the situation where some pixels are beyond the image bounds and have been given an intensity value 127 after face has been scaled is given in Figure 4.2.

4.4 RESULTS

The classification accuracies for the first experiments are shown in Table 4.3. The classification accuracy is the percentage of the faces that are correctly labeled as either male or female. The neural network and LUT Adaboost produced the best classification rates on average but there were no great differences between the methods. Furthermore, the classification rate is slightly better for the "with hair" images in average although there are two exceptions at the method level and there are no statistically significant differences between the "with hair" and "without hair" conditions (Wilcoxon signed-rank test: $z_6 = 0.734$, $p = 0.463$).

ROC curves were briefly introduced in Subsection 2.6.1 when considering face detection. However, it is possible to draw a ROC curve for a gender classifier, too. The ROC curves are shown for "without hair" images in Figure 4.3 and "with hair" images in Figure 4.4. The curve can be drawn for a method by changing the threshold value that determines the classification. For example, with the neural network we used the possible output values between -0.5 and 0.5. Changing the threshold little by little from -0.5 to 0.5 we therefore affect the fraction of faces classified as males and females. The closer the threshold is to the -0.5 the more female faces will be classified as female but at the same time more the male faces will be classified as female. The fraction of males classified correctly is presented on the y-axis and the fraction of the females classified incorrectly is presented on the x-axis. For example, a curve point at the coordinates (x, y) = (0.21, 0.84) means that 21% of the females are classified incorrectly when 84% of the males are classified correctly. The greater the area under the ROC curve is the better the method is at classifying genders. The perfect curve would be such that it goes from the lower left corner to the upper left corner and from there to the upper right corner.

Table 4.3. Classification accuracies for the classifiers with the face images with and without hair in the first experiment.

	Classification accuracy %		
Method	**without hair (24*24)**	**with hair (32*40)**	**Average**
Neural Network	92.22%	90.00%	91.11%
SVM	88.89%	82.00%	85.45%
Threshold Adaboost	86.67%	90.00%	88.34%
LUT Adaboost	88.89%	93.33%	91.11%
Mean Adaboost	88.33%	90.00%	89.17%
LBP + SVM	80.56%	92.00%	86.28%
Average classification accuracy	87.59%	89.56%	88.57%

The ROC curves show that Adaboost variants especially gave a very similar performance. However, SVM with pixel based input and multi-layer perceptron with pixel based input also gave a fairly similar performance for both with hair and without hair images. The SVM with LBP features performed clearly worse than the other methods when without hair images were used but was slightly better than the other methods when with hair images were used. The findings indicate that gender classification performance may depend more on the features used than on the actual classifier. In addition, since LUT Adaboost did produce rates similar to the other Adaboost classifiers, it seems that gender classification of the frontal manually aligned faces is a linearly separable problem, at least when Haar-like features are used, and the threshold Adaboost and the mean Adaboost classifiers can be used for the task.

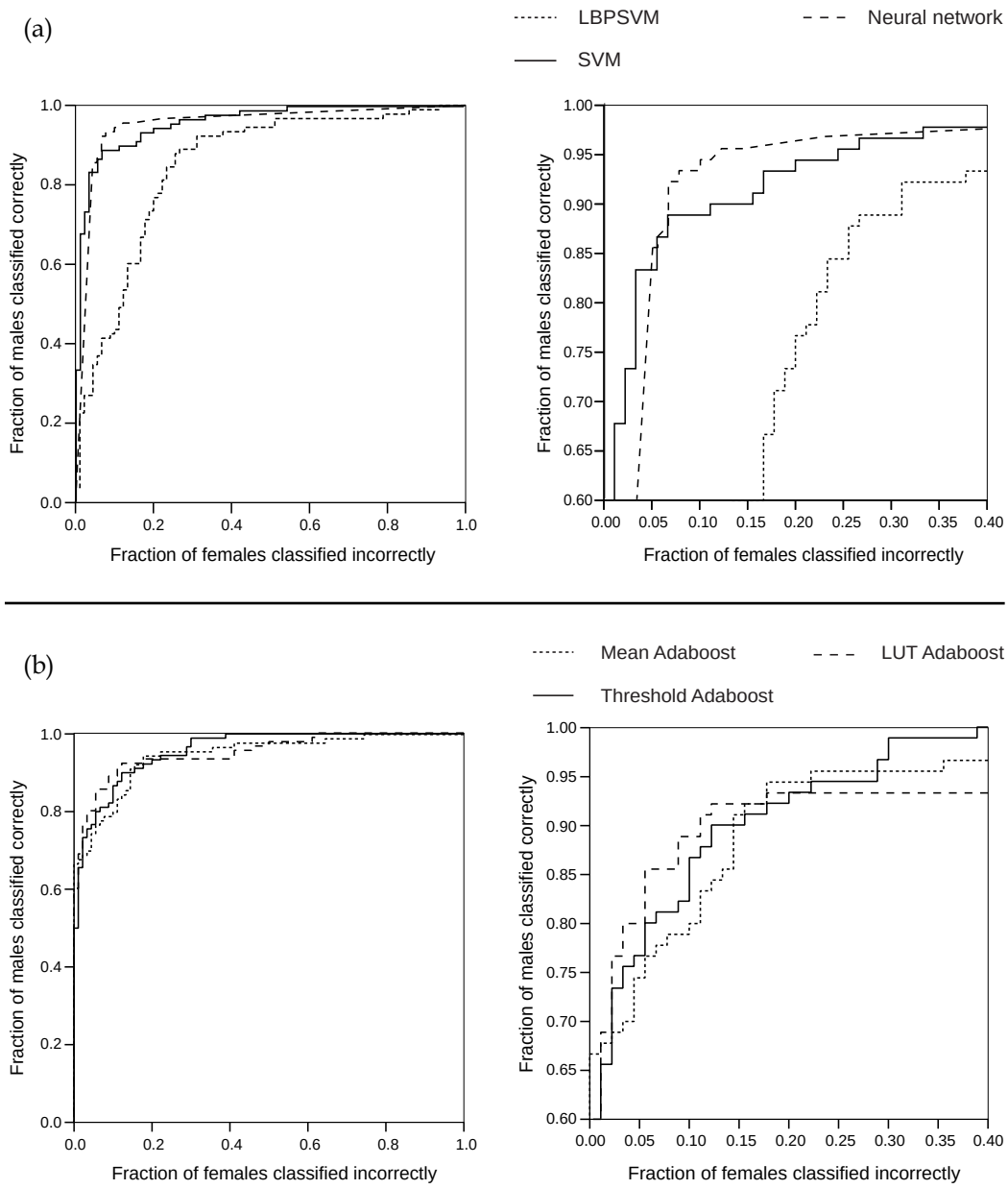


Figure 4.3. ROC curves for images without hair (24*24 size images). (a) ROC curves for the SVM with pixel based input, for the SVM with LBP features, and for the multi-layer perceptron. The top left part of the curve is zoomed on the right. (b) ROC curves for the mean Adaboost, for the threshold Adaboost, and for the LUT Adaboost. The top left part of the curve is zoomed on the right.

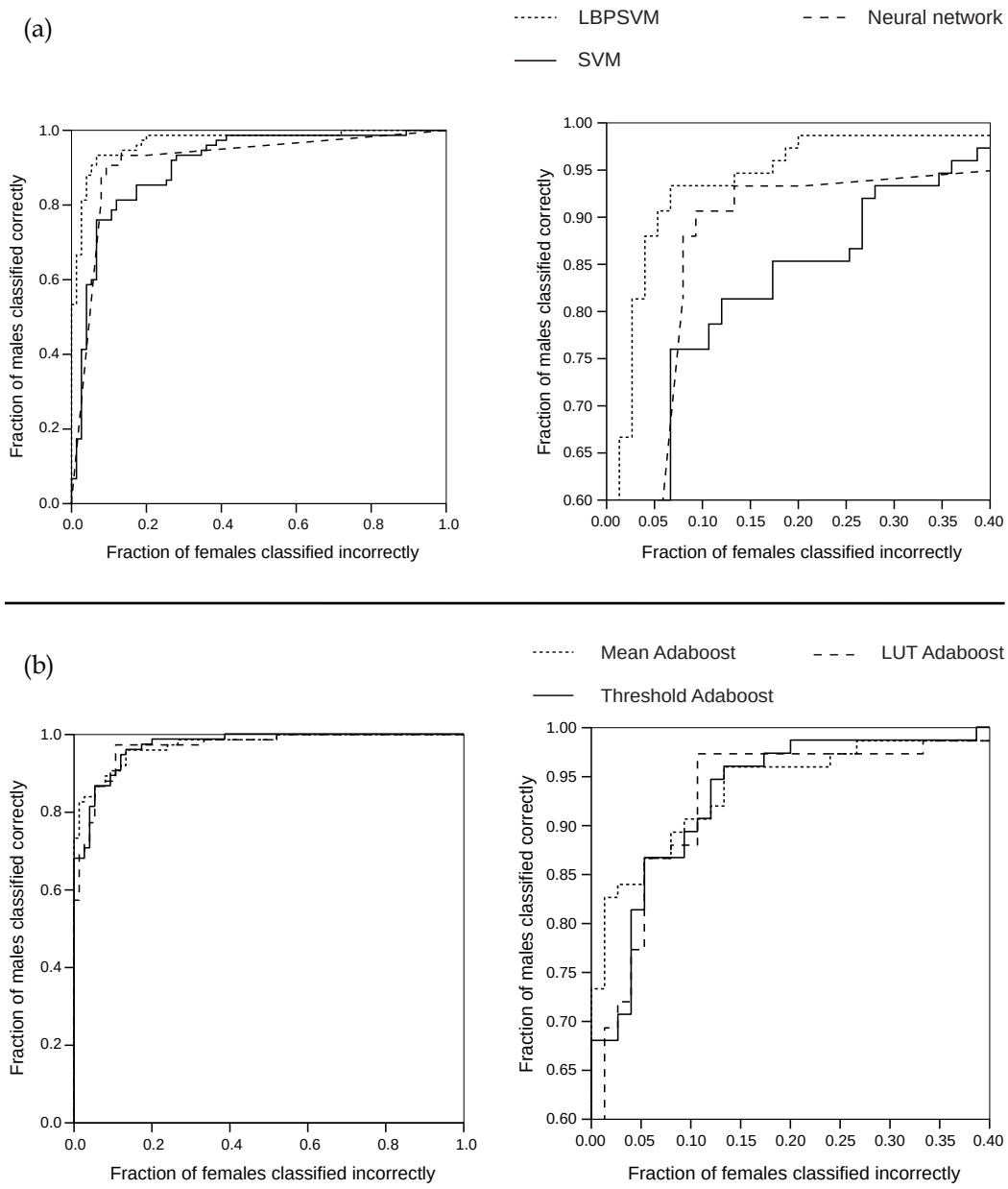


Figure 4.4. ROC curves for images with hair (32*40 size images). (a) ROC curves for the SVM with pixel based input, for the SVM with LBP features, and for the multi-layer perceptron. The top left part of the curve is zoomed on the right. (b) ROC curves for the mean Adaboost, for the threshold Adaboost, and for the LUT Adaboost. The top left part of the curve is zoomed on the right.

Now the results of the second part of the experiments are presented. While the first part of the experiments concentrated on the effects of excluding and including hair in the face images the second part of the experiments explored the effect of variations in face image quality that may be present, for example, when automatic face detection precedes gender classification.

The effects of rotation and scale averaged over all three face image sizes (24*24, 36*36, and 48*48) for the four classifiers are shown in Figure 4.5 and in Figure 4.6.

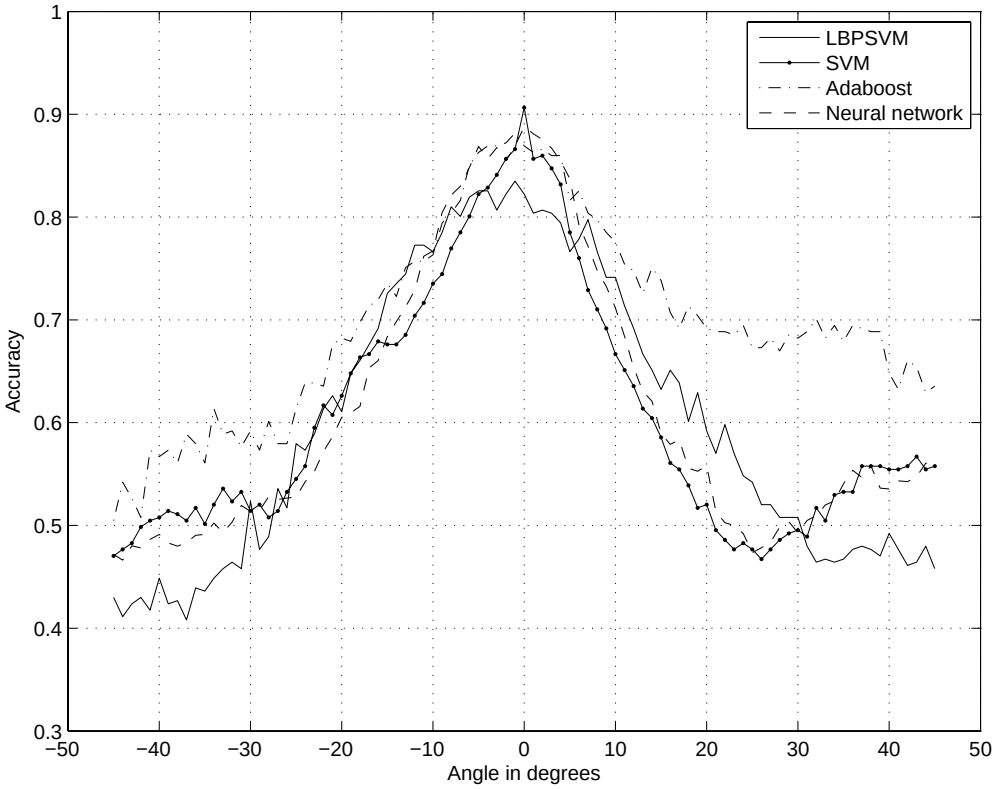


Figure 4.5. Effect of rotation on the gender classification rates when rates have been averaged over all image sizes.

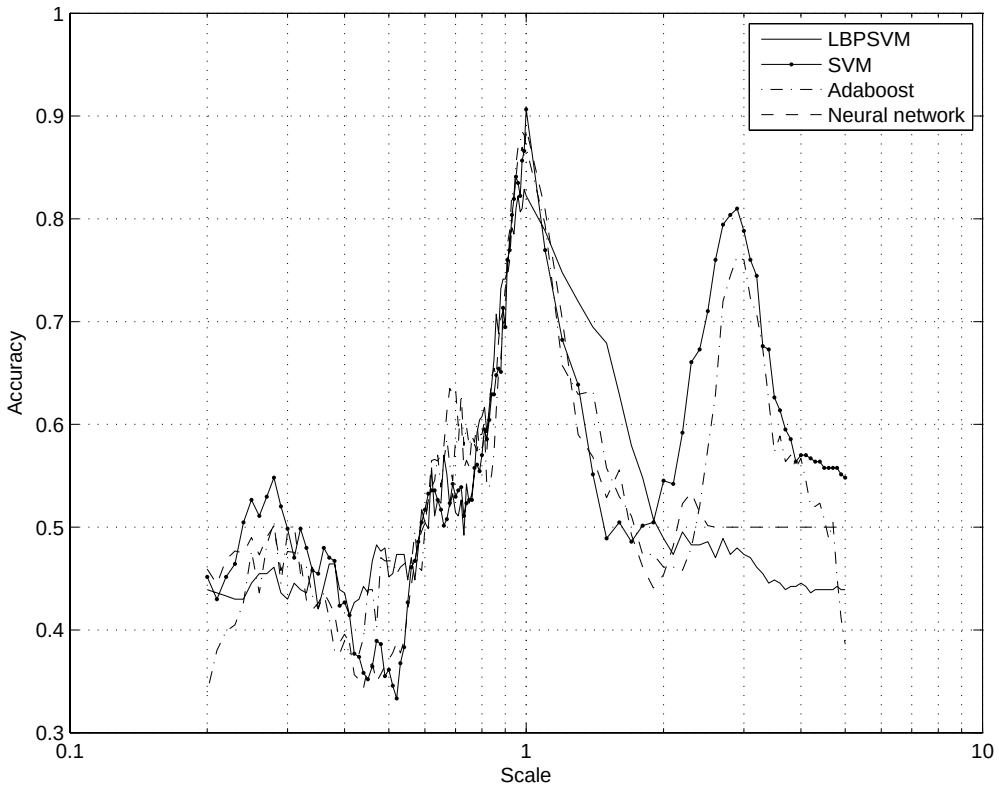


Figure 4.6. Effect of scale on the gender classification rates when rates have been averaged over all image sizes.

While considering the effect of the rotation it seems that Adaboost with Haar-like features is the most resistant method for the rotation variations. Although the best classification rates were achieved with SVM when image pixels were used as input, the classification accuracy fell fast when the face orientation was changed.

The results of the scaling are also interesting. The most striking and also the most surprising result is the high classification accuracy for Adaboost and SVM with pixel based input with the scaling factor close to 3. What is also interesting is the below chance classification accuracies for all the methods when a scaling factor between 0.3 and 0.6 or close to 0.2 was used. There is no obvious reason for the peaks and pitfalls in the performances.

The effect of the rotation and scaling averaged over the four methods is shown in Figure 4.7 and in Figure 4.8. As can be seen in the two Figures there were no large differences between different image sizes when orientation and scale were varied.

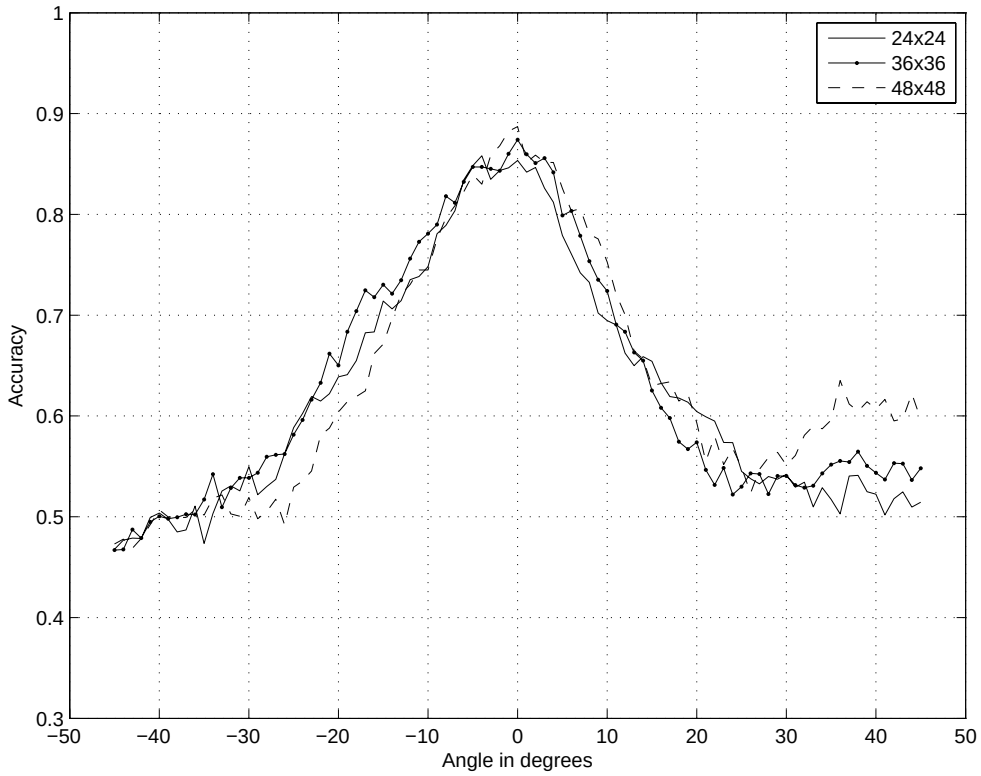


Figure 4.7. Effect of rotation on the gender classification rates when rates have been averaged over all classification methods.

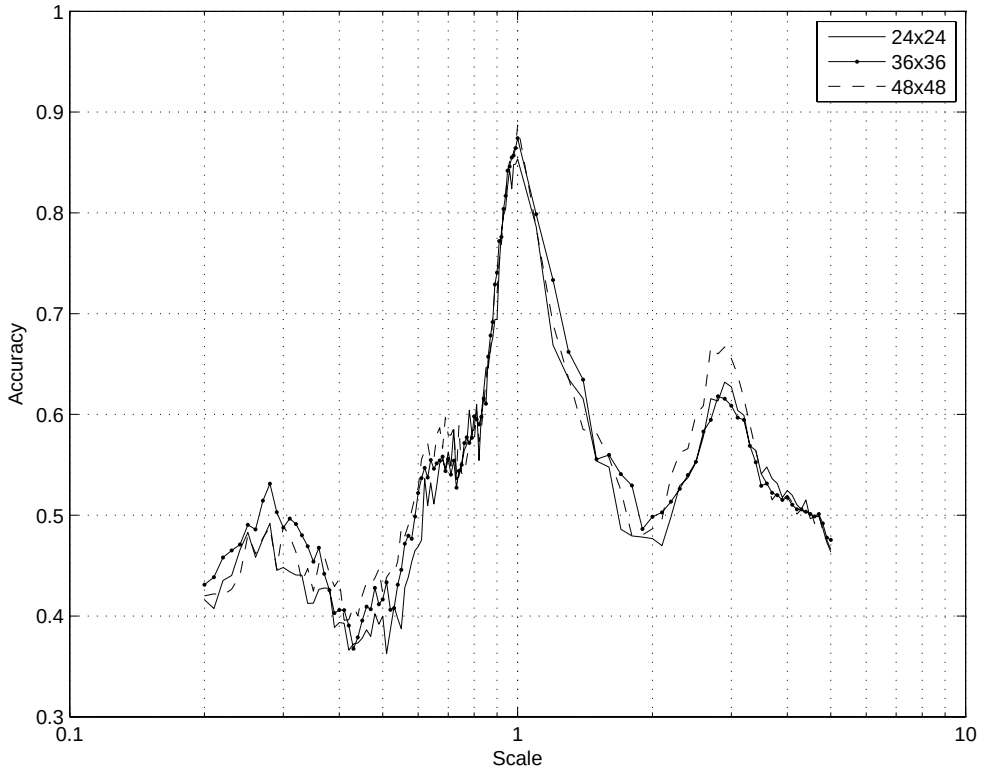


Figure 4.8. Effect of scale on gender classification rates when rates have been averaged over all classification methods.

The effect of the translation on the classification accuracy with different image sizes is shown in Figure 4.9. The accuracies are averages calculated from the accuracies of all the methods.

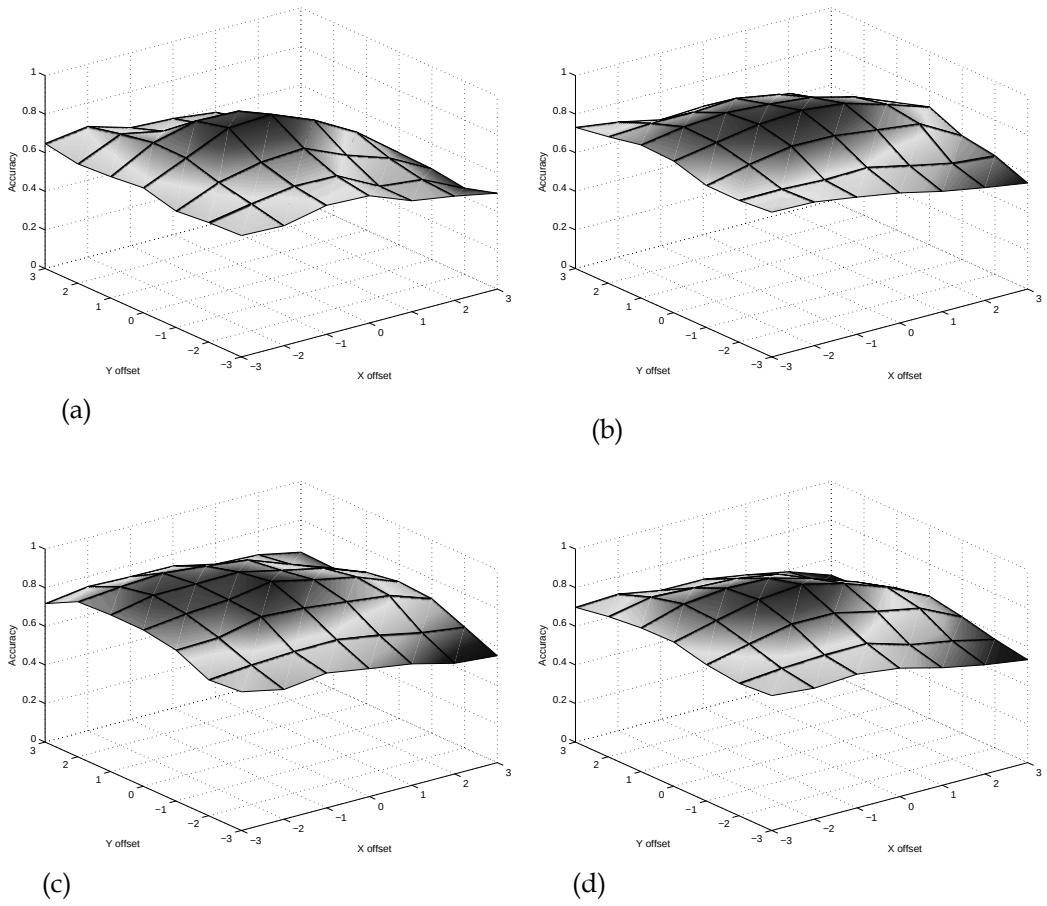


Figure 4.9. Effect of translation on classification accuracy with different image sizes. (a) 24*24 size images. (b) 36*36 size images. (c) 48*48 size images. (d) Average over all image sizes (and over all classifiers).

The accuracy decreased slightly faster for the 24*24 size face images than for the other two face image sizes, as could be expected because the relative translation was greater for the small image size.

The effects of rotation, scale, and translation on the specific methods with specific image sizes are shown in Figures of Appendix 2. The issue that one notices on scrutinizing those figures is that the classification performance curves do a little zigzag although the overall trend is almost always what one could expect. Actually the zigzag is what one could expect to see, because, for example, one percentage unit change in rotation has only little effect on the face and with such a small change a classifier could classify more faces correctly by chance.

4.5 DISCUSSION

Maybe the most interesting finding was that features may affect gender classification performance more than the machine learning method. This was indicated by the fact that the classification rates between different methods were the smallest when they used the same features. In many earlier studies (Shakhnarovich et al., 2002; Sun et al., 2002b; Wu et al., 2003a) the differences between some of the methods tested have also been relatively small, which further supports the importance of selecting proper features for the classification.

The threshold Adaboost with Haar-like features was clearly more resistant to the rotation variation than the other classifiers. The reason may be the Adaboost classifier itself, the Haar-like features or they may both contribute. The study by Baluja and Rowley (2007) suggests that at least the Adaboost classifier is important. They compared performance of the SVM classifier and Adaboost classifier, and the Adaboost classifier proved more resistant to the rotation variation. The features they used with the Adaboost were Boolean comparison values between image pixels. The SVM had pixel based input.

Moghaddam and Yang (2000) showed that face image size is not an important factor for SVM classification performance. The results of these experiments support their finding. In fact, it seems that size is not an important factor with any classifier. The effect of the translation offset was indeed larger with the smaller face image size but (usually) the alignment errors occur before the face image is resized.

Including hair in the face images improved classification performance, but not much. Abdi et al. (1995) reported that, for example, face shape affects the classification in addition to hair. In light of these facts it seems that some improvement in performance can be achieved when face outline and hair are included in the face image.

There were also some indications that gender classification with well aligned frontal faces may be a linearly separable classification problem. The Adaboost variant did not affect classification performance. In addition, when an optimal amount of hidden nodes for the neural networks were searched, the neural networks with one hidden node produced results nearly equal to the networks with two hidden nodes.

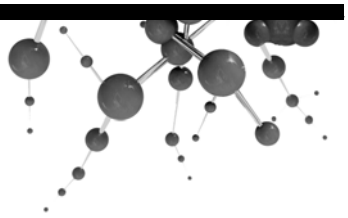
An interesting issue is also the effect of the face image scaling on the classifier performance. The peaks in performance for the SVM with pixel based input and Adaboost with scaling factor at around 3 can hardly be explained by chance alone. There were also scaling factors that produced less than 50% accuracy for all classifiers, which is not good for a two-class classification problem. Furthermore, Baluja and Rowley (2007) reported similar, although not as strong, peaks and pitfalls in classifier

performances when the scaling factor was varied. The study of the issue requires further work.

4.6 SUMMARY

In this chapter gender classification was studied experimentally from different perspectives. Gender classification was studied with manually aligned face images and with various gender classifiers. The gender classification accuracy was slightly improved when hair was included in the face images. In addition, features used with the classifier seemed to be a more important factor for classification accuracy than the type of classifier. Furthermore, the results indicate that gender classification could be a linearly separable problem when Haar-like features are used with the classifier. The sensitivity of the classification to image rotation, scale and translation was studied. The most interesting finding was that Adaboost was more resistant than the other methods when a face was rotated. The other interesting issue was that the classification accuracy was in some cases unexpectedly high or low when the scale of the face was changed.

The results reported here help to put the experiments described in the next chapter in a wider context. The topic of the next chapter is how to use automatic face detection with gender classification. Such topics as classification reliability with low quality images and classification speed are considered. These issues are important, for example, from the viewpoint of perceptual user interfaces.



5 Combining Face Detection and Gender Classification

5.1 INTRODUCTION

Face detection can be used in some applications as such. For example, the kiosk application presented in Section 7.3 took advantage of face detection. However, many more applications become available when face detection is combined with other face analysis tasks such as face recognition, age classification, and gender classification.

In this chapter I studied how face detection should be combined with gender classification. Various experiments were carried out for this purpose. There are only few studies on the topic and they are listed in Table 5.1.

This chapter is organized so that first there is some technical background on face detection, alignment and gender classification. Many technical aspects have already been handled in the previous two chapters, so they are not repeated here. After the technical background the experiments are presented and discussed. Finally, there are some concluding remarks and a summary.

Table 5.1. Existing studies combining face detection and gender classification.

Study	Face detection, normalization and gender classification method(s)	Face database(s)	Best generalization rate
Moghaddam and Yang (2002)	Maximum-likelihood estimation and eight different gender classification methods.	3,006 FERET (Phillips et al., 1998) images.	96.6
Shakhnarovich et al. (2002)	Detector by Viola and Jones (2001), no normalization. Threshold Adaboost with Haar-like features for gender classification.	3,500 images collected from the WWW.	79.0
Castrillón et al. (2003)	Detector and normalization by Castrillón (2003). Normalization with eye based alignment, scaling, and PCA. Gender classification with SVM and NNC.	48 non-public video sequences, and 1,000 images not in the sequences.	98.6
Wu et al. (2003b)	Cascade using LUT weak classifiers. SDAM (Wang et al., 2003; Xiao, 2002) based face alignment, and histogram equalization. LUT-Adaboost with Haar-like features for gender classification.	13,600 images combined from the FERET database and from the WWW.	88.0
BenAbdelkader and Griffin (2005)	FaceIT® library based face detection, eye detection, face alignment, resizing and cropping. SVM and FLD for gender classification.	12,964 images including several public databases.	94.2
Castrillón et al. (2006)	Detector by Castrillón et al. (2005). Eyes based alignment and resizing. PCA and SVM for gender classification.	6,000 images collected from the WWW and databases, and 900 non-public video sequences.	83.0
Yang et al. (2006b)	Nested cascade detector (Huang et al., 2004). Feature locating method by (Zhang et al., 2005), 3 face alignment methods, brightness normalization, and PCA. SVM, FLD and two-layer Real Adaboost for gender classification.	11,500 Chinese face images and 3,592 FERET images.	97.2

5.2 TECHNICAL BACKGROUND

Three issues are addressed in this section: how the output produced by the face detector is converted to the input used with a gender classifier, how face normalization and alignment are performed after a face has been detected, and how to use Haar-like features selected by Adaboost with a neural network.

5.2.1 From Face Detection Output to Gender Classifier Input

How the face detection output is converted to the classifier input (also other than gender classifier input) depends on the face detector used. The procedure for the two face detectors used in the experiments is described here. The face normalization and feature extraction phases (see Section 2.6 and Figure 2.14) are between face detection and classification phases and also affect conversion.

The cascaded face detector outputs the coordinates of the rectangle that determines the location of the face. Automatic face alignment was used in some of the experiments to adjust the location of the rectangle more accurately to the face. The pixel data inside the rectangle was used for creating the input.

The face detector presented in Chapter 3 outputs locations of the eyes, the nose, and the mouth. The rectangular area covering the face can be calculated based on these features. The following rules were used for the calculation of the rectangular area:

1. The width for the rectangle is calculated using the *eye distance*, d_e. The d_e is the distance between the detected locations of the left eye and the right eye.

2. Some space is left on both sides of the eyes. The width of the space is 0.25 * d_e for each side of the eyes.

3. The total width for the rectangle is 1.5 * d_e.

4. The space included above the eyes is 0.5 * d_e.

5. The height of the rectangle is 2.2 * d_e.

The rules are illustrated in Figure 5.1. After the rectangle has been calculated the pixels located inside it can be processed as with the cascaded face detector.

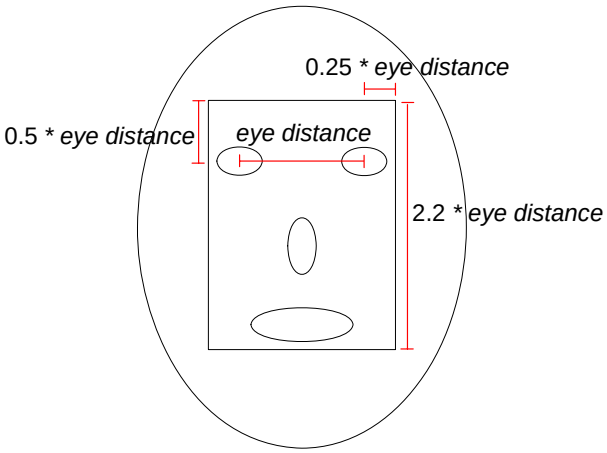


Figure 5.1. Rules used to determine the face rectangle.

Histogram equalization, PCA or some other processing can be done for the pixels before the data is inputted to the classifier, but this processing can be and often is independent of the face detection part. In these experiments the pixels inside the detected rectangle were processed as with the manually aligned faces (described in the previous chapter). The processing included resizing the rectangular area and the processing was partially dependent on the gender classification method. For example, histogram equalization was used with a neural network and with SVM, and Haar-like features were used with Adaboost for feature extraction.

5.2.2 Face Alignment

As described in Subsection 2.6.3 facial features can be used for face alignment, and one way to find the locations of the facial features is to use a shape model. Active Appearance Model (AAM) by Cootes and Taylor (2001) was chosen for the experiments described here. Besides AAM, a method called profile alignment was also used.

When AAM is used, the alignment starts by fitting the AAM model to a face (see Subsection 2.6.3 and Figure 2.21). Then the facial feature locations used for the alignment are determined. In the experiments three different alignments based on AAM shape were used. One used only eye centers, one used eye centers and nasal spine, and one used eye centers and mouth center. Since the AAM shape did not include eyes and mouth center directly they had to be calculated using the shape points. The points 14, 18, 22, and 26 were used for the eyes and the points 42 and 46 for the mouth (see Figure 2.20). The nasal spine was located at the point 53. After eye locations had been calculated, the face was rotated so that the eyes were horizontally at the same level. In addition, the face was scaled so that the distance between the eyes was the same in all faces. If nose or mouth location was used for the alignment then the face was stretched or squeezed so that the nose or mouth center was at the same vertical location related to eyes in all aligned faces.

The profile alignment is based on the face detector presented in Chapter 3. However, only the part that does the facial feature candidate search and selection was used for the alignment. The facial feature candidates were searched from the face rectangle determined by the cascaded face detector. After the facial features had been selected the alignment was done using the eye locations. The face was scaled so that the distance between located eyes was equal in all faces as with the AAM alignment. However, face rotation was not included because eyes were always selected from the same image row. Also, it was possible that eye candidates were not found with the algorithm. A benefit of the profile alignment was that it was very fast when compared to the AAM alignment.

5.2.3 Using Adaboost Selected Haar-like Features with Neural Network

Besides forming a strong classifier, Adaboost can also be used to select features that are used as an input to a classifier that is independent of the feature selection process. Littlewort et al. (2006) did this in facial expression recognition domain where their real-time system used Support Vector Machine (SVM) on the Gabor filters selected by Adaboost.

I decided to experiment with how the Haar-like features selected by Adaboost and used as input to a neural network work in gender classification domain. To best of my knowledge, this was the first time that Haar-like features were used as input to a neural network. Feature selection proceeded as usual with Adaboost. However, after features had been selected the multi-layer perceptron was trained with the features. The thresholds used with the features during the selection were no longer used. Also, the example faces had an equal weight during the training of the perceptron. The training proceeded in rounds where feature values were calculated for each example face, scaled to the range from -0.5 to 0.5, and used as input to the perceptron. Each feature had its own input node. The network weights were updated using the back-propagation algorithm.

Since each Haar-like feature had different minimum and maximum value that depended on feature type and size the feature value scaling was different for each feature. The equation that was used for the feature value scaling was:

$$f_{scaled}(x) = f(x)/((w_f * h_f * 255)/2)/2,$$

where *f(x)* was the original feature value and w_f and h_f were the width and height of the feature. However, the three-rectangle feature had non-zero mean value because the area for the dark rectangles was greater than the area for the white rectangle, as shown in Figure 2.9. Therefore, for the three-rectangle feature the mean of the possible values was calculated and the opposite number of the mean was added to *f(x)* before applying the scaling equation.

5.3 EXPERIMENTS

In the first set of experiments the face detector described in Chapter 3 and the cascaded face detector by Viola and Jones (2001) described in Subsection 2.6.1 were combined with neural network based gender classifiers. The experiments were carried out with a set of frontal face images captured by a web camera, with the FERET database (Phillips et al., 1998) images and with the images collected from the WWW. The FERET database images were used for training the neural network based gender classifiers and web camera images were used as test images. Additional tests where the WWW images were also used were carried out with the cascaded detector.

Since several classification methods were compared in the preceding chapter with manually aligned and extracted faces it was natural to continue in this path using automatic face detection. The second set of experiments was carried out with various gender classification methods, most of which were also used in the experiments of the previous chapter, but now combined with the cascaded face detector (Viola and Jones, 2001). The most important target was to find out if there are differences in classification rates between the gender classification methods when automatic face detection preceded the classification.

Finally, in the third set of experiments (Mäkinen and Raisamo, accepted) processing between face detection and gender classification was varied. In practice, five automatic face alignment methods (one of them was "no alignment" condition) and three input face image sizes were used. The timing of the alignment was also varied so that it could occur before or after the detected face was resized. The aim was to study how to improve the gender classification accuracy by means of automatic alignment when automatic face detection preceded the classification.

5.3.1 Combining Blob Face Detector with a Neural Network Gender Classifier

In the first experiment the blob face detector described in Chapter 3 was combined with a neural network gender classifier. The classification accuracy was measured with a set of web camera images.

The neural network classifier was a single-layer perceptron that was inputted with 68*68 size face images. The input pixel values were scaled to range from -0.5 to 0.5 and the network output was also between -0.5 and 0.5. The neural network was trained with 495 images of female faces and 495 images of male faces taken randomly from the FERET database (Phillips et al., 1998). Some faces had facial hair and some wore eyeglasses. The genders were determined by a researcher. The image area used as an input to the classifier during training was also bounded manually. The face area was resized to 68*68 pixels and then histogram equalized by the system.

After the network was trained the system was tested with web camera images of 13 females and 12 males. Some people wore glasses, and some had a moustache or beard. The images were captured over 10 seconds when the person was sitting in front of the web camera. On average there were 98.12 images per person. The images were 320*240 pixels in size. An example image is shown in Figure 5.2.

Figure 5.2. Web camera image used in the experiment.

Each image was analyzed by the face detector and if the probability for a face was calculated to be over 0% input was created for the gender classifier. A face was detected in 89.7% of the images. Face bounding was manually checked for all images. Face bounding was correct if the box contained eyes, nose, and mouth and the box angle was about the same as the face angle in the image. Overall bounding accuracy including all the images where a face was detected was 68.3%.

The bounding accuracy is not very high. However, the rule used for determining if bounding was correct was strict. There were, for example, images where bounding was classified as incorrect because the eyes were only partly inside the box. In practice, the gender of the person could be classified correctly even when the box did not contain eyes or mouth, because a neural network is often able to classify with partially imperfect input.

The gender classification accuracy in all images where a face was found was 71.9%. When we consider only those images where bounding was correct, the gender classification accuracy was 73.1%. However, there were large differences between individuals. Classification accuracies with correctly bounded faces for each person are shown in Figure 5.3. The grayed bars are for the males and the bars with diagonal lines are for the females. As can be seen, the classification accuracy was very low for some females while for most males it was very high.

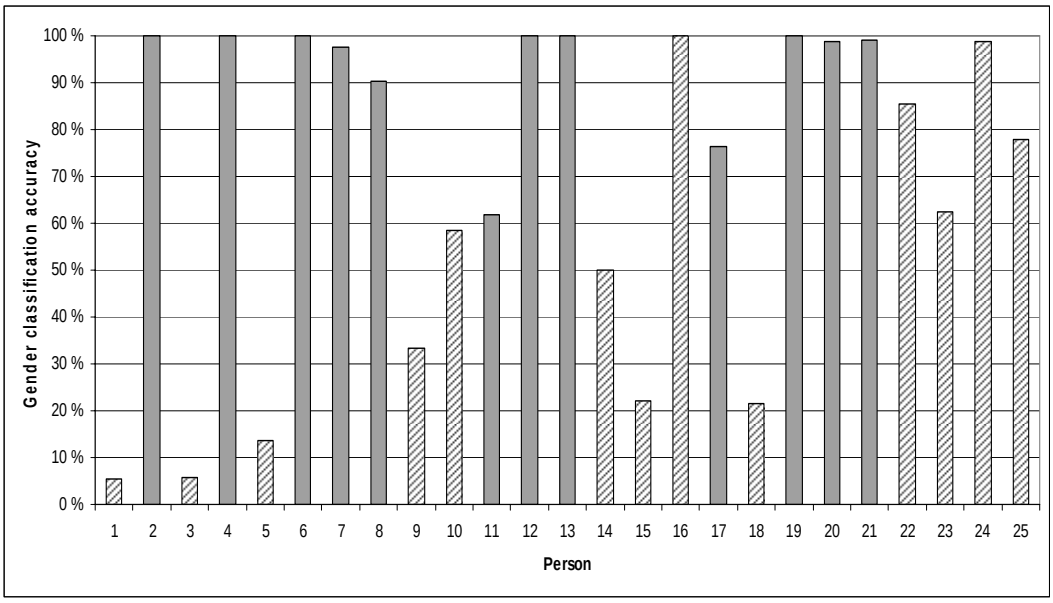


Figure 5.3. Gender classification accuracy for each person when the bounding was correct.

The neural network seemed to be trained so that it was strongly biased to classify inputs as male but equal numbers of female and male images were used for the network training. One possibility for bias would be that the face images in the training set differed significantly from the test face images. An average face image for both sets is shown in Figure 5.4, which shows that there are differences between the sets. The manual bounding for the training images and automatic bounding for the test images is one reason for the differences. However, as the experiment in the following subsection shows, different bounding is probably not the reason for the biased classification rates. Another reason could be that masculine facial features, such as beards and moustaches make male faces more often distinguishable than the feminine features make female faces.

(a)  (b) 

Figure 5.4. Average face image built (a) from the training image set and (b) from the test image set.

The classification could have been balanced by moving the threshold from zero towards the value -0.5 because the classifier output values were between -0.5 and 0.5, and negative outputs were classified as males and positive outputs as females.

5.3.2 Combining Cascaded Face Detector with a Neural Network Gender Classifier

Similar to the previous experiment, the classification accuracy was measured for a system with combined face detection and gender classification. However, this time the cascaded face detector (Viola and Jones, 2001) was used. The system was tested in two different setups.

In the first setup 946 frontal face images from the FERET database (Phillips et al., 1998) were used for training the gender classifier. The face detector detected the face in all images and the detected faces were used as input during the training. The image set was divided into a training set with 756 images and a validation set with 190 images. Both sets were selected randomly but preserving equal numbers of genders. Single-layer and multi-layer perceptrons with different input layer and hidden layer sizes were tried. The multi-layer perceptron with an input layer of 576 units (24*24 size face image) and a hidden layer of 2 hidden units were the most reliable. The same web camera images that were used in the previous experiment were also used as test images this time.

The face was correctly detected and bounded in 99.2% of the images. More than one face was detected in 1.4% of the images and because there was only one real face per image the extra face was a false positive. The gender classification rate for the correctly detected and bounded faces was 84.0%. For females the classification rate was 93.4% and for males it was 73.8%.

There were also differences between individuals. The classification rates for all the people are shown in Figure 5.5. The grayed bars are for the males and the bars with diagonal lines for the females. There was one male whose classification rate was 4.0% and another who had 28.7%. The rest had classification rates over 50% and there were 9 subjects who had 100% classification rate (6 females and 3 males.)

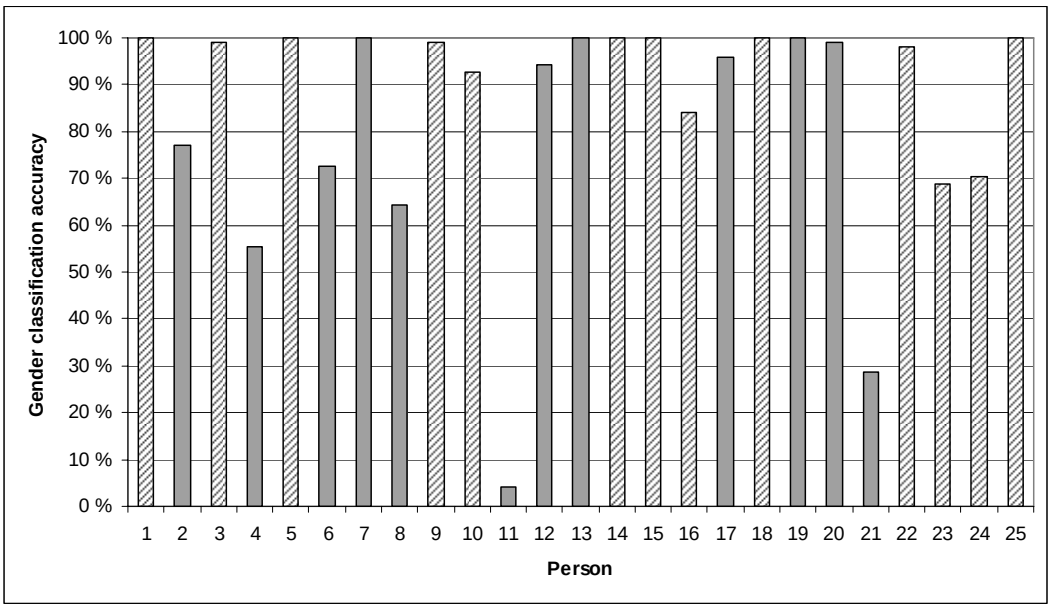


Figure 5.5. Gender classification accuracy for each person.

In the second setup both FERET images and images collected from the WWW were used as training and test images. The 946 frontal face images from the FERET database used in the first setup were also used in this setup. The WWW images contained varying quality face images, 2360 female and 2360 male faces automatically detected by the cascaded detector. Examples of the WWW images are shown in Figure 5.6. The WWW and FERET images were put together and 5-fold cross-validation tests were done with them so that 80% of the images were in the training set, 2% in the validation set and 18% in the test set at a time. The face images were scaled to a size of 24*24 pixels. This time a multi-layer perceptron with 4 hidden nodes was the most reliable. The classification accuracy was also measured for web camera images. The original web camera image set containing images of the 23 people was extended, so that there were images of 24 females and 23 males.

(a) (b)

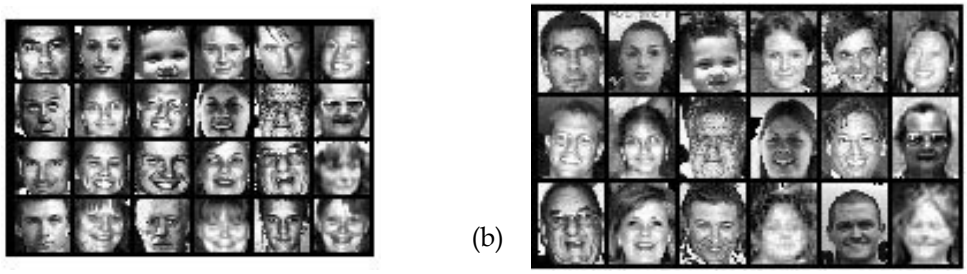


Figure 5.6. Faces detected by the cascaded face detector that have been histogram equalized. (a) Face images resized to 24*24 pixels. (b) Face area increased and resized to size of 28*36 pixels.

The faces detected by the cascaded detector contained only little or no hair. Since better classification rates have been achieved when hair has been

included in the face images (Abdi et al., 1995; Lyons et al., 2000) it was considered interesting to find out if this was also the case when using automatically detected faces in classification. Therefore, this was also studied. The detected area was increased by adding 20% to the width, 40% to the top and 12% to the bottom because this provided face images that usually contained hair in addition to the face but as little as possible of any other data of the image. However, in some cases the area could not be increased as much as intended because the image borders came across. In these cases the image was removed. In some cases a hat or some other object, for example a hand, was in front of the hair or the image was otherwise considered to be of poor quality and was removed. After this there were 3805 WWW images and 760 FERET images, both containing equal numbers of both genders. With the resized images 28*36 pixel size was used (instead of 24*24 pixel size) because this corresponded closely to the image growing percentages. Examples of the WWW images with hair information are shown in Figure 5.6. Again, a neural network with 4 hidden nodes was used.

The classification rates for the second setup are shown in Table 5.2. As can be seen from the results, the classification rates were better for the FERET and web camera images than for the WWW images. The classification rate increased for the FERET images but decreased for the WWW images when the hair data was included in the face images. With web camera images there are no noteworthy differences with different image sizes.

Table 5.2. Classification rates for the image sets.

Train and validation set	Test set	Face images with no hair (24*24 input size) Classification accuracy %			Face images with hair (28*36 input size) Classification accuracy %		
		Female	Male	Average	Female	Male	Average
FERET and WWW	WWW	74.92%	69.44%	72.18%	63.03%	61.50%	62.38%
FERET and WWW	FERET	66.91%	88.69%	77.79%	75.07%	89.51%	82.34%
FERET and WWW	Web camera	70.83%	95.65%	83.24%	66.67%	95.65%	81.16%

The classification rates in both setups for the web camera images were better than in the previous experiment where the blob face detector was used. The important difference between this and the previous experiment

was that in this experiment automatic face detection was used for both training and test images. This allowed more constant input for the neural network. However, the bias for a better classification rate for males did not disappear in the second setup even when using automatically detected faces for gender classifier training.

5.3.3 Comparison of Gender Classifiers Combined with Cascaded Face Detector

Two experiments were carried out to compare different gender classifiers. In the first experiments threshold Adaboost, mean Adaboost, LUT Adaboost (Wu et al., 2003a), threshold perceptron, mean perceptron, and LUT perceptron were compared. The three last mentioned classifiers were multi-layer perceptrons that used features selected by the Adaboost classifier.

In the experiment each system was trained with 473 female face images and 473 male face images from the FERET database. Then the systems were tested with 45 web camera images (22 females and 23 males). The faces were first detected by the cascaded detector (Viola and Jones, 2001) both in the training and testing phases. The faces found were resized to 24*24 pixels before calculating feature values. Eight bins were used with the LUT Adaboost. The feature values were scaled and used as input to the perceptron, which had one hidden layer with two hidden nodes. Several perceptrons were trained with each feature selection method and the results for the perceptrons with the best average classification rates are shown in Table 5.3. All the 946 images were used when features were selected by Adaboost because of its over-fitting resistance (Freund and Schapire, 1997). With perceptrons 756 images were used for the actual training and 190 images for the validation to avoid over-fitting to the training data. However, there did not seem to be over-fitting, possibly because all the 946 FERET images were used for the feature selection and the validation images were part of that set. With the Adaboost classifiers several different thresholds were experimented with. The best average results are shown in Table 5.3 although in some cases more balanced performance between males and females would have been achieved if a different threshold had been used.

There were no great differences between the classifiers or conditions. There was no statistically significant difference in classification accuracy between perceptrons and Adaboost classifiers (Wilcoxon signed-rank test: $z_9 = 1.198$, $p = 0.231$). However, the average classification rate was better for the perceptrons in six cases of nine and equal in one case. Furthermore, there were no statistically significant differences between the conditions with different numbers of features (50 vs. 200 features: $z_6 = 1.156$, $p = 0.248$; 200 vs. 500 features: $z_6 = 1.355$, $p = 0.176$; 50 vs. 500 features: $z_6 = 1.625$, $p = 0.104$) although the classification rates were higher in average when more features were added. Finally, there was a statistically significant difference

between threshold Adaboost and LUT Adaboost when they were compared as feature selection methods (z_6 = 2.060, p = 0.039, threshold Adaboost better selection method) but there were no statistically significant differences in other cases (mean Adaboost vs. threshold Adaboost: z_6 = 0.542, p = 0.588; mean Adaboost vs. LUT Adaboost: z_6 = 0.674, p = 0.500).

Table 5.3. Results for the Adaboost and perceptron classifiers with the web camera images.

Classifier	Number of features	Classification accuracy %		
		Female	Male	Average
Threshold Adaboost	50	70.83%	86.96%	78.72%
	200	62.50%	86.96%	74.47%
	500	75.00%	73.91%	74.47%
Threshold perceptron	50	66.67%	82.61%	74.47%
	200	70.83%	82.61%	76.60%
	500	75.00%	82.61%	78.72%
LUT Adaboost	50	58.33%	69.57%	68.83%
	200	54.17%	91.30%	72.34%
	500	62.50%	86.96%	74.47%
LUT perceptron	50	62.50%	82.61%	72.34%
	200	54.17%	95.65%	74.47%
	500	91.67%	52.17%	72.34%
Mean Adaboost	50	45.83%	82.61%	68.83%
	200	70.83%	69.57%	70.21%
	500	68.18%	95.65%	82.22%
Mean perceptron	50	45.83%	82.61%	68.83%
	200	75.00%	78.26%	76.60%
	500	83.33%	82.61%	82.98%

The feature weights for the perceptrons with 50 features are visualized in Figure 5.7. Since each input node corresponds to a feature and each input node has a connection to both hidden nodes, the absolute values of both connection weights were summed to get the total feature weight. The features selected first tended to have stronger weight than the features selected later. This was to be expected, since the Adaboost algorithm

selects such features first that separate the classes in the example image sets best. However, after the 50 first features there were no obvious differences in the feature weights.

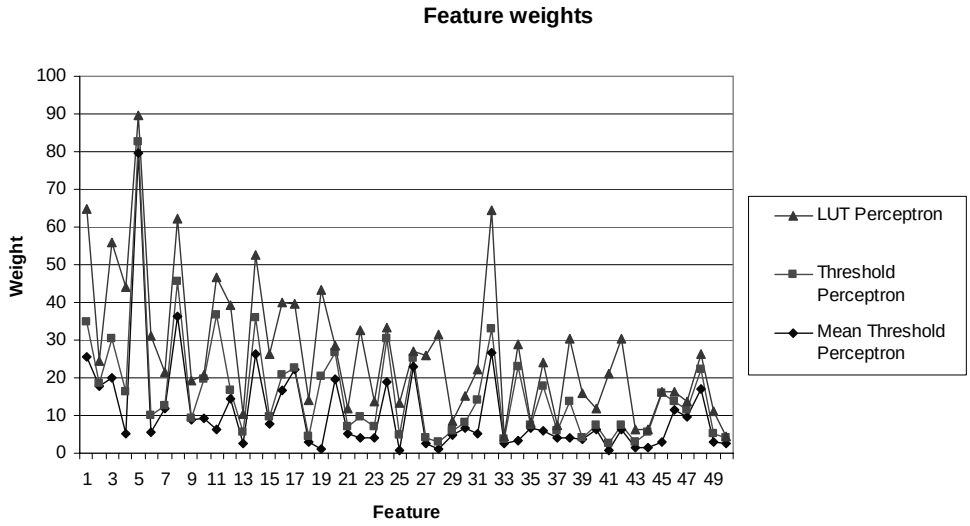


Figure 5.7. First 50 feature weights for the perceptrons.

The first five features selected by each Adaboost method are shown in Figure 5.8. The first three features are similar to each other. They only differ in size, but after that there begins to be variation. The same variation cannot be seen with the feature weights. Instead, the weight variation seems to be more dependent on the feature selection round. The way that Adaboost weights the training examples in each round of the feature selection probably has an effect on this.

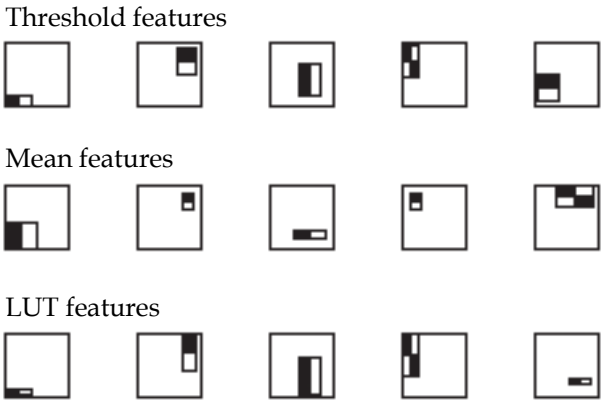


Figure 5.8. First five features selected by the Adaboost methods.

In the second experiment two appearance-based gender classification methods and four feature-based methods were compared. The appearance-based methods were multilayer neural network and SVM. Both methods took histogram equalized image pixels as input. The

feature-based methods were threshold Adaboost, LUT Adaboost (Wu et al., 2003a), mean Adaboost, and an SVM classifier with LBP features. Two image databases were used in the experiment: the FERET image database (Phillips et al., 1998) and the WWW image database. The methods were tested with face images with and without hair. The face images were detected by the face detector and faces without hair were scaled to a size of 24*24 pixels. There were 900 FERET face images and 4720 WWW face images. The face images with hair were created by adding 10% to width on the both sides (20% in total), 40% to the top and 12% to the bottom of the image and then scaling them to the size of 32*40 pixels. There were 760 FERET face images and 3808 WWW face images with hair. The number of face images with hair was smaller than those without hair because some face areas would have gone partially beyond the original image bounds and such faces had to be omitted. For each experiment 80% of the FERET images were put in the training set and 20% of the FERET images in the test set. The WWW images were used only as test images.

Table 5.4. Best parameters for the methods with face images with and without hair.

	Face images without hair (24*24)	**Face images with hair (32*40)**
Neural Network		
Number of hidden nodes	1	1
Input-hidden layer learning rate	0.04163	0.04163
Hidden-output layer learning rate	0.7071068	0.7071068
SVM (RBF Kernel)		
C	8.0	2.0
Γ	0.0078125	0.0078125
LBP + SVM (RBF Kernel)		
C	8.0	2.0
Γ	0.03125	0.0078125
LUT Adaboost		
Number of bins	6	12

Because with the neural network and SVM there is a danger of over-fitting to the training data, some of the training images were used for validation when selecting optimal parameters for the methods. For the neural network 2% of the training images were separated in the validation set and several neural networks were trained using different numbers of hidden neurons and learning rates. Each neural network was then tested with the test images. For the SVM the best parameters were searched with five-fold cross-validation, so that 20% of the training images were in the validation set at a time. After the optimal parameters had been selected the final classifier was trained with the whole training set. 500 features were selected for each Adaboost classifier. For the LUT Adaboost 4, 6, 8 and 12 bins were tried. The best parameters found experimentally for different image sizes with and without hair are shown in Table 5.4.

The best classification rates for each method are shown in Table 5.5. As can be seen, there are differences between the classifiers but the differences between with and without hair conditions for each classifier are as great as the differences between the classifiers. For example, SVM with pixel based input had the best classification rate when the FERET images without hair were used but with the FERET images with hair it had the poorest classification rate.

The effect of including or excluding facial hair seems to depend on the image set, because with the FERET images classification accuracy was better with 4 out of the 6 methods, while with the WWW images the situation was just the opposite. The reason could be that when hair is included in the image some background is also included and with the FERET images the background was fairly uniform unlike with the WWW images. Non-uniform background could cause problems for the classifiers. However, the difference between "with hair" condition and "without hair" condition was not statistically significant (Wilcoxon signed-rank test for FERET images: $z_6 = 0.734$, $p = 0.463$; WWW images: $z_6 = 1.363$, $p = 0.173$).

Table 5.5. Classification accuracies for the classifiers.

	FERET images Classification accuracy %			WWW images Classification accuracy %		
Method	**without hair (24*24)**	**with hair (32*40)**	**Average**	**without hair (24*24)**	**with hair (32*40)**	**Average**
Neural Network	83.89%	90.07%	86.98%	65.95%	61.29%	63.62%
SVM	84.44%	72.85%	78.65%	66.48%	57.41%	61.95%
Threshold Adaboost	82.22%	83.44%	82.83%	66.29%	66.75%	66.52%
LUT Adaboost	80.56%	87.42%	83.99%	66.19%	64.81%	65.50%
Mean Adaboost	76.67%	87.42%	82.05%	66.14%	67.02%	66.58%
LBP + SVM	75.56%	74.83%	75.20%	67.25%	66.54%	66.90%
Average classification accuracy %	80.56%	82.67%	81.61%	66.38%	63.97%	65.18%

5.3.4 Using Face Alignment between Face Detection and Gender Classification

In this experimental setup combinations of automatic face detection, face alignment and gender classification were created. The purpose was to study how automatic face alignment between face detection and gender classification could improve the system performance. For this purpose the three AAM (Cootes and Taylor, 2001) based automatic alignment methods and the profile alignment method described in Subsection 5.2.2 were used. The "no alignment" condition was also included, likewise the manual alignment condition. The cascaded face detector (Viola and Jones, 2001) and four different gender classifiers, multi-layer perceptron with pixel based input, SVM with pixel based input, SVM with LBP features, and threshold Adaboost with Haar-like features were used.

The AAM model was built of 14 face images, 7 female faces and 7 male faces, from the IMM database (Stegmann et al., 2003). Building the AAM model from 50 FERET (Phillips et al., 1998) faces annotated manually for facial landmarks was also tried but it turned out that the alignment results were clearly poorer for the unseen FERET face images with this model.

This was probably due the fact that the FERET images used for building the model varied somewhat more in poses and lighting conditions than the IMM database images.

Although the AAM model was built from the IMM database faces, the FERET image database was used when the actual face detection, face alignment and gender classification combinations were tested because it contained more faces from a larger group of individuals than the IMM database did. Probably superior classification rates would have been achieved with the IMM images since they were more consistent in quality and of higher resolution.

FERET face images were used for training gender classifiers and testing the combinations. The locations of the nasal spine and mouth were annotated manually in addition to the eyes.

The training images were such that AAM alignment and profile alignment were successfully applied to the detected faces. The determination of the successful alignment was such that:

1. The Euclidean distance between real left and right eye center was calculated.

2. The Euclidean distances from the automatically determined facial feature locations to the real ones were calculated (e.g. automatically determined mouth center distance to real mouth center.)

3. The ratios of the distances to real eye distance were calculated with the equation $r = d_f / d_e$, where r was the calculated ratio, d_f was the feature distance, and d_e was the real eye distance.

4. The calculated ratios were compared to predetermined ratios and if all the calculated ratios were smaller than the predetermined ratios then the automatic alignment was considered to be successful and otherwise unsuccessful.

The features used to determine successful alignment were the eye centers, the nasal spine and the mouth center for AAM alignment, and the eye centers for profile alignment. Figure 5.9 shows the facial landmarks and the eye distance used for determining successful alignment. The predetermined ratio was 0.25 for the eye centers, and 0.2 for the nasal spine and the mouth center. A ratio of 1 would mean that the distance from the automatically determined facial feature location to the real location was one eye distance.

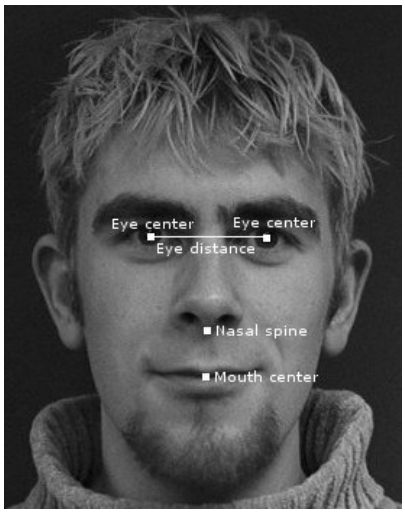


Figure 5.9. The decision whether the alignment is successful or not is based on the facial landmarks and on the eye distance shown in the image. The example face is from the IMM database (Stegmann et al., 2003).

With the profile alignment it was possible that the eye centers were not located but the images were still preserved in the training set. The face location determined directly by the face detector was used in these cases instead of the profile alignment. The same procedure was followed while testing. Naturally, the badly aligning face images were not removed from the test images since the aim was to find out if the alignment was useful when used in combination with face detection and gender classification. In many real applications there is no way to determine if alignment has been successful or not after doing face detection and before doing gender classification. However, such face images were removed from the test set that would have led face area to be partially or wholly beyond this image bounds after alignment.

The number of images for training was 304 and for testing 107. Both sets were the same as those used in Chapter 4 in the experiment where face orientation, size, and face image offset were varied. Since the sets were equal it was possible to compare the results achieved in the experiments.

The experiments proceeded so that first the gender classifiers were trained using automatically detected and, in some combinations, aligned faces. If an alignment method was used then it was used during both training and testing. Only successfully detected and automatically well aligning faces were used for training and the same faces with or without alignment were used for training all the recognizers. Separate classifiers were trained for each image resolution/alignment method/classifier combination. After training the classifiers they were tested with the test images. There were several test variables that were varied in the experiments and this way the 120 combinations were produced each with a unique test condition. The variables and their conditions are shown in Table 5.6. Note that timing of the alignment had no effect on the combinations with the "no alignment"

and "manual alignment" conditions. In total, the tests produced 120 classification rates, one for each combination, that were used in the analysis below.

Table 5.6. Test variables used to create the 120 detector/gender classification combinations.

Variable	Nmber of conditions	Conditions
Gender classification method	4	SVM with LBP features Neural network with face pixels SVM with face pixels Adaboost with Haar-like features
Alignment method	6	None Manual alignment with eyes Profile AAM with eyes AAM with eyes and nasal spine AAM with eyes and center of the mouth
Input image size	3	24*24 36*36 48*48
Timing of the alignment	2	Alignment before resizing face image Alignment after resizing face image

For each combination having a neural network, two neural networks were trained: a network with one hidden neuron and another with two hidden neurons, and the better performing neural networks were used when reporting the results. For all the other methods there was one trained classifier per combination.

When AAM shape was fitted to a face the maximum number of iterations allowed was 50. The initial position of the AAM shape was horizontally at the middle of the detected face. Vertically the initial position was slightly below the detected face center because a part of the jaw tended to be

below the detected face area. The initial shape was also scaled so that the shape had the same width as the detected face.

The training proceeded as in the previous experiments. The Adaboost classifiers were trained with the whole training set. All the Adaboost classifiers had 500 Haar-like features. With the neural networks 2% of the training images were separated to the validation set. The neural networks were trained for 1000 rounds and the neural network of the round with the lowest validation error was selected for the testing. For SVM the best parameters (cost and gamma) were searched using five-fold cross-validation. 20% of the training images were in the validation set at the time. After the best parameters were found the final SVM classifier was trained with the whole training set. After training the classifiers each were tested with the test face images.

The average gender classification rate for all cases where alignment was used, either profile or AAM, was 82.1% as can be seen in Table 5.7. A somewhat surprising result was that the average gender classification rate was higher when alignment was not used. It was 84.6% with no alignment. The difference was also statistically significant (Wilcoxon signed-rank test: $z_{12} = 2.118$, $p = 0.034$.) If we consider AAM and profile alignment separately there was no statistically significant difference for AAM alignment ($z_{12} = 1.804$, $p = 0.071$) but for profile alignment there was ($z_{12} = 2.002$, $p = 0.045$.) Anyhow, the average classification rate for AAM alignment was 82.0% and for profile alignment 82.4%, which were both smaller when compared to "no alignment" cases. The only classifier with some improvement in the performance when automatic alignment was used was SVM with LBP features and even so only when AAM alignments were used. Manual alignment yielded slightly better results (87.1%) than "no alignment" in average but the difference was not statistically significant.

Table 5.7. Average classification rates for methods with different alignment types.

	Classification rate %				
Alignment	**LBP+SVM**	**Neural network**	**SVM**	**Threshold Adaboost**	**Average**
None	77.88%	88.47%	86.29%	85.67%	84.58%
Manual	82.24%	88.47%	90.65%	86.92%	87.07%
Automatic	79.17%	83.29%	83.37%	82.59%	82.11%
Average classification rate %	79.76%	86.74%	86.77%	85.06%	84.58%

When the automatic alignments were compared to each otherit can be seen in Table 5.8 that the AAM alignment with eyes and mouth produced the best average classification rate. The AAM alignment with eyes and nasal spine resulted in the poorest classification rate. However, the differences were not great, and the difference of the AAM eyes and mouth alignment to the AAM eyes and nasal spine alignment was the only statistically significant one ($z_{24} = 2.583$, $p = 0.010$.)

Table 5.8. Average classification rates for methods with different alignments.

	Classification rate %				
Alignment	**LBP+SVM**	**Neural network**	**SVM**	**Threshold Adaboost**	**Average**
None	77.88%	88.47%	86.29%	85.67%	84.58%
Manual	82.24%	88.47%	90.65%	86.92%	87.07%
Profile	76.01%	87.23%	82.87%	83.33%	82.36%
AAM with eyes	78.04%	82.09%	83.64%	83.18%	81.74%
AAM eyes and nasal spine	81.78%	78.66%	81.15%	81.78%	80.84%
AAM eyes and mouth	80.84%	85.20%	85.83%	82.09%	83.49%
Average classification rate %	79.47%	85.02%	85.07%	83.83%	83.35%

All the methods achieved the best classification rate with manually aligned faces, as can be seen in Table 5.8. However, with the neural network an equal classification rate was achieved without alignment. The methods had the poorest classification rate with the AAM eyes and nasal spine alignment with the exception of the SVM with LBP features, which yielded the poorest performance with the profile alignment. When the methods were compared with each other, the SVM with pixel based input, the neural network and the Adaboost methods showed statistically significant differences from the SVM with LBP features method using automatic alignment ($z_{24} = 3.018$, $p = 0.003$, $z_{24} = 2.415$, $p = 0.016$, and $z_{24} = 2.420$, $p = 0.016$ respectively.) The situation was the same when comparison was made with manual and no alignment conditions included (SVM vs. LBP+SVM: $z_{30} = 3.518$, $p = 0.000$; Neural network vs. LBP+SVM: $z_{30} = 3.015$, $p = 0.003$; Threshold Adaboost vs. LBP+SVM: $z_{30} = 2.888$, $p = 0.004$). There were no statistically significant differences between the other three methods.

Table 5.9. Average classification rates when using different alignments and alignment was done before or after resizing the face.

	Classification rate %		
Alignment	**Alignment before resize**	**Alignment after resize**	**Average**
Profile	83.80%	80.92%	82.36%
AAM with eyes	82.32%	81.15%	81.74%
AAM eyes and nasal spine	82.40%	79.28%	80.84%
AAM eyes and mouth	84.74%	82.24%	83.49%
Average classification rate %	83.32%	80.90%	82.11%

Table 5.10. Average classification rates for gender classification methods when alignment was done before or after resizing the face.

	Classification rate %		
Classifier	**Alignment before resize**	**Alignment after resize**	**Average**
LBP+SVM	81.46%	76.87%	79.17%
Neural network	84.66%	81.93%	83.30%
SVM	84.35%	82.40%	83.38%
Threshold Adaboost	82.79%	82.40%	82.60%
Average classification rate %	83.32%	80.90%	82.11%

As can be seen from Table 5.9 the alignment methods worked best if they were used before resizing the face image and the result was statistically significant (z_{48} = 3.668, p = 0.0002.) As can be seen from Table 5.10 the situation was the same for all gender classifiers. The order of the alignment and image resizing did not affect the manual alignment or the no alignment cases, so they are not shown in Table 5.9.

The classification rates for different face image sizes are shown in Table 5.11 and in Table 5.12. The best classification rates were achieved with image size 36*36 but the difference from images of size 48*48 was not statistically significant. However, when image size 36*36 was compared to the image size 24*24 the difference was statistically significant (z_{40} = 3.263,

p = 0.001), likewise when images of size 48*48 were compared to images of size 24*24 (z_{40} = 1.960, p = 0.050.)

Table 5.11. Average classification rates for different alignments with different face sizes.

	Classification rate %			
Alignment	**24*24**	**36*36**	**48*48**	**Average**
None	86.21%	82.94%	84.58%	84.58%
Manual	85.28%	87.38%	88.55%	87.07%
Profile	83.29%	82.94%	80.84%	82.36%
AAM with eyes	78.27%	83.76%	83.18%	81.74%
AAM eyes and nasal spine	77.92%	83.06%	81.54%	80.84%
AAM eyes and mouth	81.31%	85.16%	84.00%	83.49%
Average classification rate %	82.05%	84.21%	83.78%	83.35%

Table 5.12. Average classification rates for gender classification methods with different face sizes.

	Classification rate %			
Classifier	**24*24**	**36*36**	**48*48**	**Average**
LBP+SVM	76.92%	79.07%	82.06%	79.35%
Neural network	84.21%	85.89%	82.90%	84.33%
SVM	82.62%	86.54%	84.02%	84.39%
Threshold Adaboost	81.50%	84.58%	83.93%	83.34%
Average classification rate %	*81.31%*	84.02%	83.23%	82.85%

Finally, the alignment accuracies with each alignment condition were measured because the automatic alignment methods produced poorer classification rates than when no alignment was used. Since measures to calculate alignment error have been proposed in the literature it was decided to use them. Jesorsky et al. (2001) proposed a measure that was based on the real eye locations and estimated eye locations. The equation can be written as

$$d_{eye} = \frac{\max(d_l, d_r)}{\| C_l - C_r \|},$$

where d_l is the distance between real left eye center C_l and estimated left eye center $\widetilde{C}_l$, and d_r is the distance between real right eye center C_r and

estimated right eye center $\widetilde{C}_r$. In addition to the above measure the measures proposed by Rodriguez et al. (2006) were used: horizontal translation (Δ_x), vertical translation (Δ_y), scale (Δ_s), and rotation (Δ_α). These measures provided more specific knowledge of the alignment error type.

The alignment accuracy measures for each alignment condition are shown in Table 5.13. The measures are averages calculated from all test images. For the horizontal translation measure, vertical translation measure, and for the rotation measure the averages were calculated from the absolute values because these values show how much the measure differs from zero on average, while original positive and negative values would produce a value close to zero.

Measures for the "no alignment" condition are also shown. However, one should be careful when comparing the measure values of other alignment conditions to the "no alignment" condition, because in "no alignment" condition the face detector yielded only the location of the face box and not the eye locations. Therefore, the eye locations were estimated from the face box in the "no alignment" condition. The measures that can safely be compared to the other alignment conditions are Δ_x and Δ_α because the horizontal center of the eyes can also be assumed to be in the middle of the face box in the "no alignment" condition and the rotation error for the detected face is always the same as the rotation of the face in the image. All the other alignment conditions can safely be compared with all the measures.

Table 5.13. Alignment measures for each alignment condition.

Alignment	Alignment before resize	Image size	d_{eye}	Δ_s	$\lvert\Delta_x\rvert$	$\lvert\Delta_y\rvert$	$\lvert\Delta_\alpha\rvert$
No alignment			0.31047	1.07776	0.03585	0.27853	2.45126
Profile	Yes		0.19594	1.20140	0.05634	0.06048	2.45126
	No	24*24	0.28016	1.08092	0.05965	0.22617	2.15392
		36*36	0.27838	1.06166	0.04463	0.22273	2.44515
		48*48	0.23113	1.09620	0.04951	0.15483	2.64148
AAM with eyes	Yes		0.10918	0.96449	0.03964	0.05433	1.79862
	No	24*24	0.14888	0.95662	0.08011	0.03692	2.35281
		36*36	0.13511	0.96570	0.07220	0.03696	2.76628
		48*48	0.12788	0.96324	0.06673	0.04070	2.40018
AAM eyes and nasal spine	Yes		0.46098	0.96449	0.04134	0.42846	1.79862
	No	24*24	0.56037	0.95662	0.07893	0.50174	2.35281
		36*36	0.72225	0.96570	0.07104	0.67542	2.76628
		48*48	0.70618	0.96324	0.06813	0.65577	2.40018
AAM eyes and mouth	Yes		0.36788	0.96449	0.03562	0.33977	1.79862
	No	24*24	0.43928	0.95662	0.07249	0.38627	2.35281
		36*36	0.46145	0.96570	0.06800	0.40816	2.76628
		48*48	0.47414	0.96324	0.06384	0.42392	2.40018

As can be seen in Table 5.13 when the $\lvert\Delta_x\rvert$ of the "no alignment" condition is compared to the other alignment conditions only the "AAM alignment with eyes and mouth" condition together with the "alignment happened before resizing" condition had smaller $\lvert\Delta_x\rvert$ error. Indeed, the average classification rate for this condition was 84.74%, which is higher than that

of the "no alignment" condition (84.58%). What is also interesting is that d_{eye} was smaller for all the conditions where alignment occurred before resizing than for the conditions where alignment occurred after resizing. Also, $|\Delta_y|$ was smaller for all the "alignment before resizing" conditions than for the "alignment after resizing" conditions except with the "AAM alignment with eyes" condition. This concurs with the measured classification rates, which were better when the alignment happened before resizing the face images. The $|\Delta_x|$ is the only measure that was consistently smaller for the alignment conditions with 36*36 and 48*48 size images than with 24*24 images where alignment occurred after resizing the face image. There were statistically significant differences in classification accuracies between 24*24 size images and 36*36 size images, and between 24*24 size images and 48*48 size images, so Δ_x is probably one reason for the differences. Finally, one can also see that "AAM eyes and mouth" alignment was more accurate than "AAM eyes and nasal spine" alignment, which was also shown as statistically significant difference in the gender classification rates.

5.4 DISCUSSION

When all the above experiments are considered several conclusions can be drawn. The best combined and fully automatic system was achieved when the cascaded face detector (Viola and Jones, 2001) was combined with a gender classifier without alignment, good quality face images were used and face images were scaled to the size 36*36 pixels when inputted to the classifier. The blob face detector found the faces fairly reliably but the location accuracy of the detection could have been better. Now the gender classification rate for the combination with cascaded face detector and neural network gender classifier was 84% and about 10 percentage units higher than for the combination with the blob detector.

The images collected from the WWW proved to be a challenging target for gender classification. The best classification rates for the WWW face images were just over 70%, for the web camera images just over 80%, and for the FERET images slightly over 90%. The images that are of web camera image quality are commonly available in perceptual user interfaces and 80% classification accuracy is usable for many applications. However, the accuracy can be improved by using a reliable alignment method.

As the results of the last experiment show, the automatic alignment did not increase the gender classification rate. Instead, the classification rate decreased. The effect was the same with all automatic alignment methods whether the alignment was done before or after image resizing. Also, with the exception of the SVM with LBP features, the gender classification

method did not affect the results. The results also support the statement by Shakhnarovich et al. (2002) that face alignment "is not entirely robust in the presence of significant pose, lighting, and quality variation". However, if we take into account that manual alignment actually did increase the classification rates the automatic alignment would also do so if it could be done reliably.

One possible way to improve alignment would be to tune the parameters for the alignment methods. However, for example the Jacobian training scheme that we used with AAM, was found to be good in the study by Stegmann et al. (2003). Also, adding more faces to AAM model building might help. One possibility would be to use some other alignment method than AAM. Wu et al. (2003b) used Simple Direct Appearance Model (SDAM) (Wang, 2003; Xiao, 2002) for alignment. Yet another possible way to improve the alignment results could be to use several methods for locating facial features and do the alignment based on the results of all methods. For example, eyes could be located using three different methods and the average of the locations found would be used for the alignment.

If we consider the effect of face image resizing on alignment and gender classification, the alignment should be done before resizing. The image size after resizing was not that important. The best classification rates were achieved with an image size of 36*36 pixels, but the difference in classification rates from image size of 48*48 pixels was not statistically significant. However, Wu et al. (2003b) also achieved the best gender classification rate with images of size 36*36 pixels. In addition, our results support the statement by Moghaddam and Yang (2002) that the performance of SVM depends mainly on the number of training images and not so much on the input resolution. This also seems to fit for the other gender classification methods, because there were no great classification rate differences with different image sizes for any of the methods.

When the classification rates of the gender classifiers were compared, they performed equally well with the exception of the SVM with LBP features. SVM with LBP features performed clearly worse than the other methods and the difference was statistically significant. However, it had the interesting characteristic that the classification performance improved when image size used as input was increased. This behavior did not happen with the other methods.

Whether hair should be included in the face images seems to depend on the images used in the application. With the FERET images the classification rate improved usually a little when hair was included but with the WWW images the effect was usually the opposite. With web camera images the classification rate was about the same with and without

hair. Different results for the FERET and the WWW images may follow from the fact that with the FERET images the background varied only slightly but with the WWW images the background often had wide variations and there was more background visible when hair was included in the images. If the combined system is used in a perceptual user interface and the scene background can be controlled, then including hair in the face images could be useful for the classification.

The other important issue in addition to achieving a high classification rate is the speed of the combined system. Face detection takes most time especially with large images, so the speed of the detector is the most important factor. Naturally, a system that does not include an alignment step is faster than the one that does. In addition, the hardware and the gender classification method have an effect.

The speed of the system with neural network depends on four main factors: (1) the time spent on scaling the sub-image, (2) the time spent on histogram equalization, (3) the time spent on scaling the input values, and (4) the number of nodes in the network. The speed of an Adaboost classifier depends mainly on the number of the rectangular features in the classifier. The Adaboost method can be combined effectively with the cascaded detector, especially if no alignment is used (Shakhnarovich et al., 2002). The creation of the integral image takes some time, but because it is already created for the cascaded face detector it does not need to be created separately for the gender classifier. With SVM the feature vector size and the number of support vectors determine the classification speed.

If the classification rate is the most important issue then SVM may be the best classifier, because the best classification rates have been reported for it in many studies (BenAbdelkader and Griffin, 2005; Castrillón-Santana et al., 2003; Moghaddam and Yang, 2002; Yang et al., 2006b) and it also performed well in the experiments described in this and the previous chapter. However, Adaboost probably achieves the highest classification speed and a neural network may offer a very good compromise between speed and accuracy.

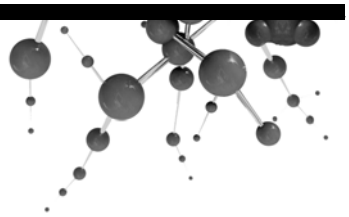
5.5 SUMMARY

Various experiments with fully automatic systems that detected faces and classified genders were described. The blob face detector described in Chapter 3 and the cascaded face detector by Viola and Jones (2001) were the face detectors used. The cascaded face detector, besides being able to analyze gray scale images, proved to locate faces more accurately and was used in most of the experiments. The gender classifiers included neural network, SVM with pixel based input, SVM with LBP features, threshold Adaboost, mean Adaboost, and LUT Adaboost (Wu et al., 2003a). Various

face alignment methods were also used. Depending on the experiment, the FERET database images (Phillips et al., 1998), a set of web camera images or the WWW images were used.

Image quality was an important factor for the gender classification accuracy; FERET images produced the best classification rates while the WWW images produced the worst rates. However, input image size was rather an unimportant factor for classification accuracy. The 36*36 and 48*48 input image sizes were equally good resolutions. The 24*24 image size was almost as good as the other two. Including hair in the face images was useful with the FERET images but with the WWW images the classification accuracy suffered. Automatic face alignment would probably increase the classification accuracy if reliable enough although in the experiments this did not happen. However, the classification rates for the manually aligned faces were higher than for the faces detected by the detector, which indicates that automatic alignment would be useful if the alignment methods were reliable enough. Of the methods SVM with pixel based input was on average the most reliable.

Considering the systems from the HCI viewpoint, besides accuracy, the classification speed is also of importance. Face detection takes the most processing time but the cascaded face detector can analyze 320*240 size images in real-time with standard PC. Thus all the experimental combined systems run in real-time on a standard PC.



6 Tools for Face Analysis

6.1 INTRODUCTION

To carry out the experiments described in Chapters 3 and 4 various software tools were needed. They were needed to manage photos and face data, to train and experiment with face detectors and with gender classifiers, and for other smaller tasks. Most of the tools were implemented as a part of this dissertation for the specific purposes because such tools did not exist. The most important tools are available at the web address http://www.cs.uta.fi/hci/mmig/vision/.

However, some freely available software such as OpenCV (2006), LIBSVM (Chang and Lin, 2001), AAM-API (Stegmann et al., 2003), and Microsoft VisionSDK (2000) was utilized. OpenCV provided functions for image handling, such as reading an image from a file, rotating the image, and displaying it. It also provided an implementation of the cascaded face detector (Viola and Jones, 2001) and it was used in the experiments where the detector was needed. The default frontal face cascade provided with the detector was also used. The LIBSVM was used for training and testing the SVM classifiers. The AAM-API provided tools to create AAM shape models and C++ functions to fit the shape models to the novel faces. Microsoft VisionSDK was used for image capturing from the web camera but because the current version of OpenCV provides capturing functions and Microsoft VisionSDK is no longer available, OpenCV will be used in the future.

The most important tools that were implemented in this dissertation are described next. The first one is the tool that was important when the rules were defined for the blob face detector. Next, the face database tool that was used for organizing photos and face data is described. After that a tool that was used for neural network training and for most of the gender

classification experiments is described. Finally, a tool that was used for Adaboost training is described.

6.2 Blob Face Detector Tool

When the blob face detector was developed there was a need for a tool to visually inspect the results of the face detection and adjust the rules and other parameters for the detector. The most important tool that was used for the purpose is now introduced.

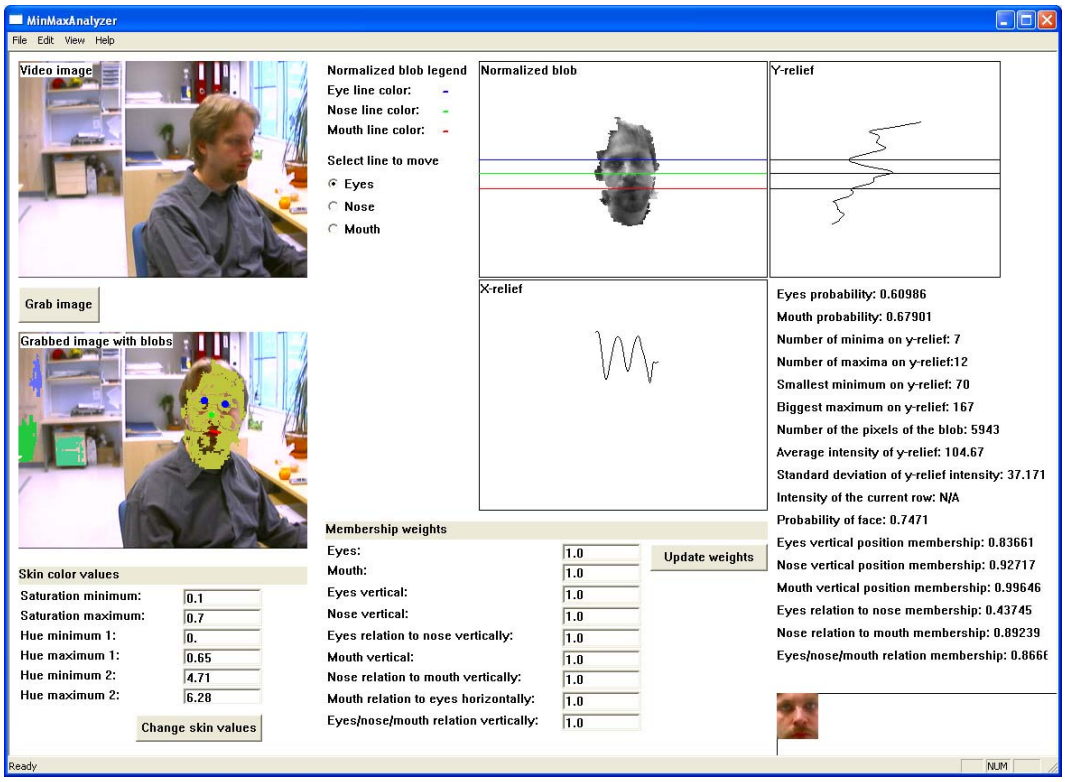


Figure 6.1. User interface of the blob face detector tool.

The user interface of the blob face detector tool is shown in Figure 6.1. The video image captured from the web camera is shown on the top left corner. When the user clicked the "Grab image" button the current video frame was stored and shown below the video image. If there were skin colored blobs on the image then they were also shown in the grabbed image. By clicking a blob facial feature candidate search, feature selection and face probability calculation was done for the blob. As a result, the blob and its vertical and horizontal intensity profiles were shown on the top right part of the user interface. Various probabilities and other information of the blob were also shown. Furthermore, possibly detected facial features were shown in the grabbed image, and the face rectangle that was created based on the eye locations was shown at the bottom right corner of the user interface.

It was useful to see the blobs and the information for the selected blob but the real value of the tool came from the adjustments that could be made for the skin color model and for the facial feature candidate search. These values were modified according to the experience that the continued use of the tool provided.

6.2.1 Adjustment of the Skin Color Model

It was important to select proper values for the skin color model so that the face areas would be found with as little background as possible. Since the camera settings and lighting affected the skin color and color of the other objects it was necessary to adjust the skin color model for the different imaging conditions.

The skin color model could be adjusted with the controls at the bottom left corner on the user interface. The skin color model was based on the HSV color model and the hue and saturation values were used. Possible values for saturation were between 0 and 1 and for the hue between 0 and 2π radians. There were two maxima and minima for the hue because the values of the red were at both ends of the hue "circle" while the green and blue were between.

By changing the values the number and size of the blobs detected was changed in the grabbed image based on the new values. Then by clicking a blob the probabilities and possible facial feature locations were calculated for it.

6.2.2 Adjustment of the Facial Feature Candidate Search

Although there was general anthropometric face information available the selection of the probability rules and their weights was done experimentally. It was easiest to visually inspect what kinds of vertical and horizontal profiles the blobs, face and non-face, typically had. A graphical tool was also the easiest and the fastest way to experiment with different weights for the probability rules.

Facial feature candidate selection could be manually affected by two means: moving a horizontal line for the eyes, the nose, or the mouth in the "normalized blob" view or by changing the weights for the probability rules. When the user moved a line in the "normalized blob" view the horizontal profile of the corresponding row and the probabilities shown were updated. When the user changed the weights for the probability rules the facial feature candidates were searched that produced the highest face probability with the weights. The probabilities were updated and the lines in the normalized blob view were moved to the new locations.

6.3 FACE DATABASE TOOL

A large set of face images was needed to carry out the experiments. Some of the images were from public databases such as the FERET database (Phillips et al., 1998) or the IMM database (Stegmann et al., 2003), some were collected from the WWW, and some were captured with a web camera. The face database tool was created to facilitate handling of these images. The tool was created for Microsoft Windows system using Visual C++.

Figure 6.2. User interface of the face database tool.

The user interface of the tool is shown in Figure 6.2. The database that was opened and the images that were loaded are shown in the window. The face rectangles and other information such as the facial features locations can be shown if the data has been added to the faces. Most of the functionality is available through the menus.

The tool included various small features such as storing images in a database without face information, saving the faces as images, viewing vertical profiles of the faces, and fitting AAM shapes to the faces. These facilitated some tasks and many of the figures in this thesis are also results of these features. The tool had two main features, face editing and storing, and these features are described next.

6.3.1 Editing Faces

There were two ways to add information on faces in the images. One was by drawing a face rectangle on the image and the other was by detecting the faces automatically. The face rectangles could be resized and facial feature locations could be added to the face by clicking the nose tip, for example. It was also possible to calculate facial feature positions for the

already defined face rectangles using the profile alignment and to fit AAM models to the faces, and to delete such faces that did not align properly. Finally, gender ground truths could be added for the faces.

There were sometimes thousands of images and even more faces that needed to be annotated, so effort was made to make the annotation as efficient as possible. For example, the gender ground truth information was added so that when the user pressed either m (male) or f (female) on the keyboard the gender information was stored, and the next face was automatically selected or if there were no more faces in the image then the first face in the next image was selected. The user could also undo the actions by pressing u on the keyboard.

6.3.2 Storing Face Data in Varying Formats

Most of the experiments were carried out on a Microsoft Windows system but the SVM training and testing was done on a Linux system and the Adaboost classifiers were trained on an IBM eServer Cluster provided by CSC - Scientific Computing Ltd. The cluster consisted of 8 IBM p690 nodes and each node had 32 processors. Adaboost training was done in one of the nodes and 16 processors were used for the major training tasks.

It would have been possible to copy the face databases to all three systems and implement, for example, integral image creation for the Adaboost classifiers on IBM eServer Cluster. However, it was considered practical that as little code as possible was duplicated between the systems. Therefore, the feature vectors (inputs) for the SVM classifiers and integral images for the Adaboost classifier training were created using the face database tool. The SVM data for the sensitivity analysis (see Chapter 4) was also created with the tool. The SVM feature vectors were stored in a text file in LIBSVM format and the integral images were stored as separate text files. The text format was chosen for the integral images instead of the binary one because the text format was easily readable from the training program in the cluster. The binary file representation in a Microsoft Windows system is little-endian while in IBM eServer Cluster it is big-endian. The C++ applications in the both systems use corresponding binary file format and reading a binary file in the cluster would have been more complex than reading a text file.

6.4 Face Analysis Tool

Face Analysis Tool was mostly used for running various tests. However, all the neural networks were trained using this tool. The tool had different views and when a face database was opened the view looked almost identical to the face database tool as shown in Figure 6.3.

Figure 6.3. The face analysis tool view is almost identical to the face database tool when a face database is opened.

As with the face database tool there were some miscellaneous features. For an Adaboost classifier there was an option to drop out a specified number of Haar-like features and then save the classifier. It was also possible to view the Haar-like features and histogram equalized and resized faces.

6.4.1 Neural Network Training

Before neural networks could be trained two face databases needed to be opened: training database and validation database. As was explained in 2.4.2, both the training face examples and validation face examples were needed to avoid over-fitting to the training data.

If the user wanted to train a neural network with histogram equalized face images then after the databases had been opened a neural network could be created and trained by selecting corresponding action from the Network menu. However, in the case of training a neural network that used Haar-like features as input (see Subsection 5.2.3) the user had to load the features first.

When a new network was created the user could select parameters for it such as number of hidden nodes in the hidden layer and learning rates for the hidden and output layers. When the training was initiated there was a choice of how many rounds the network was trained and whether the examples were shuffled after every round. It was possible to repeat the training as many times as necessary.

Figure 6.4 displays the view presented to the user after training. The y-axis represented the training error and the x-axis the training round. There were two curves: one showed training error and the other validation error. The user could select a specific round from the view and inspect the errors

for the round. The user could also save the network from the selected round for later use.

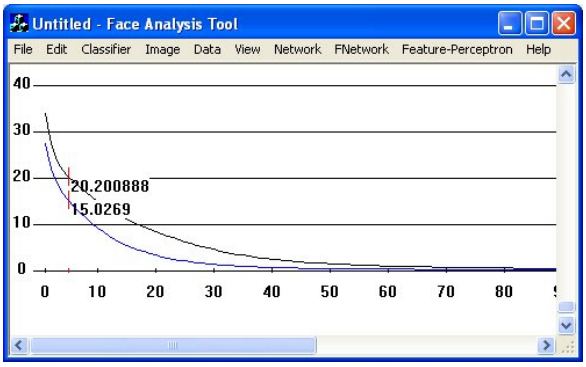


Figure 6.4. Network training error view.

6.4.2 Testing Gender Classifiers

Running the tests was simple. After a test face database was opened the user just had to select a test and a trained classifier. Then the test was run and a test log was created and stored for further analysis. The user could also run a group of tests at the same run.

The format of the log depended on the test and on the analysis that was to be carried out for the results. For example, a test that was carried out with Adaboost to find out gender classification accuracy with manually aligned faces and with 24*24 pixel face images created a log that included a line for each analyzed face and each line contained the image name that contained the face, the face location, the real gender of the face, and the classified gender of the face.

6.5 Parallel Training Tool for Discrete Adaboost

The parallel training tool now described was a command line tool. It was used with a set of training face examples to train an Adaboost classifier with a certain number of features. It worked with all the Adaboost variants, the threshold Adaboost, the mean Adaboost, and the LUT Adaboost (Wu et al., 2003a) that were used in the experiments.

The training algorithm for the discrete Adaboost was given in Subsection 2.4.2. Training the Adaboost classifier with a sequential algorithm is very time consuming in the case of gender classification. The reason is that there is a large set of Haar-like features to select from at each feature selection round. With the 24*24 image size there are 125,616 features, with the 36*36 size there are 645,516 features, and with the 48*48 size there are 2,055,360 features to select from. If there are, for example, 1000 face images for the training then with the 48*48 size images this means that (almost) 2,055,360,000 feature errors have to be calculated in every training round. If there is enough memory capacity in the computer then it is sufficient to

calculate the value for each feature with each face image only once at the beginning of the training and store the values in the memory. However, even then the feature errors have to be calculated for each feature with each image in every round because the weights of the images change in every round.

In practice, the training of an Adaboost classifier with 500 features using 1,000 face images on a computer with a Pentium III processor would have taken a day or a few days. Since there was a need to train several hundred classifiers for the experiments it was necessary to make the training of the classifiers perform faster. The solution was to parallelize the training algorithm.

6.5.1 Algorithm for Parallel Training

The parallel algorithm that I implemented was based on a suggestion by Gregory Shaknarovich. He advised me to divide the feature value and error calculation between many processes so that in each feature selection round each process would calculate the errors for a feature subset and new weights for an image subset. I implemented the parallel algorithm using Message Passing Interface (MPI) and ran it on an IBM eServer Cluster 1600.

An issue that had to be solved before the parallel algorithm could be used was how to make the messaging between the computing tasks efficient (each task was a process running on a different CPU). If the tasks had waited for messages from the other tasks for long periods of time then the benefit of dividing the feature error calculation between the tasks would have diminished. The pseudocode for the parallel algorithm that was found to be efficient with 16 processors (i.e. the maximum number of processors experimented with) is shown in Figure 6.5. The MPI operations that were used in the actual C++ implementation on the specific places are also shown.

```
Divide the features equally between N computing tasks.
Calculate the feature values with the face images for each feature.
FOR T rounds where T is the number of features to select
        Normalize the face image weights so that the sum of the weights is 1
        Calculate the error for each feature
        IF my task ID equals to 0
                FOR the tasks with IDs from j = 1 to N
                        MPI_Recv: receive the smallest feature error from the task j
                FOR the tasks with IDs from j = 1 to N
                        IF the task j has the feature with the smallest error
                                MPI_Send: request the feature from the task j
                                MPI_Recv: receive the feature from the task j
                        ELSE
                                MPI_Send: notify the task j that it has not the
                                feature with the smallest error
                                MPI_Send: send the smallest error for the task j
```

```
            Store the feature with the smallest error on this round
        ELSE
            MPI_Send: send the smallest feature error to task with ID 0
            MPI_Recv: receive notification from task with ID 0 whether I have
            the feature with the smallest error
            IF I am the task that has the feature with the smallest error
                MPI_Send: send the feature with the smallest error to task
                with ID 0
            ELSE
                MPI_Recv: receive the smallest error from the task with ID 0
        IF I am the task t that had the feature that was selected on this round
            Divide the feature values of the selected feature to N parts,
            separately for the positive and negative example face images
            FOR all the tasks, j = 0 to N, j≠t
                MPI_Send: send a positive example part of the feature values
                for the task j
                MPI_Send: send the start index of the part for the task j
            Update my part of the positive face examples weights
            FOR all the tasks, j = 0 to N, j≠t
                MPI_Send: send a negative example part of the feature values
                for the task j
                MPI_Send: send the start index of the part for the task j
            Update my part of the negative face examples weights
        ELSE
            MPI_Recv: receive a part of the feature values from the task t
            MPI_Recv: receive the start index of the part t
            Update my part of the positive face examples weights
            MPI_Recv: receive a part of the feature values t
            MPI_Recv: receive the start index of the part t
            Update my part of the negative face examples weights
        MPI_Barrier: Wait for all the tasks to reach this point
        FOR all the tasks with IDs from i = 0 to N
            IF I am the task i
                FOR all the tasks with IDs from j = 0 to N, j≠i
                    MPI_Send: send the positive example weights that I
                    updated for the task j
                    MPI_Send: send the start index of the example
                    weights for the task j
                    MPI_Send: send the negative example weights that I
                    updated for the task j
                    MPI_Send: send the start index of the example
                    weights for the task j
            ELSE
                MPI_Recv: receive the positive example weights from the task
                i
                MPI_Recv: receive the index of the positive example weights
                that were received from the task i
                MPI_Recv: receive the negative example weights from the task
                i
                MPI_Recv: receive the index of the negative example weights
                that were received from the task i
            MPI_Barrier: Wait for all the tasks to reach this point
IF my task ID equals to 0
    Form the strong classifier of the selected features
```

Figure 6.5. Pseudocode for the Discrete Adaboost parallel training algorithm.

At first the features were divided equally between the tasks and the feature values were calculated within each task. Then the training took place in rounds so that in each round each computing task selected the feature with the smallest training error and sent the error to the computing task with id 0. The task with id 0 then requested the feature from the task that had the feature with the smallest error and sent the smallest error to the other tasks. It also stored the selected feature. Then the task that had the feature with the smallest error divided the corresponding feature values between the tasks because it had calculated all the feature values for that feature. There were as many feature values as there were face training example images. Each task updated the part of the face example weights of which it had received the feature values and sent the updated weights to the other tasks. At the end, when all the features had been selected, the task with id 0 formed the strong classifier.

Various other algorithms and MPI operations such MPI_Isend and MPI_Bcast were also tried but the above algorithm proved to be the most efficient. The training of the 500 feature classifier with 946 face images of size 48*48 pixels took less than an hour when the training was divided between 16 processors.

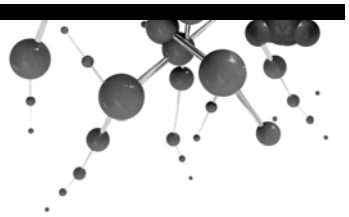
6.6 SUMMARY

The four most important tools developed for face analysis were described. The blob face detector tool was used when the detector was developed. It was used for selecting and adjusting the parameters for the detector. The face database tool was used for organizing face databases and editing the face data on the database images. The face analysis tool was used for experimenting with trained Adaboost and neural network classifiers. It was also used for neural network training. The parallel training tool for discrete Adaboost was used to efficiently train the Adaboost classifiers.

Besides the four tools described numerous smaller tools were implemented. One was crawling software implemented with Java for retrieving images from the World Wide Web. There were also Java programs for sorting a database, for combining databases together, for separating a percentage of the images from our database to another (for example, from training images to validation images), for creating a database file from the images in a directory, for copying images in a database to a specified directory, and for calculating the number of distinct Haar-like features with a certain face image size. A UNIX shell script was also created for training and testing the SVMs with LIBSVM (Chang and Lin, 2001) in the experiments.

The tools described in this chapter were vital to the face analysis research. They facilitated the development of the methods and carrying out the

experiments. The tools also evolved during the years and will do so in the future. One possibility would be to build a toolbox for carrying out gender classifier evaluations and make it publicly available as the separate tools are currently available. This would facilitate the comparison of novel gender classifiers and existing ones.



7 Applications

7.1 Introduction

Face analysis arouses increasing interest among researchers. There are two main reasons for this. The first reason is that the required hardware is nowadays inexpensive enough for many applications and many people already have web cameras at home that can be used in face analysis applications. For example, all currently available Apple computers have an integrated web camera. The second reason is that the face analysis techniques are mature enough to be used in many applications. However, there are still many research problems to be solved.

The applications drive the research. Face analysis techniques can be used in many kinds of applications. As the techniques mature and become more reliable, new types of applications can be built.

In this chapter the main viewpoint is how face analysis has been and could be used in HCI. At first, examples of the existing applications are given. A thorough description of the information kiosk application developed in the University of Tampere using face detection to enhance interaction is presented in its own section. Two ongoing face analysis application projects are then described. The last section contains ideas for future applications of face analysis.

7.2 Face Analysis in Existing Applications

Many applications that include computer vision and face analysis have recently been introduced. The applications in the security and surveillance field, and computer games are probably the best known. However, applications also exist in many other areas.

The applications are divided here into two categories: applications that use face analysis techniques but not other perceptual technologies and applications that use face analysis together with other computer vision technologies, gaze tracking, audio, and haptics.

7.2.1 Applications of Stand-Alone Face analysis

Security and surveillance have gained importance recently. One typical application in this field is face detection and recognition from a video image. For example, suspects can be found at airports and other public places by analyzing captured video automatically and reporting possible matches to the security personnel. Another possibility is to use face recognition as a biometric authentication method instead or with finger print based or iris based methods. Zhao et al. (2003) also listed other face recognition applications than those related to security and surveillance. Some of the specific applications were games, virtual reality, training programs, and human-robot interaction.

Situations where face recognition based authentication could be used include all the situations where identity verification is needed, such as building access, computer access, or even authentication for bank transactions and on-line shopping. For example, Lenovo has included biometric face recognition authentication in some new laptops that it sells in India. The Face Recognition Homepage (2007) lists 21 companies that supply commercial face recognition software, and 22 organizations most of which were companies participated in the latest Face Recognition Vendor Test (FRVT, 2006). Omron released the first commercial face recognition software for mobile phones in 2005 (Omron, 2005). These examples show the commercial potential of face recognition.

However, there are some issues that have to be taken into account when applying face recognition in security and surveillance applications. The reliability of the technology is one issue. The technological challenges related to face analysis were discussed in Chapter 2. The recognition task is much easier if there is only one person or a small group that need to be recognized. For example, a laptop might have only one user or a few users, making the task easier, while an online shopping system might have thousands or more users, making it very difficult to authenticate users reliably. It might also be impractical to store and access the large amount of face data needed. Issues such as what if somebody tries to use a photo of the person for authentication have to be considered. One option that can be used in some situations is to use a thermal camera in addition to a normal camera because a photo has a rather constant temperature while different parts of the face have different temperatures. In some countries laws on privacy may also disallow surveying people with video camera, storing the video data, and using the video data with other personal

information without the knowledge or prior approval of the person observed.

Information search and data mining is an area where face analysis is already in use. Google included an option to search for images containing faces to their image search in May 2007. Microsoft followed in July and included face and portrait filtering in their "Live Search Images" engine. Exalead engine also allows searching images with faces.

Examples of the searches using the above three search engines are shown in Figure 7.1 and in Figure 7.2. As can be seen, the search results also include non-human faces and images without faces, and the results depend somewhat on the search engine used.

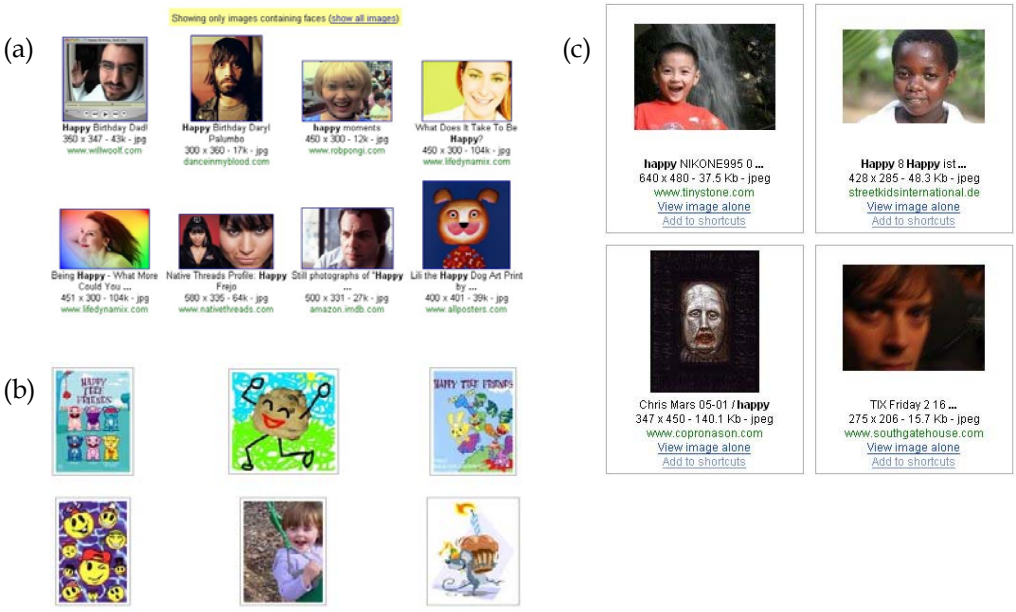


Figure 7.1. First face image search results using the word "happy" with the (a) Google, (b) Microsoft Live, and (c) Exalead search engines (search carried out on 16th August, 2007).

One could claim that it is perfectly fine to include non-human faces in the search results but it would probably also be useful to be able to search only human faces because users are probably sometimes interested only in the human faces. The best situation would be if the user could specify exactly what kind of faces to search, for example, to search only smiling faces of Martin Luther King.

As Figure 7.2 shows, the detection reliability could be improved too. When a user tries to find faces with the search word "map" the results contain hardly any faces. The maps contain many details and face detection algorithms may find face-like areas in the images because of that.

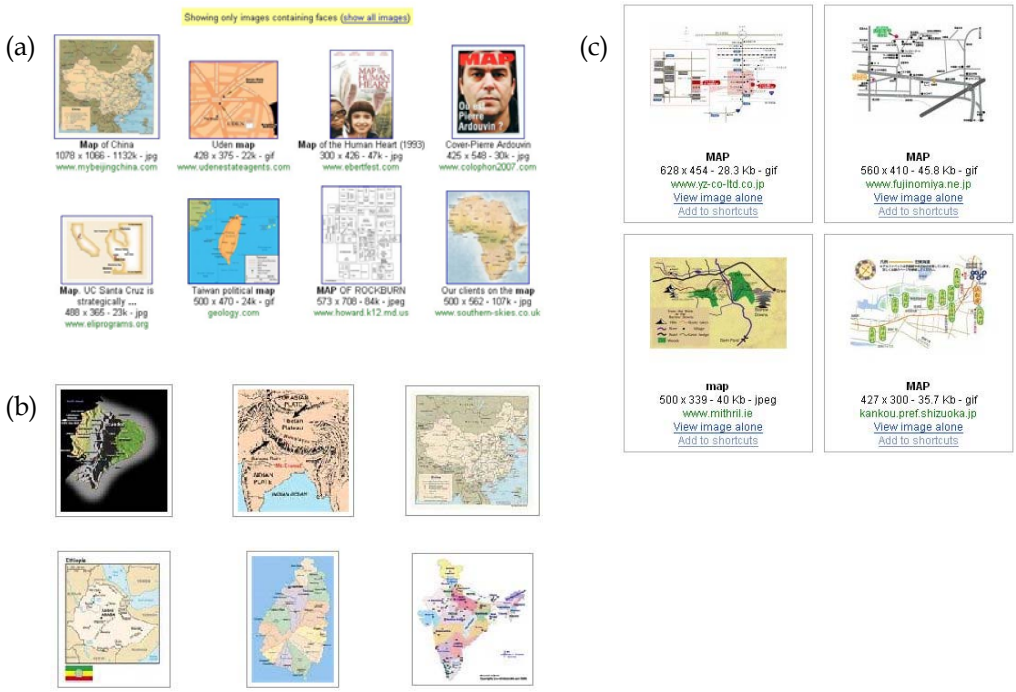


Figure 7.2. First face image search results using the word "map" with the (a) Google, (b) Microsoft Live, and (c) Exalead search engines (search carried out on the 16th of August, 2007).

Besides the search engines there are many other types of applications where it would be useful to be able to search faces. For example, if we have a large collection of photos on our computer that we have taken over several years and we would like to find photos that contain a specific person this would be possible with software that does face detection and recognition.

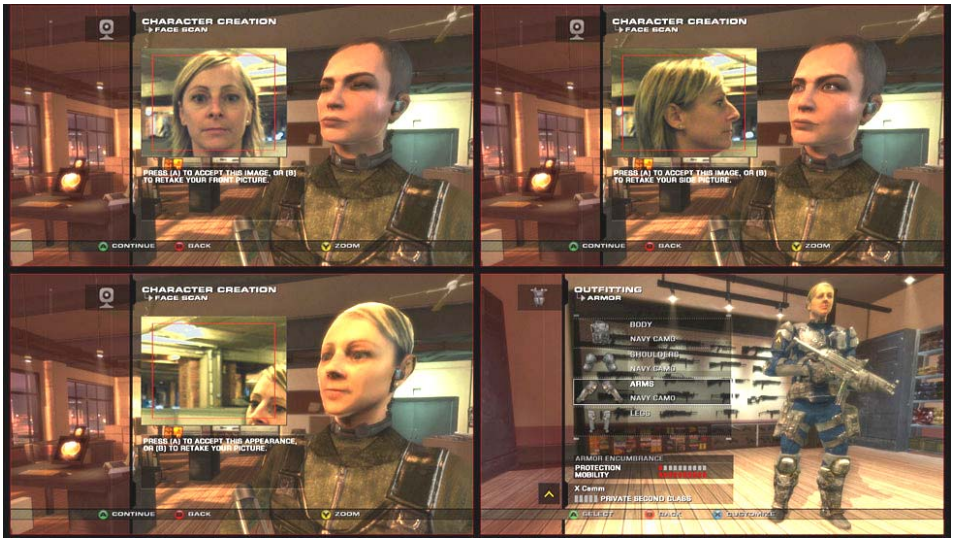


Figure 7.3. Steps to create a game character with a player's face in the "Rainbow Six Vegas" game. (Image from http://ve3d.ign.com/images/fullsize/3946/Other/General, IGN Entertainment, Inc.)

Face analysis has a lot of potential in computer and console games. Probably the simplest use is to include a user's face (or the whole body of the user) in the game character. For example, the Xbox 360 game "Rainbow Six Vegas" has this option. The face texture is captured from frontal and side views as shown in Figure 7.3.

The more complex examples include controlling a game with face or head movements. Gorodnichy and Roth (2004) presented two such games: BubbleFrenzy and NousePong. The head movements were registered by tracking the player's nose. In the BubbleFrenzy game the purpose was to drop bubbles of the same colors by shooting them with bubbles. The user turned a bubble turret for a desired direction by rotating the head to the left or to the right. The controlling was implemented so that even small head rotations were sufficient to turn the turret over the whole 180° range. The users preferred to control the turret by head rotation instead of the mouse. Pressing the spacebar launched a bubble from the turret but it would be possible to use, for example, eye blinking as Gorodnichy and Roth did in a painting application also presented in the article. In the NousePong game two players played against each other. There was a ball bouncing over the table and the players tried to bounce the ball back to the other player by moving a club horizontally with their head movements.

Face detection has also been used in digital cameras. Many companies including Canon, Fujifilm, Panasonic, Pentax, and Sony have included face detection in some of their camera models. This feature makes it possible to adjust camera parameters such as focus, white balance, and flash level automatically and optimally for the faces. The inclusion of the face detection technology in the cameras means that the detection has to be reliable. Otherwise users would just get frustrated while using the feature.

With cameras it is also possible to use skin color for face detection (non-painted faces). This way it is possible to remove image regions that cannot contain faces and improve detection speed and accuracy. The color model should be such that all the skin colored regions are found and the actual decision whether the regions contain faces or not is made by the further analysis.

Other application areas for face analysis include drowsiness detection in cars, user tests, and arts. Ayoob et al. (2003) implemented drowsiness detection by observing the driver's eyes and the driver was alerted if the eyes were closed often and for many seconds at the time.

Facial expression analysis can be used in user tests for observing user emotions. Noldus provides the FaceReader™ tool (FaceReader, 2007) that classifies facial expressions and other facial attributes and, they mention

usability testing and market research among other application areas for the FaceReader™.

Figure 7.4. Video installation at the Art Center Pasadena that makes an art of automatic expression classification (the image captured from the video shown at the page http://www.christian-moeller.com/display.php?project_id=36).

Finally, the Cheese (Cheese, 2003) was a video installation at the Art Center Pasadena developed by Caltech and the Machine Perception Laboratories of the University of California, San Diego. The installation included videos taken of the actresses who tried to hold a smile as long as possible. An example photo of the installation is shown in Figure 7.4. The facial expression classification system determined if the smile was real enough and a signal was played when the smile dropped below a specified threshold.

7.2.2 Applications with Face Analysis and Other Perceptual Technologies

Although face analysis can be applied alone in many applications, even more applications become available when it is used with other perceptual technologies such as speech recognition and haptic feedback. Next some example applications that take advantage of face analysis together with one or more perceptual technologies are discussed.

Person identification based on multiple input channels is a topic that has been under intensive research. As an example, Brunelli and Falagvina (1995), Chibelushi et al. (1997), and Faraj and Bigun (2007) used speech along with facial cues for person identification. Ali et al. (2006) integrated face and fingerprint biometrics. In all the studies the identification accuracy was improved by combining the input channels together.

Multiple modalities and input channels have been used with interactive agents. Darrell et al. (2002) created an agent dialog prototype that listened to user's speech commands while the user was facing the agent. The agent indicated whether or not it was listening to the user by a corresponding facial expression. An interaction situation with the agent is shown in Figure 7.5.

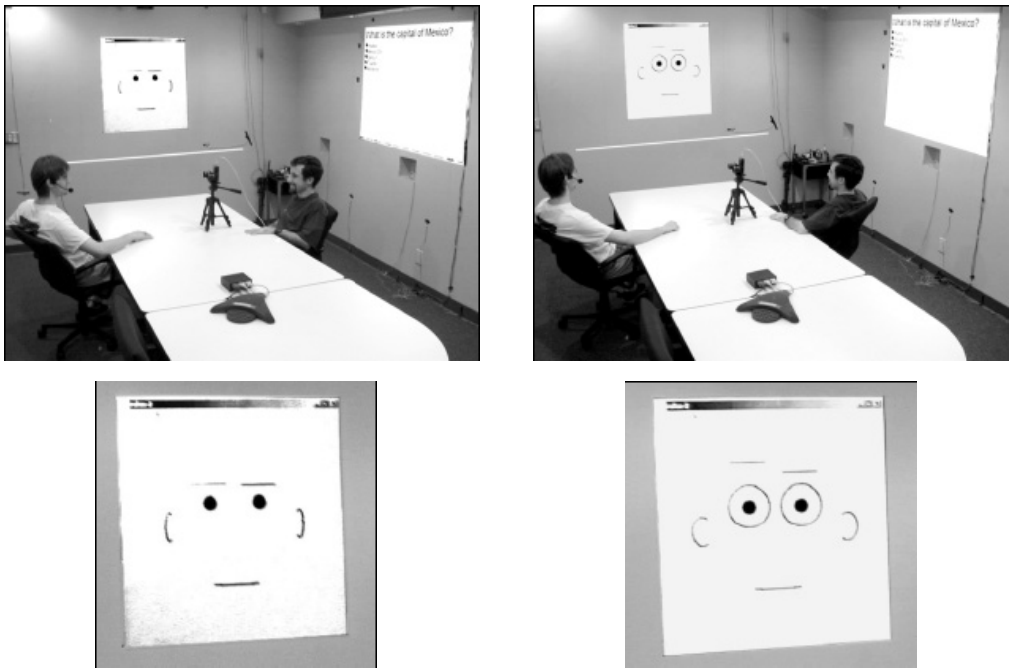


Figure 7.5. Interactive agent that listens to speech commands when the user is facing it. On the left the people are talking to each other and the agent is inactive, while on the right the users are facing the agent and it is listening to speech commands. (Darrell et al., 2002).

Bevacqua et al. (2006) proposed an interactive agent that would recognize a user's facial expressions, head movements, and hand gestures and would act according to the interpretation and behavior model of the agent.

Maybe a little surprisingly, although there are computer vision games with different perceptual technologies, not many computer games making use of both face analysis and other perceptual technologies have been published. The first computer vision games aimed at consumers (Höysniemi, 2006) came with the Intel® Play™ Me2Cam Virtual Game System. The players interacted in the world by touching virtual objects with their hands and head (D'Hooge and Goldsmith, 2001). The system used background subtraction to determine a player's location, determined head location and hand locations with a heuristic model, and measured the amount of motion while tracking the player. Besides Me2Cam, computer games that include computer vision, such as QuiQui's Giant Bounce (Hämäläinen and Höysniemi, 2002) that also includes sound input, Kick Ass Kung-Fu (Hämäläinen et al., 2005), EyeToy Games (EyeToy, 2005), and Xbox LIVE Vision Games (Xbox LIVE Vision, 2006), have been created. Some of them use face analysis and some of them use other computer vision technologies, but none use both.

Drowsiness detection was mentioned in the previous subsection. A related system has been included in some Toyota car models. The Pre-Crash Safety system detects the direction of the driver's face and warns the driver if there is a danger of collision when the driver is not facing

forward (Toyota, 2007). The warning includes both audible and visible warning as well as car braking.

7.3 An Example Application: Information Kiosk with an Interactive Agent

Information kiosks and other types of kiosks, such as entertainment kiosks, advertisement, and service kiosks (Borchers et al., 1995) are fairly common nowadays. For example, train tickets can often be bought from kiosks and many airplanes have entertainment kiosks that can be used in flight by passengers. Since the kiosks can usually be used by anyone, also by people with no knowledge of computers, it is important that kiosks are as easy to use as possible. The other properties that are important depend somewhat on the kiosk type. For example, advertisement kiosks should attract users to use them and entertainment kiosks should naturally also be entertaining.

Multimodal interaction is one possible way to facilitate and improve kiosk usage. Now a more profound look is taken at a multimodal information kiosk developed at the University of Tampere by the Multimodal Interaction Group (Mäkinen et al., 2002). Christian and Avery (1998, 2000) have described studies with similar kiosks.

7.3.1 Overview of the Kiosk

The kiosk provided information on the museums in the city of Tampere, Finland. The kiosk, including a touch screen, a web camera, and speakers is shown in Figure 7.6 and a closer view of the interface is given in Figure 7.7.

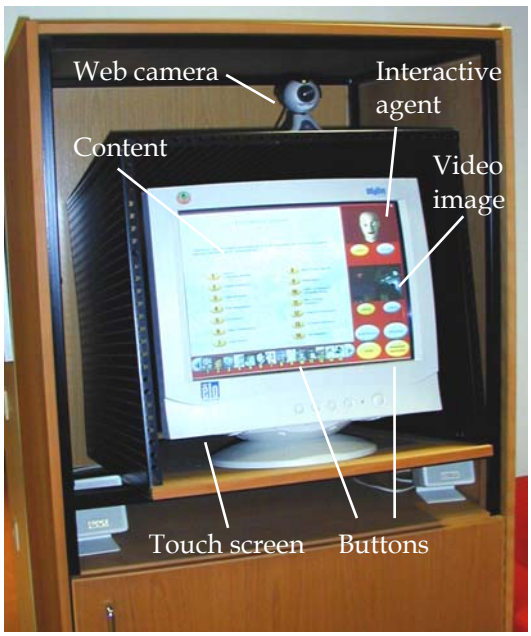


Figure 7.6. Multimodal kiosk providing information on the museums in Tampere.

The content of the kiosk was in Finnish. The user navigated through the content by touching the buttons on the screen. In addition to the content, which occupied most of the space there was an interactive agent that spoke Finnish at the top right corner of the display, and the video image captured by the web camera was shown below the agent. The agent was developed by Olives et al. (1999). The agent could turn in different directions and could show different facial expressions. The user could put the agent to sleep and wake it up at any time. The user could also hide the video image.

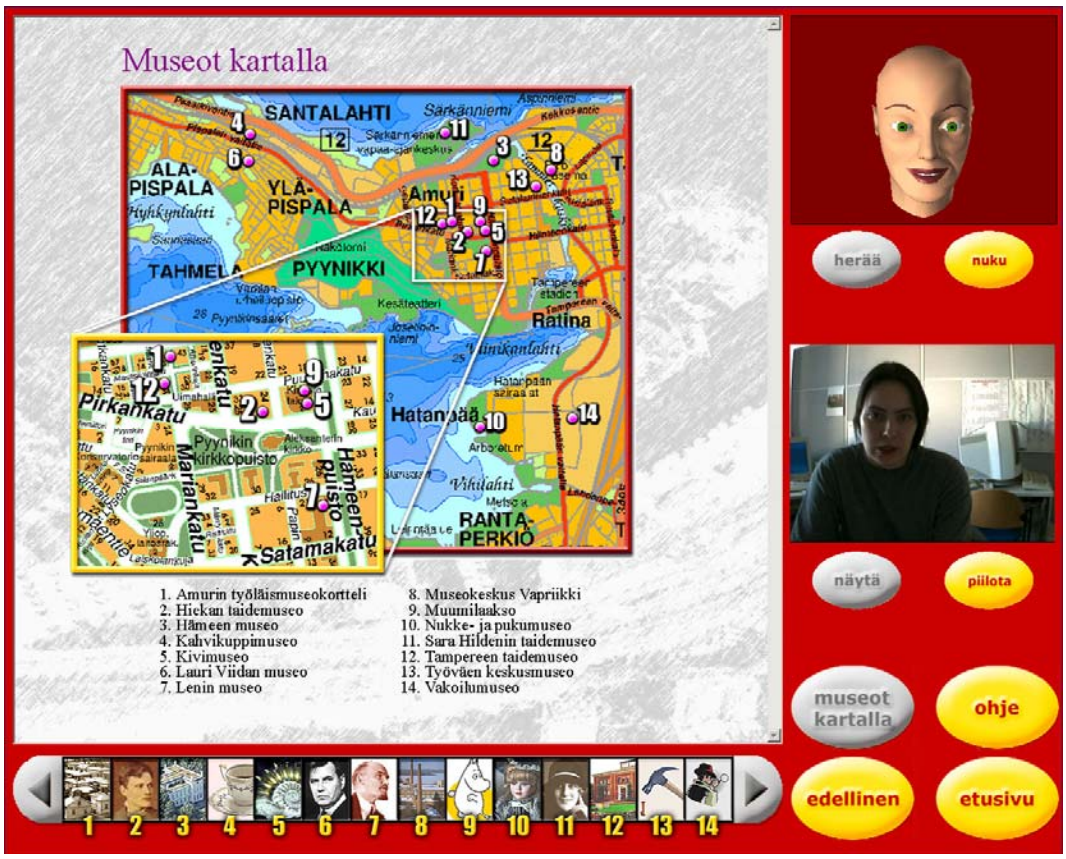


Figure 7.7. User interface of the kiosk.

7.3.2 Face Analysis Component

The actions that the interactive agent performed were possible using face detection. The blob face detector described in Chapter 3 was used for this purpose.

When a face was detected in several video frames the user was greeted. The location of the face in the video image determined the direction in which the interface agent turned. In addition, the agent provided help for the user. This happened if the user's face was continuously detected (there could be a few frames without detection) but for a while the user did not press any buttons on the display. When the user left the kiosk and the face could not be detected the agent said goodbye to the user.

7.3.3 Experiments with the Kiosk

An experiment with the kiosk was carried out and the results of interest from the viewpoint of the thesis are described here. A more thorough analysis of the whole experiment will be found in our article (Mäkinen et al., 2002). There were 20 test participants aged between 11 and 64 years. The participants were given various tasks. Most answers to the tasks could be found in the textual content but some of the answers were given by the interactive agent. After the participant had completed all the tasks he or she was given a questionnaire. There was also an interview at the end of the test.

The majority of the participants reported that the appearance and the voice of the interface agent was neither displeasing nor pleasing. In addition, the majority of the participants did not notice that the agent had various expressions nor did they notice that the agent turned in their direction. Some of the participants mentioned that it was hard to concentrate on using the kiosk while the agent was talking and the opinions of the information that the agent gave were rather diverse.

At the time when the experiments were carried out the computer vision based help system had not yet been implemented. Therefore, the agent greeted and said goodbye to the participants during the tests but all the other comments were triggered when the participant navigated to a certain page.

The face detection worked without problems with six participants but with the rest of the participants there were between 3 and 46 greetings and farewells. The users could turn and move their heads and as it was found in the face detection experiment reported in Section 3.3 the face detection did not always work well in these situations. Furthermore, sometimes participants moved their faces out of the web camera view.

In general, the participants thought that the agent was well suited for kiosk applications and also for other types of applications, although the users did not always listen to what the agent said. In addition, although a few participants were irritated by the repeated greetings and farewells during use some of the participants responded to the agent when it said good bye to them at the end of the test. Nevertheless, for an agent to be more useful and user friendly it should adapt to the users and to the usage situations.

7.3.4 Discussion

There are many options to improve the interface agent. All the perceptual intelligence of the agent was based solely on face detection. With facial expression classification the agent could make assumptions about the usage situation. For example, the agent could respond with a smile when the user is smiling and help could be provided when the user looks angry.

Body posture recognition could be used for the same purpose. If the age group of the user were recognized the agent could change its speaking style. For example, it could use less formal language when speaking to children. With gender classification the agent could greet the user stylishly, for example by saying "good day, sir" or "good day, madam" and it could also talk about different things.

Speech recognition would also be useful for the agent. The agent could be conversational, so that a user could talk to the agent. There are also situations when there is more than one user at the kiosk. In these situations users may speak to each other and if the agent can perceive this then it can be silent during the conversation. Gaze tracking would be useful, too. It could provide the user's focus of attention and the agent could decide, for example, when to speak (Christian and Avery, 2000) or based on this knowledge what kind help to provide for the user.

Our kiosk studies and many other studies (Christian and Avery, 1998, 2000; Pentland, 2000; Turk and Kölsch, 2004) show that when the perceptual technologies become more robust and when they are combined it is possible to make the interaction with computers more pleasing and natural.

7.4 New Applications for Face Analysis

There are two ongoing constructive research projects that are applying our results on automatic face analysis: a learning environment with an attentive agent and a demographic data collection application.

7.4.1 A Learning Environment with an Attentive Agent

The learning environment is targeted at young and adult students both in individual and collaborative learning. The environment is being developed on the AtGentive project (AtGentive, 2007). It makes use of the agents that command the attention of the students. The agent should be able to perceive users and their actions and do reasoning to be able to affect users' attention and support the users in their learning.

The system has a component that perceives user's attention. Keyboard and mouse events provide information on the user's actions but they cannot be used to determine whether the user is sitting next to the computer and is inactive, for example, due to thinking or if the user has left the computer. To be able to perceive the presence of the user a face detection component will be included to the learning environment. With the component it is also possible to determine how many users there are at the computer and use this information in the reasoning.

The learning environment could include other face analysis technologies, too. For example, a student's face could be recognized when he or she

starts a learning application and personal settings could be loaded. During a learning session a student's expressions could be observed and difficulty or, for example, the contents of the tasks could be modified accordingly. If eye tracking was a part of the system then it would be possible to gain knowledge of the student's focus of attention. The eye tracking information could be combined with the expression information.

7.4.2 Demographic Data Collection

Automatic face analysis is to be used in a product developed by Audio Riders Oy (Audio Riders, 2007) that can be used for providing tailored sound milieu in shops and department stores. The milieu can include various types of audio, for example music, advertisements, and announcements played for customers. The product includes a small loudspeaker with an integrated Linux server (see Figure 7.8) and it is currently used by many companies such as IKEA and several shopping centers in Finland.

Figure 7.8. Linux server integrated on a small size circuit board.

Face analysis, applying the results of this thesis, is planned to be used for demographic data collection in the product. The audio content played could be selected according to the number of customers in the store, their genders, and ages. The volume level is automatically adjusted based on a sound pressure technology that takes into account the audio content played, the ambient noise, and the noise level variations. The number of faces that have been detected and their distances from the camera could also be used for the volume adjustments.

The demographic data can also be used for analyzing customer behavior in a shop or, for example, in a museum. The number of people that visit the shop during the day and their gender and age distributions could be interesting infromation, likewise, how much time they spend in different parts of the store. An art museum owner might be interested to know how long visitors watch each painting in the museum.

7.5 IDEAS FOR FUTURE WORK

As the perceptual technologies become more reliable and hardware improves, there will be more and more applications with perceptual capabilities. Next some ideas of possible applications will be given showing the potential of automatic face analysis and the potential that different modalities form together.

7.5.1 Home and Office Applications

Computer games could benefit much from more elaborate analysis of players' behavior. A massive multiplayer online role-playing game like the well known World of Warcraft (WoW), which is played by millions of people around the world is a good example. A screenshot of the WoW is shown in Figure 7.9. In the WoW each player has a game character that he or she controls in the fantasy world and carries out missions that vary from collecting ingredients for a cook so that he or she can make a delicious soup to slaying a dragon that threatens the world. An important part of the game consists of interaction between the players through their game characters. Game characters can talk to each other; they can fight against each other or try to defeat a common enemy, or they can even trade items and other things.

Figure 7.9. Screenshot from the World of Warcraft game (from the website: http://www.blizzard.com/).

Computer vision could be used to enhance the behavior of the character. Facial expressions of the player could be analyzed and copied to the character so that other players could see the mood of the other players through their characters. Also, interaction with a non-player character (NPC, a game character that the computer controls) could be affected. The

NPC could act differently depending on the expressions of the game character. Hand gesture recognition could be used, too. For example, many game characters can cast spells. A certain hand gesture pattern could be required to cast a certain spell and the effect of the spell could also depend of the quality of the gesture. This feature could cause funny effects if the player had not practiced the pattern well enough. One could also include speech recognition in the spell casting and if the player was wearing gloves that give tactile feedback varying effects could be produced.

The same techniques can be applied to remote presence applications and to virtual interaction in general. The people can be detected and their identities, facial expressions, and gestures can be recognized and this information can be used at the remote location or in the virtual world. For example, in the case of the remote video meeting it could be useful to show only the faces of the remote participants to save desktop space for the other applications or to make the facial expressions of the participants more visible.

The active badge location system (Want et al., 1992) was developed to provide the locations of the people in an office environment. The badge provided the location of each individual in the office unless they wanted privacy and switched the badge off. One could replace or enhance the system by using face analysis. Face detection and recognition could be used for determining locations similarly to the original system. However, cameras would be placed around the office environment instead of a large number of badges worn by the personnel.

As with the original system, privacy would have to be taken into account. An individual should have the option to switch the tracking off at any time. One possibility would be to show a hand gesture such as the stop hand signal to a camera. The person could allow tracking, for example, by the thumbs up gesture.

One could easily add information on the visitor to the system. The visitor could face a camera and write his or her name and other necessary information using an information booth. Speech recognition could be used instead of writing. Information could also be provided on the people whose data were missing from the system. For example, one could check that his co-worker is talking with three strangers in a lobby (again speech recognition would be needed as well as the facing directions and fairly exact locations of the persons). It would also be possible to provide more information such as the genders of the strangers or even show their faces to the interested party, but again privacy would need to be taken into account.

Perceptual technologies can be of great benefit to disabled people. For example, a person whose hands and legs are paralyzed can still communicate with his or her face. If the computer can recognize speech, track user's gaze and recognize user's facial gestures, the computer can be used by a paralyzed person. There is already research on using eye tracking alone and with facial gestures for interaction. For example, Hyrskykari (2006) presented an application where gaze was used as a reading aid. When a user read text, for example in a foreign language, and stopped to some word an explanation of the word was given.

Surakka et al., (2004) studied how gazing and frowning could be used together for pointing and selecting objects shown on a screen. The frowning was registered with electrodes attached above the user's *corrugator supercilii* muscle. However, for commercial applications computer vision-based frowning recognition could be a more user-friendly option.

7.5.2 Mobile Applications

A large number of mobile phones and multimedia devices have an integrated camera and some models have two cameras, so computer vision can and has been used in mobile devices. The Upcode application (Upcode, 2007) and the Omron face recognition software (Omron, 2005) were mentioned earlier in this thesis. Here future applications are considered.

Currently, the mobile phone cameras can usually capture still images of several megapixels (for example, 3.1 megapixels that equals to 2048*1536 pixels) although the video image resolution could be as low as 176*144 or 176*120 (QCIF, Quarter Common Intermediate Format). Especially the still image resolution is good enough for many computer vision applications. The processing performance of mobile devices is the biggest restricting factor for the computer vision applications at the moment. However, the processors are constantly becoming faster and more memory is included in mobile devices. Some processing can also be distributed through the wireless connection to efficient servers. This probably means that mobile devices will have various computer vision applications in the future.

People usually carry mobile devices with them so, for example, when they get to know new people at a meeting or see someone on the street they probably have the device available. A simple application for automatic face analysis would be when storing a person's contact information on the phone. One could take a photo of the person and the face would be detected from the image. The face image would then be stored along with the contact information on the phone.

For a visually impaired person a mobile phone with automatic face analysis could be of great help. The person could attach the phone to his

or her backpack strap next to the chest. The accessibility application would then analyze the video image captured with the mobile phone camera and notify the user of the interesting events. These events could include a familiar person walking nearby, a group of people standing in the street, or a person looking towards the user. The user could have earphones attached to the mobile phone and the notifications could happen by audio feedback. For example, a voice could say that "there are five about 30 year old females standing ahead". Haptic feedback could be preferred to audio with certain events and situations such as when discussing with another person. In such a situation, for example, the results of the automatic facial expression analysis could be converted into haptic feedback.

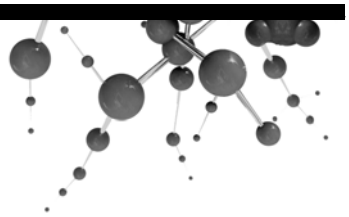
Also, for people who suffer from *prosopagnosia* (face blindness, the inability to recognize faces) the mobile face detection and recognition would be useful. They could use mobile phone similarly to the visually impaired people.

7.6 SUMMARY

In this chapter various examples of existing and future applications including face analysis and other perceptual technologies were given. The kiosk application developed at the University of Tampere was described in more detail.

Many of the novel applications for automatic face analysis require only that existing techniques are taken into use in the applications. For example, in usability testing, face detection and facial expression analysis can be performed automatically and logged with the other logged data. However, there are also more complex applications that may include many modalities. While applications of face analysis already exist in fields such as games and security they are just a fraction of the applications that will be created in the future.

The applications described showed the potential of face analysis, especially when applied together with other perceptual technologies such as other computer vision technologies, auditory input and output, eye tracking, and haptics. Probably the most successful future applications will be multimodal.



8 Conclusions

Automatic face analysis is a field that uses knowledge from many other fields such as pattern recognition, machine learning, signal processing, psychology, and neurology. Knowledge of how mammals and humans see has brought ideas to computational algorithms. For example, Gabor wavelets and neural networks, which are widely used in automatic face analysis, have natural origins. They are based on ideas that work well in humans and other mammals. On the other hand, many of the algorithms originate from technical fields because automatic face analysis is ultimately about signal processing, learning, and pattern recognition. One could say that automatic face analysis is about fitting the algorithms of nature and of computers together.

The creation of face analysis techniques and algorithms means that one must consider the reliability and speed of the algorithms. One is often gained at the expense of another. For example, algorithms that create a 3D model of the face are usually more reliable than those that use 2D models but also require more processing power, which makes them slower. In practice, the 3D algorithms are currently too slow to be used in most of the applications in the HCI field. This will change in the future when computers become faster.

In this dissertation the focus was on frontal 2D face analysis. Work on face analysis and on gender classification was reviewed. Most of these studies have introduced new algorithms that aim to improve classification reliability. However, studies where algorithms are compared thoroughly with various face datasets and with automatic face detection and alignment are rare. In this dissertation various gender classification algorithms were studied and compared. Some of the gender classification algorithms were novel. The algorithms were also combined with face

detection and alignment. The two face detectors used in the experiments were the novel blob face detector that was introduced in Chapter 3 and the cascade detector introduced by Viola and Jones (2001).

Tools are needed to improve the existing techniques and to experiment with novel ones. Various tools were implemented and used to carry out the experiments that were described. Also, a parallel training algorithm for discrete Adaboost was created because Adaboost training with a large dataset is very time consuming.

The blob face detector proved to be a fast and reliable detector for frontal faces when color images are available and the skin color model is good enough. However, the locations of the faces detected were not very accurate in the experiments. Better gender classification accuracies were achieved when the cascaded face detector was used because faces were located more accurately with it.

Experiments where no alignment, manual alignment, and several automatic alignment methods were used between face detection and gender classification showed that automatic alignment would increase classification accuracies if working properly. This is supported by the fact that manual alignment increased the classification accuracies compared to the no alignment condition. However, the automatic alignment methods that were used in the experiments decreased the classification accuracies. To improve the methods, one could select a larger set of face images to create a face model, vary the parameters of the methods, or select a completely different alignment method.

When faces are detected automatically there may be horizontal and vertical inaccuracies in the face location, the face may be rotated, or the area that is detected as a face may be larger or smaller than the actual face. An experiment where these parameters were varied was carried out. The Adaboost gender classifier with Haar-like features proved to be more reliable for rotations than the other classifiers.

Besides alignment, various other issues were considered. An SVM with pixel-based input proved to be the most reliable classifier when the input data was high quality, which is in line with many other studies (BenAbdelkader and Griffin, 2005; Castrillón-Santana et al., 2003; Moghaddam and Yang, 2002; Yang et al., 2006b). Nevertheless, the experiments indicate that the features used for the gender classification may be more important than the machine learning method. Face image size used for the gender classification was not an important factor for classification accuracy.

The best gender classification rates that were achieved with frontal faces were slightly over 90% for high quality images. However, only images of

web camera quality can be assumed in many applications and with these images the highest classification rates were slightly over 80%.

Inputs to the classifiers were either histogram equalized image pixels or based on Haar-like features, or on LBP. Pixels were used as input to the neural network and to the SVM, Haar-like features were used as input to the Adaboost classifiers, and LBP features were used as input to the SVM classifier. The results of the experiments indicated that features are more important to classification accuracy than the classifier. For example, in the results described in Section 4.4 classifiers were compared using manually aligned face images with or without hair. The classifiers that used other than pixel-based input benefitted from the inclusion of hair in the face images while the classifiers that used pixel-based input benefitted from the exclusion of hair from the face images. This issue could have been studied further using each feature type with each classifier but this issue was left for future work.

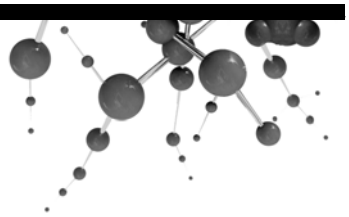
Furthermore, it is possible that higher gender classification accuracies would have been achieved if, for example, the face images had been filtered with Gabor wavelets or if independent component analysis (ICA) had been used. Good results have been achieved with Gabor wavelets in face recognition (Shen and Bai, 2006). However, image pixels are readily available to be used as input after histogram equalization and no further calculations are needed, and values for Haar-like features and for LBP features are fast to calculate.

One could also consider classify gender on the basis of biometric features. For example, distances between detected facial features such as eyes, nose, and mouth could be used for classifying gender because there are differences in face shape and in facial feature locations between genders. The problem with these features is that besides gender, facial expressions, ethnicity, and age also affect face shape and the relations between facial features.

In many HCI applications it is important for the user to receive real-time feedback. Therefore the face analysis methods that are used in the applications should be fast. The face detection methods and method combinations used in the experiments meet these requirements. As computers become faster, methods and features requiring a lot of processing power can be used and better face detection and classification accuracies can be achieved. However, mobile devices have much less processing power than PCs and increasing numbers of applications are being developed for them. As many mobile devices have cameras face analysis can be used in these applications, but, again, face analysis methods have to be fast enough. Ultimately, it is likely that there will always be applications where the speed of the methods used is important.

The results in Chapters 3, 4, and 5 are interesting from the HCI perspective. The novel face detection method presented in Chapter 3 is fast enough to be used in applications that run on a standard PC. When face detection is used in perceptual applications the faces detected may have variations in rotation, translation, and scale, and it is useful to know how these misalignments affect classification accuracies. The results in Chapter 4 address this issue. The face detectors and gender classifiers used in the experiments are usable with frontal faces, but for arbitrarily rotated faces, which are common in real applications, the methods cannot be used as such. One could use, for example, the rotation invariant multi-view face detector proposed by Huang et al. (2007) to detect arbitrarily rotated faces and train separate gender classifiers for each rotation case. The face analysis process should be fully automatic in many applications and all the results in Chapter 5 are for combined systems that are fully automatic. The classification accuracy achieved with web camera images is interesting because they are likely to be used in many home applications. Web cameras are inexpensive and many people have them. Gender classification accuracy was measured for web camera images in some of the experiments in Chapter 5.

Face analysis techniques have a wide field of application from computer games to learning applications. Existing applications were considered and ideas for future applications were presented. Recently face detection has been included in many digital cameras and search engines. This indicates that face analysis techniques are maturing and in the near future more applications will most probably be seen. However, there is still a lot of research to be done to create methods that work well in all kinds of conditions: indoors, outdoors, with partially occluded faces, profiles, and so on. There is also a lot of research to be done to find out how to make the best use of the face analysis techniques in the broad range of applications in the HCI field.



9 References

Abdi, H., Valentin, D., Edelman, B. & O'Toole, A. J. (1995), More about the difference between men and women: Evidence from linear neural network and principal component approach, *Neural Computation* **7**(6), 1160-1164.

Aggarwal, J. K. & Cai, Q. (1999), Human motion analysis: A review, *Computer Vision and Image Understanding* **73**(3), 428-440.

Aleksic, P. S. & Katsaggelos, A. K. (2006), Automatic facial expression recognition using facial animation parameters and multistream HMMs, *IEEE Transactions on Information Forensics and Security* **1**(1), 3-11.

Ali, I., Ali, U., Shahzad, M. & Malik, A. (2006), Face and fingerprint biometrics integration model for person identification using Gabor filter, *in IEEE International Conference on Computer Systems and Applications*, pp. 140-143.

Arulampalam, M., Maskell, S., Gordon, N. & Clapp, T. (2002), A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking, *IEEE Transactions on Signal Processing* **50**(2), 174-188.

Ascension (2007), ReActor2, http://www.ascension-tech.com/applications/animation.php.

AtGentive (2007), AtGentive: attentive agents for collaborative learners, http://www.atgentive.com/.

Audio Riders (2007), Audio Riders Oy, http://www.audioriders.fi/cgi/index.cgi?lang=eng&s=company.

Ayoob, E. M., Grace, R. & Steinfeld, A. (2003), A user-centered drowsy-driver detection and warning system, *in Proceedings of the Conference on Designing for User Experiences* (DUX'03), ACM Press, New York, NY, USA, pp. 1-4.

Baluja, S. & Rowley, H. A. (2007), Boosting sex identification performance, *International Journal of Computer Vision* **71**(1), 111-119.

Baudouin, J. & Humphreys, G. W. (2006), Configural information in gender categorisation, *Perception* **35**, 531-540.

BenAbdelkader, C. & Griffin, P. (2005), A local region-based approach to gender classification from face images, *in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (CVPR'05) - Workshops, IEEE Computer Society, Washington, DC, USA.

Bevacqua, E., Raouzaiou, A., Peters, C., Caridakis, G., Karpouzis, K., Pelachaud, C. & Mancini, M. (2006), Multimodal sensing, interpretation and copying of movements by a virtual agent, *in Proceedings of Perception and Interactive Technologies* (PIT'06).

Beymer, D. & Flickner, M. (2003), Eye gaze tracking using an active stereo head, *in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (CVPR'03).

Blackburn, D., Bone, J. & Phillips, P. (2001), FRVT 2000 evaluation report, *Technical Report*, National Institute of Justice, USA.

Borchers, J., Deussen, O. & Knörzer, C. (1995), Getting it across: Layout issues for kiosk systems, *SIGCHI Bulletin* **27**(4), 68-74.

Boser, B. E., Guyon, I. M. & Vapnik, V. N. (1992), A training algorithm for optimal margin classifiers, *in Proceedings of the 5th Annual Workshop on Computational Learning Theory* (COLT'92), ACM Press, New York, NY, USA, pp. 144-152.

Bradski, G. (1998), Real time face and object tracking as a component of a perceptual user interface, *in Proceedings of the 4th IEEE Workshop on Applications of Computer Vision* (WACV'98), pp. 214-219.

Brewster, S. A., Wright, P. C. & Edwards, A. D. N. (1993), An evaluation of earcons for use in auditory human-computer interfaces, *in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI'93), ACM Press, New York, NY, USA, pp. 222-227.

Bruce, V., Burton, A. M., Hanna, E., Healey, P., Mason, O., Coombes, A., Fright, R. & Linney, A. (1993), Sex discrimination: how do we tell the difference between male and female faces?, *Perception* **22**(2), 131-152.

Brunelli, R. & Falagvina, D. (1995), Person identification using multiple cues, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **17**(10), 955-966.

Brunelli, R. & Poggio, T. (1995), HyberBF Networks for gender classification, *in Proceedings of the DARPA Image Understanding Workshop*, pp. 311-314.

Buchala, S., Davey, N., Frank, R. J. & Gale, T. M. (2004), Dimensionality reduction of face images for gender classification, *in Proceedings of the 2nd International IEEE Conference on Intelligent Systems*, pp. 88-93.

Buchala, S., Gale, T. M., Davey, N., Frank, R. J. & Foley, K. (2005), Global and feature based gender classification of faces: A comparison of human performance and computational models, *Progress in Neural Processing* **16**, 349-360.

Buddharaju, P., Pavlidis, I., Tsiamyrtzis, P. & Bazakos, M. (2007), Physiology-based face recognition in the thermal infrared spectrum, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **29**(4), 613-626.

Burton, A. M., Bruce, V. & Dench, N. (1993), What's the difference between men and women? Evidence from facial measurement, *Perception* **22**, 153-176.

Castrillón-Santana, M., Déniz-Suárez, O., Guerra-Artal, C. & Hernández-Tejera, M. (2005), Real-time detection of faces in video streams, *in Proceedings of the 2nd Canadian Conference on Computer and Robot Vision* (CRV'05), pp. 298-305.

Castrillón-Santana, M., Déniz-Suárez, O., Hernández-Sosa, J. & Domínguez-Brito, A. (2003), Identity and gender recognition using the ENCARA real-time face detector, *in Conferencia de la Asociación Española para la Inteligencia Artificial* (CAEPIA'03).

Castrillón-Santana, M., Déniz-Suárez, O., Lorenzo-Navarro, J. & Hernández-Tejera, M. (2006), Gender and identity classification for a naive and evolving system, *in 2nd Workshop on Multimodal User Authentication* (MMUA'06).

Castrillón-Santana, M. F. (2003), On real-time face detection in video streams. An opportunistic approach., *PhD Thesis*, Universidad de Las Palmas de Gran Canaria.

Chang, C. & Lin, C. (2001), LIBSVM: A library for support vector machines.

Cheese (2003), Cheese video installation, http://www.christian-moeller.com/display.php?project_id=36.

Chellappa, R., Wilson, C. & Sirohey, S. (1995), Human and machine recognition of faces: A survey, *Proceedings of the IEEE* **83**(5), 705-741.

Cheng, Y. D., O'Toole, A. J. & Abdi, H. (2001), Classifying adult's and children's faces by sex: Computational investigations of subcategorical feature encoding, *Cognitive Science* **25**, 819-838.

Chibelushi, C., Mason, J. & Deravi, N. (1997), Integrated person identification using voice and facial features, *in IEE Colloquium on Image Processing for Security Applications* (Digest No: 1997/074), pp. 4/1-4/5.

Choi, S., Kim, C. & Choi, C. (2007), Shadow compensation in 2D images for face recognition, *Pattern Recognition* **40**(7), 2118-2125.

Christian, A. D. & Avery, B. L. (2000), Speak out and annoy someone: Experience with intelligent kiosks, *in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI'00), ACM Press, New York, NY, USA, pp. 313-320.

Christian, A. D. & Avery, B. L. (1998), Digital smart kiosk project, *in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI'98), ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, pp. 155-162.

Clutterbuck, R. & Johnston, R. A. (2004), Demonstrating the acquired familiarity of faces by using a gender-decision task, *Perception* **33**(2), 159-168.

Colmenarez, A., Frey, B. & Huang, T. S. (1999), Detection and tracking of faces and facial features, *in Proceedings of the International Conference on Image Processing* (ICIP'99), pp. 651-661.

Comaniciu, D., Ramesh, V. & Meer, P. (2000), Real-time tracking of non-rigid objects using mean shift, *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (CVPR'00), pp. 142-149.

Cootes, T. & Taylor, C. (2001), Statistical models of appearance for medical image analysis and computer vision.

Costen, N., Brown, M. & Akamatsu, S. (2004), Sparse models for gender classification, *in Proceedings of the 6th IEEE International Conference on Automatic Face and Gesture Recognition* (FG'04), pp. 201-206.

Cottrell, G. W. & Metcalfe, J. (1990), EMPATH: Face, emotion, and gender recognition using holons, *in* Richard Lippmann, John E. Moody & David S. Touretzky, ed., *Proceedings of the Advances in Neural Information Processing Systems* 3 (NIPS), Morgan Kaufmann, pp. 564-571.

CSU (2003), The CSU face identification evaluation system.

D'Hooge, H. & Goldsmith, M. (2001), Game design principles for the Intel® Play™ Me2Cam* virtual game system, *Intel Technology Journal* **5**(4).

Damasio, A. (1994), *Descartes' Error*, Grosset/Putnam.

Darrell, T., Tollmar, K., Bentley, F., Checka, N., Morency, L., Rahimi, A. & Oh, A. (2002), Face-responsive interfaces: From direct manipulation to perceptive presence, *in Proceedings of the 4th International Conference on Ubiquitous Computing* (UbiComp'02), Springer-Verlag, London, UK, pp. 135-151.

Daugman, J. G. (1988), Complete discrete 2-D Gabor transforms by neural networks for image analysis and compression, *IEEE Transactions on Acoustics, Speech, and Signal Processing* **36**(7), 1169-1179.

Duda, R. O., Hart, P. E. & Stork, D. G. (2001), *Pattern Classification*, A Wiley-Interscience Publication.

Edelman, B., Valentin, D. & Abdi, H. (1998), Sex classification of face areas: How well can a linear neural network predict human performance, *Journal of Biological Systems* **6**(3), 241-263.

Ekman, P. (1982), *Handbook of Methods in Nonverbal Behaviour Research*, Cambridge University, chapter: Methods for measuring facial actions, pp. 45-90.

Ekman, P. & Friesen, W. V. (1978), *Facial Action Coding System: A Technique for the Measurement of Facial Movement*, Consulting Psychologists Press.

Ekman, P. & Friesen, W. V. (1971), Constants across cultures in the face and emotion, *Journal of Personality and Social Psychology* **17**(2), 124-129.

Ekman, P., Rosenberg, E. & Hager, J. (1998), Facial action coding system affect interpretation dictionary (FACSAID), http://face-and-emotion.com/dataface/facsaid/description.jsp.

EyeToy, Sony EyeToy, 2005.

Face Recognition Homepage (2007), http://www.face-rec.org/.

Faraj, M. & Bigun, J. (2007), Audio-visual person authentication using lip-motion from orientation maps, *Pattern Recognition Letters* **28**(11), 1368-1382.

Farkas, L. G. (1994), *Anthropometry of the Head and Face*, Raven Press.

Fasel, B. & Luettin, J. (2003), Automatic facial expression analysis: A survey, *Pattern Recognition* **36**(1), 259-275.

Fasel, B. & Luettin, J. (2000), Recognition of asymmetric facial action unit activities and intensities, *in Proceedings of the 15th International Conference on Pattern Recognition* (ICPR'00), pp. 1100-1103.

Fellous, J. (1997), Gender discrimination and prediction on the basis of facial metric information, *Vision Research* **37**(14), 1961-1973.

FG-NET, FG-NET Aging database, 2007.

Freund, Y. & Schapire, R. E. (1997), A decision-theoretic generalization of on-line learning and an application to boosting, *Journal of Computer and System Sciences* **55**(1), 119-139.

Friedman, J., Hastie, T. & Tibshirani, R. (1998), Additive logistic regression: a statistical view of boosting, *Technical Report*, Stanford University.

FRVT (2006), Face recognition vendor test, http://www.frvt.org/.

Gavrila, D. M. (1999), The visual analysis of human movement: A survey, *Computer Vision and Image Understanding* **73**(1), 82-98.

Golomb, B. A., Lawrence, D. T. & Sejnowski, T. J. (1990), SEXNET: A neural network identifies sex from human faces, *in Proceedings of the Advances in Neural Information Processing Systems* 3 (NIPS), Morgan Kaufmann, pp. 572-579.

Gonzalez, R. C. & Woods, R. E. (2002), *Digital Image Processing*, Prentice Hall.

Gorodnichy, D. O. & Roth, G. (2004), Nouse 'use your nose as a mouse' perceptual vision technology for hands-free games and interfaces, *Image and Vision Computing* **22**(12), 931-942.

Graf, A. B. A. & Wichmann, F. A. (2002), Gender classification of human faces, *in Proceedings of the 2nd International Workshop on Biologically*

Motivated Computer Vision (BMCV'02), Springer-Verlag, London, UK, pp. 491-500.

Gray, M. S., Lawrence, D. T., Golomb, B. A. & Sejnowski, T. J. (1995), A perceptron reveals the face of sex, *Neural Computation* **7**(6), 1160-1164.

Gutta, S., Wechsler, H. & Phillips, P. J. (1998), Gender and ethnic classification of face images, *in Proceedings of the 3rd IEEE International Conference on Automatic Face and Gesture Recognition* (FG'98), pp. 194-199.

Hakulinen, J. (2006), Software tutoring in speech user interfaces, *PhD Thesis*, University of Tampere.

Hayashi, J., Yasumoto, M., Ito, H. & Koshimizu, H. (2002), Age and gender estimation based on wrinkle texture and color of facial images, *in Proceedings of the 16th International Conference on Pattern Recognition* (ICPR'02), pp. 405-408.

Heisele, B., Poggio, T. & Pontil, M. (2000), Face detection in still gray images, *Technical Report*, Massachusetts Institute of Technology, AI Memo 1687.

Hewett, T., Baecker, R., Card, S., Carey, T., Gasen, J., Mantei, M., Perlman, G., Strong, G. & Verplank, W. (1992), *ACM SIGCHI Curricula for Human-Computer Interaction*, ACM Press.

Hjelmås, E. & Low, B. K. (2001), Face detection: A survey, *Computer Vision and Image Understanding* **83**(3), 236-274.

Hosoi, S., Takikawa, E. & Kawade, M. (2004), Ethnicity estimation with facial images, *in Proceedings of the 6th IEEE International Conference on Automatic Face and Gesture Recognition* (FG'04), pp. 195-200.

Huang, C., Ai, H., Li, Y. & Lao, S. (2007), High-performance rotation invariant multiview face detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **29**(4), 671-686.

Huang, C., Ai, H., Wu, B. & Lao, S. (2004), Boosting nested cascade detector for multi-view face detection, *in Proceedings of the 17th International Conference on Pattern Recognition* (ICPR'04), IEEE Computer Society, Los Alamitos, CA, USA, pp. 415-418.

Huang, J. & Wechsler, H. (1999), Eye detection using optimal wavelet packets and radial basis functions (RBFs), *International Journal of Pattern Recognition and Artificial Intelligence* **13**(7), 1009-1025.

Huang, L. & Shimizu, A. (2006), A multi-expert approach for robust face detection, *Pattern Recognition* **39**(9), 1695-1703.

Huang, L., Shimizu, A. & Kobatake, H. (2005), Robust face detection using Gabor filter features, *Pattern Recognition Letters* **26**(11), 1641-1649.

Huang, X., Acero, A. & Hon, H. (2001), *Spoken Language Processing: A Guide to Theory, Algorithm and System Development*, Prentice Hall.

Hyrskykari, A. (2006), Eyes in attentive interfaces: Experiences from creating iDict, a gaze-aware reading aid, *PhD Thesis*, University of Tampere.

Hyrskykari, A., Majaranta, P. & Räihä, K. (2005), From gaze control to attentive interfaces, *in Proceedings of HCI International* (HCII'05).

Hämäläinen, P. & Höysniemi, J. (2002), A computer vision and hearing based user interface for a computer game for children, *in Proceedings of the 7th ERCIM Workshop on User Interfaces for All*, pp. 299-318.

Hämäläinen, P., Ilmonen, T., Höysniemi, J., Lindholm, M. & Nykänen, A. (2005), Martial arts in artificial reality, *in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI'05), ACM Press, New York, NY, USA, pp. 781-790.

Höysniemi, J. (2006), Design and evaluation of physically interactive games, *PhD Thesis*, University of Tampere.

Iga, R., Izumi, K., Hayashi, H., Fukano, G. & Ohtani, T. (2003), A gender and age estimation system from face images, *in Proceedings of the Annual Conference on Society of Instrument and Control Engineers* (SICE'03), pp. 756-761.

Jain, A. & Huang, J. (2004a), Integrating independent components and linear discriminant analysis for gender classification, *in Proceedings of the 6th IEEE International Conference on Automatic Face and Gesture Recognition* (FG'04), pp. 159-163.

Jain, A. & Huang, J. (2004b), Integrating independent components and support vector machines for gender classification, *in Proceedings of the 17th International Conference on Pattern Recognition* (ICPR'04), pp. 558-561.

Jesorsky, O., Kirchberg, K. J. & Frischholz, R. (2001), Robust face detection Using the Hausdorff distance, *in Proceedings of the 3rd International Conference on Audio- and Video-Based Biometric Person Authentication* (AVBPA '01), Springer-Verlag, London, UK, pp. 90-95.

Johansson, G. (1975), Visual motion perception, *Scientific American* **232**(6), 76-88.

Kainulainen, A., Hakulinen, J. & Turunen, M. (2007), Speech and sounds in awareness systems, *Unpublished Manuscript*.

Kakumanu, P., Makrogiannis, S. & Bourbakis, N. (2007), A survey of skin-color modeling and detection methods, *Pattern Recognition* **40**(3), 1106-1122.

Kanade, T. (1977), Computer recognition of human faces, *Interdisciplinary Systems Research* **47**.

Kanwisher, N., McDermott, J. & Chun, M. M. (1997), The fusiform face area: A module in human extrastriate cortex specialized for face perception, *The Journal of Neuroscience* **17**(11), 4302-4311.

Karwath, A. (2005), Size comparison of a tick and a match, http://commons.wikimedia.org/wiki/Image:Tick_male_size_comparison_(aka).jpg.

Kawano, T., Kato, K. & Yamamoto, K. (2004), A comparison of the gender differentiation capability between facial parts, *in Proceedings of the 17th International Conference on Pattern Recognition* (ICPR'04), pp. 350-353.

Kelly, M. D. (1970), Visual identification of people by computer, *PhD Thesis*, Stanford University.

Kim, H., Kim, D., Ghahramani, Z. & Bang, S. Y. (2006), Appearance-based gender classification with Gaussian processes, *Pattern Recognition Letters* **27**(6), 618-626.

Klatzky, R. L. & Lederman, S. J. (2002), *Encyclopedia of Cognitive Science*, Macmillan Ref. Ltd, London, chapter: Haptic perception, pp. 508-512.

Kohir, V. V. & Desai, U. B. (1998), Face recognition using a DCT-HMM approach, *in Proceedings of the 4th IEEE Workshop on Applications of Computer Vision* (WACV '98), pp. 226-231.

Kwon, Y. H. & da Vitoria Lobo, N. (1999), Age classification from facial images, *Computer Vision Image Understanding* **74**(1), 1-21.

Kämäräinen, J. (2003), Feature extraction using Gabor filters, *PhD Thesis*, Lappeenranta University of Technology.

Kölsch, M., Turk, M., Höllerer, T. & Chainey, J. (2004), Vision-based interfaces for mobility, *in Proceedings of the 1st Annual International*

Conference on Mobile and Ubiquitous Systems: Networking and Services (MOBIQUITOUS'04), pp. 86-94.

Lanitis, A. (2002), On the significance of different facial parts for automatic age estimation, *in Proceedings of the 14th International Conference on Digital Signal Processing* (DSP'02), pp. 1027-1030.

Lanitis, A., Draganova, C. & Christodoulou, C. (2004), Comparing different classifiers for automatic age estimation, *IEEE Transactions on Systems, Man and Cybernetics* **34**(1), 621-628.

Lanitis, A., Taylor, C. & Cootes, T. (2002), Toward automatic simulation of aging effects on face images, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**(4), 442-455.

Lapedriza, A., Marín-Jiménez, M. J. & Vitria, J. (2006), Gender recognition in non controlled environments, *in Proceedings of the* IEEE *International Conference on Pattern Recognition* (ICPR'06), pp. 834 - 837.

Lewis, P. A. & Critchley, H. D. (2003), Mood-dependent memory, *Trends in Cognitive Sciences* **7**(10), 431-433.

Li, S., cheng, Y. S., Zhang, H. J. & Cheng, Q. S. (2002), Multi-view face alignment using direct appearance models, *in Proceedings of the 5th IEEE International Conference on Automatic Face and Gesture Recognition* (FG'02), pp. 309-314.

Li, S. & Zhang, Z. (2004), FloatBoost learning and statistical face detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **26**(9), 1112-1123.

Lian, H. & Lu, B. (2006), Multi-view gender classification using local binary patterns and support vector machines, *in Proceeding of the 3rd International Symposium on Neural Networks* (ISNN'06), pp. 202-209.

Lien, J. J. (1998), Automatic recognition of facial expressions using hidden Markov models and estimation of expression intensity, *PhD Thesis*, Robotics Institute, Carnegie Mellon University.

Lienhart, R. & Maydt, J. (2002), An extended set of Haar-like features for rapid object detection, *in Proceedings of the International Conference on Image Processing* (ICIP'02), pp. 900-903.

Littlewort, G., Bartlett, M. S., Fasel, I., Susskind, J. & Movellan, J. (2006), Dynamics of facial expression extracted automatically from video, *Image and Vision Computing: Special Issue on Face Processing in Video Sequences* **24**(6), 615-625.

Lisetti, C. & Rumelhart, D. (1998), Facial expression recognition using a neural network, *in Proceedings of the 11th International Flairs Conference*.

Lu, X., Chen, H. & Jain, A. K. (2006), Multimodal facial gender and ethnicity identification, *in Proceedings of the International Conference on Biometrics* (ICB'06), pp. 554-561.

Lu, X. & Jain, A. (2004), Ethnicity identification from face images, *in Proceedings of the SPIE International Symposium on Defense and Security: Biometric Technology for Human Identification*.

Lyons, M. & Akamatsu, S. (1998), Coding facial expressions with Gabor wavelets, *in Proceedings of the 3rd IEEE International Conference on Automatic Face and Gesture Recognition* (FG'98), pp. 200-205.

Lyons, M., Budynek, J., Plante, A. & Akamatsu, S. (2000), Classifying facial attributes using a 2-D Gabor wavelet representation and discriminant analysis, *in Proceedings of the 4th IEEE International Conference on Automatic Face and Gesture Recognition* (FG'00), pp. 202-207.

Matsumoto, D. (1993), Ethnic differences in affect intensity, emotion judgments, display rule attitudes, and self-reported emotional expression in an American sample, *Motivation and Emotion* **17**(2), 107-123.

Miao, J., Yin, B., Wang, K., Shen, L. & Chen, X. (1999), A hierarchical multiscale and multiangle system for human face detection in a complex background using gravity-center template, *Pattern Recognition* **32**(7), 1237-1248.

Mindstroms, LEGO MINDSTRTOMS NXT, http://mindstorms.lego.com/, 2006.

Moeslund, T. B. & Granum, E. (2001), A survey of computer vision-based human motion capture, *Computer Vision and Image Understanding* **81**(3), 231-268.

Moghaddam, B. & Yang, M. (2002), Learning gender with support faces, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**(5), 707-711.

Moghaddam, B. & Yang, M. (2000), Gender classification with support vector machines, *in Proceedings of the 4th IEEE International Conference on Automatic Face and Gesture Recognition* (FG'00), pp. 306-311.

Morimoto, C. H. & Mimica, M. R. M. (2005), Eye gaze tracking techniques for interactive applications, *Computer Vision and Image Understanding* **98**(1), 4-24.

Mäkinen, E., Patomäki, S. & Raisamo, R. (2002), Experiences on a multimodal information kiosk with an interactive agent, *in Proceedings of the NordiCHI 2002*, pp. 275-278.

Mäkinen, E. & Raisamo, R. (accepted), Evaluation of gender classification methods with automatically detected and aligned faces, *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Mäkinen, E. & Raisamo, R. (2002), Real-time face detection for kiosk interfaces, *in Proceedings of APCHI 2002*, pp. 528-539.

Nielsen, J. (1993), *Usability Engineering*, Academic Press, Boston, MA.

Nintendo, Nintendo Wii console, 2006.

Nishino, Satoshi, lgarashi Sachiyo & Matsuda, Atsushi (2004), Gender determining method using thermography, *in Proceedings of the International Conference on Image Processing* (ICIP'04), pp. 2961-2964.

Niu, Z., Shan, S., Yan, S., Chen, X. & Gao, W. (2006), 2D cascaded AdaBoost for eye localization, *in Proceedings of the 18th International Conference on Pattern Recognition* (ICPR'06), pp. 1216-1219.

Noldus (2007), FaceReader, http://www.noldus.com/site/doc200705001.

O'Toole, A., Deffenbacher, K., Valentin, D., McKee, K., Huff, D. & Abdi, H. (1998), The perception of face gender: The role of stimulus structure in recognition and classification, *Memory and Cognition* **26**(1), 146-160.

Ojala, T., Pietikäinen, M. & Harwood, D. (1996), A comparative study of texture measures with classification based on featured distributions, *Pattern Recognition* **29**(1), 51-59.

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.

Olives, J., Sams, M., Kulju, J., Seppala, O., Karjalainen, M., Altosaar, T., Lemmetty, S., Toyra, K. & Vainio, M. (1999), Towards a high quality Finnish talking head, *in Proceedings of the 3rd IEEE Workshop on Multimedia Signal Processing*, pp. 433-437.

Omron (2005), Omron announces "OKAO vision face recognition sensor", world's first face recognition technology for mobile phones, http://www.omron.com/news/n_280205.html.

OpenCV (2006), OpenCV 1.0, Open source computer vision library, http://www.intel.com/technology/computing/opencv/index.htm

Oza, N. C. (2001), Online ensemble learning, *PhD Thesis*, University of California, Berkeley.

Pantic, M. & Rothkrantz, L. J. (2000), Automatic analysis of facial expressions: The state of the art, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**(12), 1424-1445.

Partala, T. & Surakka, V. (2003), Pupil size variation as an indication of affective processing, *International Journal of Human-Computer Studies* **59**(1-2), 185-198.

Pavlovic, V., Sharma, R. & Huang, T. (1997), Visual interpretation of hand gestures for human-computer interaction: a review, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **19**(7), 677-695.

Pentland, A. (2000), Perceptual user interfaces: Perceptual intelligence, *Communications of the ACM* **43**(3), 35-44.

Phillips, P., Grother, P., Micheals, R., Blackburn, D., Tabassi, E. & Bone, J. (2003), FRVT 2002 evaluation report (NISTIR 6965), *Technical Report*, National Institute of Standards and Technology, USA.

Phillips, P. J., Scruggs2, W. T., O'Toole, A. J., Flynn, P. J., Bowyer, K. W., Schott, C. L. & Sharpe, M. (2007), FRVT 2006 and ICE 2006 large-scale results (NISTIR 7408), *Technical Report*, National Institute of Standards and Technology, Gaithersburg, USA, MD 20899.

Phillips, P. J., Wechsler, H., Huang, J. & Rauss, P. J. (1998), The FERET database and evaluation procedure for face recognition algorithms, *Image and Vision Computing* **16**(5), 295-306.

Phimoltares, S., Lursinsap, C. & Chamnongthai, K. (2007), Face detection and facial feature localization without considering the appearance of image context, *Image and Vision Computing* **25**(5), 741-753.

Picard, R. W. (1997), *Affective Computing*, MIT Press.

Piccardi, M. & Jan, T. (2003), Recent advances in computer vision, *The Industrial Physicist* **9**(1), pp. 18-21.

Qualisys (2007), Qualisys, http://www.qualisys.se/.

Reachin (2007), Reachin display, http://www.reachin.se/.

Rigoll, G., Eickeler, S. & Muller, S. (2000), Person tracking in real-world scenarios using statistical methods, *in Proceedings of the 4th IEEE International Conference on Automatic Face and Gesture Recognition* (FG'00), pp. 342-347.

Rodriguez, Y., Cardinaux, F., Bengio, S. & Mariéthoz, J. (2006), Measuring the performance of face localization systems, *Image and Vision Computing* **24**(8), 882-893.

Roth, D., Yang, M. & Ahuja, N. (2000), A SNoW-based face detector, *in Advances in Neural Information Processing Systems* (NIPS'00).

Rowley, H. A., Baluja, S. & Kanade, T. (1998a), Neural network-based face detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**(1), 23-38.

Rowley, H., Baluja, S. & Kanade, T. (1998b), Rotation invariant neural network-based face detection, *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (CVPR'98), pp. 38-44.

Rurainsky, J. & Eisert, P. (2003), Template-based eye and mouth detection for 3D video conferencing, *in Proceedings of the 8th International Workshop on Visual Content Processing and Representation* (VLBV'03), pp. 23-31.

Saatci, Y. & Town, C. (2006), Cascaded classification of gender and facial expression using active appearance models, *in Proceedings of the 7th IEEE International Conference on Automatic Face and Gesture Recognition* (FG'06), pp. 393-400.

Samal, A. & Iyengar, P. A. (1992), Automatic recognition and analysis of human faces and facial expressions: A survey, *Pattern Recognition* **25**(1), 65-77.

Schapire, R. E. & Singer, Y. (1999), Improved boosting algorithms using confidence-rated predictions, *Machine Learning* **37**(3), 297-336.

Scheenstra, A., Ruifrok, A. & Veltkamp, R. (2005), A survey of 3D face recognition methods, *in Proceedings of the 5th International Conference on Audio- and Video-Based Biometric Person Authentication* (AVBPA'05), pp. 891-899.

Schneiderman, H. & Kanade, T. (2000), A statistical method for 3D object detection applied to faces and cars, *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (ICPR'00), pp. 746-751.

Sharma, P. & Reilly, R. (2003), A colour face image database for benchmarking of automatic face detection algorithms, *in Proceedings of the 4th EURASIP Conference focused on Video/Image Processing and Multimedia Communications*, pp. 423-428.

Selker, T., Lockerd, A. & Martinez, J. (2001), Eye-R, a glasses-mounted eye motion detection interface, *in Extended Abstracts on Human Factors in Computing Systems* (CHI '01), ACM Press, New York, NY, USA, pp. 179-180.

Sergent, J., Ohta, S. & MacDonald, B. (1992), Functional neuroanatomy of face and object processing: A positron emission tomography study, *Brain* **115**(1), 15-36.

Shakhnarovich, G., Viola, P. A. & Moghaddam, B. (2002), A unified learning framework for real time face detection and classification, *in Proceedings of the 5th IEEE International Conference on Automatic Face and Gesture Recognition* (FG'02), pp. 14-21.

Shan, C., Tan, T. & Wei, Y. (2007), Real-time hand tracking using a mean shift embedded particle filter, *Pattern Recognition* **40**(7), 1958-1970.

Shen, L. & Bai, L. (2006a), A review on Gabor wavelets for face recognition, *Pattern Analysis and Applications* **9**(2), 273-292.

Shen, L. & Bai, L. (2006b), MutualBoost learning for selecting Gabor features for face recognition, *Pattern Recognition Letters* **27**(15), 1758-1767.

Sobottka, K. & Pitas, I. (1996), Segmentation and tracking of faces in color images, *in Proceedings of the 2nd IEEE International Conference on Automatic Face and Gesture Recognition* (FG'96), pp. 236-241.

Starner, T. & Pentland, A. (1995), Real-time American sign language recognition from video using hidden Markov models, *in Proceedings of the International Symposium on Computer Vision* (ISCV'95), pp. 265-270.

Stegmann, M. B., Ersboll, B. K. & Larsen, R. (2003), FAME - a flexible appearance modelling environment, *IEEE Transactions on Medical Imaging* **20**(10), pp. 1319-1331.

Sun, N., Zheng, W., Sun, C., Zou, C. & Zhao, L. (2006), Gender classification based on boosting local binary pattern, *in Proceedings of the 3rd International Symposium on Neural Networks* (ISNN'06), pp. 194-201.

Sun, Z., Bebis, G., Yuan, X. & Louis, S. J. (2002b), Genetic feature subset selection for gender classification: A comparison study, *in Proceedings*

of the IEEE Workshop on Applications of Computer Vision (WACV'02), pp. 165-170.

Sun, Z., Yuan, X., Bebis, G. & Louis, S. (2002a), Neural-network-based gender classification using genetic search for Eigen-feature selection, *in Proceedings of the International Joint Conference on Neural Networks* (IJCNN'02), pp. 2433-2438.

Sung, K. & Poggio, T. (1998), Example-based learning for view-based human face detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**(1), 39-51.

Surakka, V., Illi, M. & Isokoski, P. (2004), Gazing and frowning as a new human-computer interaction technique, *ACM Transactions on Applied Perception* **1**(1), 40-56.

Tamminen, T. & Lampinen, J. (2006), Sequential Monte Carlo for Bayesian matching of objects with occlusions, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **28**(6), 930-941.

Tamura, S., Kawai, H. & Mitsumoto, H. (1996), Male/female identification from 8×6 very low resolution face images by neural network, *Pattern Recognition* **29**(2), 331-335.

Tan, X., Chen, S., Zhou, Z. & Zhang, F. (2006), Face recognition from a single image per person: A survey, *Pattern Recognition* **39**(9), 1725-1745.

Tian, Y., Kanade, T. & Cohn, J. (2001), Recognizing action units for facial expression analysis, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **23**(2), 97-115.

Tivive, F. H. C. & Bouzerdoum, A. (2006), A shunting inhibitory convolutional neural network for gender classification, *in Proceedings of the 18th International Conference on Pattern Recognition* (ICPR'06), pp. 421-424.

Tobii eye tracker, http://www.tobii.com/, 2007.

Toyota (2007), Pre-crash safety, http://www.toyota.eu/06_Safety/03_understanding_active_safety/06_precrash_safety.aspx.

Tranel, D., Damasio, A. R. & Damasio, H. (1988), Intact recognition of facial expression, gender, and age in patients with impaired recognition of face identity, *Neurology* **38**(5), 690-696.

Tsao, D. (2006), Eppendorf 2006 winner: A dedicated system for processing faces, *Science* **314**(5796), 72-73.

Tsao, D. Y., Freiwald, W. A., Tootell, R. B. H. & Livingstone, M. S. (2006), A cortical region consisting entirely of face-selective cells, *Science* **311**(5761), 670-674.

Turk, M. & Kölsch, M. (2004), Perceptual interfaces (2003-33), *Technical Report*, University of California.

Ueki, K., Hayashida, T. & Kobayashi, T. (2006), Subspace-based age-group classification using facial images under various lighting conditions, *in Proceedings of the 7th IEEE International Conference on Automatic Face and Gesture Recognition* (FG'06).

Upcode, http://www.upcode.fi/, 2007.

Vicon (2007), Vicon MX, http://www.vicon.com/.

Viola, P. & Jones, M. (2001), Rapid object detection using a boosted cascade of simple features, *in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (CVPR'01), pp. 511-518.

VisionSDK (2000), Microsoft VisionSDK.

Walavalkar, L., Yeasin, M., Narasimhamurthy, A. M. & Sharma, R. (2003), Support vector learning for gender classification using audio and visual cues, *International Journal of Pattern Recognition and Artificial Intelligence* **17**(3), 417-439.

Wang, J. J. & Singh, S. (2003), Video analysis of human dynamics - a survey, *Real-Time Imaging* **9**(5), 321-346.

Wang, T., Ai, H. & Huang, G. (2003), A two-stage approach to automatic face alignment, *in Proceedings of the 3rd International Symposium on Multispectral Image Processing and Pattern Recognition*, pp. 558-563.

Want, R., Hopper, A., Falcão, V. & Gibbons, J. (1992), The active badge location system, *ACM Transactions on Information Systems* **10**(1), 91-102.

Whittle, M. W. (1996), Clinical gait analysis: A review, *Human Movement Science* **15**(3), 369-387.

Wilhelm, T., Böhme, H. & Gross, H. (2005), Classification of face images for gender, age, facial expression, and identity, *in Proceedings of the 15th International Conference on Artificial Neural Networks: Biological Inspirations* (ICANN'05), pp. 569-574.

Wiskott, L., Fellous, J., Krüger, N. & von der Malsburg, C. (1995), Face recognition and gender determination, *in Proceedings of the*

International Workshop on Automatic Face and Gesture Recognition, pp. 92-97.

Wren, C., Azarbayejani, A., Darrell, T. & Pentland, A. (1997), Pfinder: Real-time tracking of the human body, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **19**(7), 780-785.

Wu, B., Ai, H. & Huang, C. (2003a), LUT-based Adaboost for gender classification, *in Proceedings of International Conference on Audio and Video-Based Biometric Person Authentication* (AVBPA'03), pp. 104-110.

Wu, B., Ai, H. & Huang, C. (2003b), Real-time gender classification, *in Proceedings of the 3rd International Symposium on Multispectral Image Processing and Pattern Recognition*, pp. 498-503.

Wu, K., Otoo, E. & Shoshani, A. (2005), Optimizing connected component labelling algorithms, *in Proceedings of the SPIE, Medical Imaging 2005: Image Processing*, pp. 1965-1976.

Xbox LIVE Vision (2006), http://www.xbox.com/en-US/hardware/x/xboxlivevision/.

Xiao, X. (2002), Face detection and retrieval, *Master's Thesis*, Tsinghua University.

Yang, J. & Waibel, A. (1996), A real-time face tracker, *in Proceedings of the 3rd IEEE Workshop on Applications of Computer Vision* (WACV'96), IEEE Computer Society, Washington, DC, USA, pp. 142-147.

Yang, M., Kriegman, D. & Ahuja, N. (2002), Detecting faces in images: A survey, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**(1), 34-58.

Yang, T., Li, S., Pan, Q., Li, J. & Zhao, C. (2006a), Reliable and fast tracking of faces under varying pose, *in Proceedings of the 7th IEEE International Conference on Automatic Face and Gesture Recognition* (FG'06), pp. 421-426.

Yang, Z., Li, M. & Ai, H. (2006b), An experimental study on automatic face gender classification, *in Proceedings of the 18th IEEE International Conference on Pattern Recognition* (ICPR'06), pp. 1099-1102.

Yilmaz, A., Javed, O. & Shah, M. (2006), Object tracking: A survey, *ACM Computing Surveys* **38**(4), 13.

Yin, P., Essa, I. & Rehg, J. M. (2004), Asymmetrically boosted HMM for speech reading, *in Proceedings of the International Conference on Computer Vision and Pattern Recognition* (CVPR'04).

Yow, K. C. & Cipolla, R. (1997), Feature-based human face detection, *Image and Vision Computing* **15**(9), 713-735.

Zhang, L., Ai, H., Xin, S., Huang, C., Tsukiji, S. & Lao, S. (2005), Robust face alignment based on local texture classifiers, *in Proceedings of the IEEE International Conference on Image Processing* (ICIP´05), pp. 354-357.

Zhao, W., Chellappa, R., Phillips, P. & Rosenfeld, A. (2003), Face recognition: A literature survey, *ACM Computing Surveys* **35**(4), 399-458.

Zhou, H. & Hu, H. (2004), A survey - human movement tracking and stroke rehabilitation (CSM-420), *Technical Report*, University of Essex.

Appendix 1

Summary of the existing gender classification studies in chronological order. The list of the studies from the computer vision viewpoint is fairly complete. However, the psychological (including neurophysiology) studies are just some representative examples.

Study	**Viewpoint**	**Main contribution(s)**	**Automatic face detection used?**	**Machine learning techniques**	**Data used in the experiments**	**Best classification rate %**
Tranel et al., 1988	Psychology	Facial expression, gender and age classification were studied in patients with impairment in face identity recognition	Not applicable	Not applicable	Not applicable	Not applicable
Cottrell and Metcalfe, 1990	Computer vision	Auto-associative networks were used for face, gender and emotion recognition	No	Auto-associative network, perceptron	Non-public, 160 images of 10 male and 10 female subjects, 64*64 face image size	100
Golomb et al., 1990	Computer vision	Auto-associative networks were used for gender classification	No	Auto-associative network, perceptron	Non-public, 45 male and 45 female faces, 30*30 face image size	91.9
Bruce et al., 1993	Psychology	Importance of the 2D-, 3D-shape, texture, and their interrelationships investigated	Not applicable	Not applicable	3D laser-scans of the faces without hair and	Not applicable

		for human performance			eyes closed	
Burton et al., 1993	Psychology	Importance of the different measures of faces were inspected by comparing discriminant function analysis results to human performance	No	Discriminant function analysis	91 male faces, 88 female faces, full face and profile face	94
Abdi et al., 1995	Psychology	Investigation of the neural networks for gender classification Analysis of the importance of different facial features for gender classification	No	PCA, RBF network, perceptron	Non public database, 80 male faces, 80 female faces, face image size 151*225	91.8
Brunelli and Poggio, 1995	Computer vision	HyberBF networks trained with automatically extracted geometrical features were experimented with	No	HyperBF network	Non-public, 21 males, 21 females	79
Gray et al., 1995	Computer vision	Experiments with different face resolutions when using neural network for classification	No	Perceptron	Non-public, 44 male and 46 female faces; 10*10, 15*15, 22*22, 30*30 and 60*60 face image size	81
Wiskott et al., 1995	Computer vision	An object recognition system that uses graphs to represent faces was presented and	No	Elastic graph matching, Gabor	Non-public, 73 males and 39 females, 7*4	91.3

		applied to face and gender recognition		wavelets	rectangular grid	
Tamura et al., 1996	Computer vision	Very low resolution face images were used for neural network based gender classification	No	Neural network	Non-public, Japanese faces, two data sets each containing 30 males and 30 females; 32*32, 32*24, 16*16, 16*12, 8*8 and 8*6 face image size	97 for the best classifier and 93 for the best classifier with 8*8 size face images
Fellous, 1997	Psychology	Horizontal and vertical facial measurements were used for gender classification and the results compared to the results of earlier studies	No	Discriminant function analysis, PCA	Non-public and FERET (Phillips et al., 1998) images, 57 males and 52 females, five facial measurements selected from 22	89.7
Edelman et al., 1998	Psychology	Human and neural network performance were compared with full face images, top portion of the faces and bottom portion of the faces	No	Auto-associative network, perceptron, PCA	Non-public, 80 male faces and 80 female faces	78
Gutta et al., 1998	Computer vision	An ensemble of RBF networks and inductive decision trees was used for gender and ethnic classification	No	RBF network, Inductive decision tree	FERET database (Phillips et al., 1998), 1906 male and 1100 female face images of 1009 subjects, 256*384 face image size	96

O'Toole et al., 1998	Psychology	Effects of femininity, masculinity and attractiveness as rated by humans on the gender classification and distinguishing seen faces from unseen faces were studied	No	PCA	Non-public, 75 males, 75 females, 150*225 face image size	Not applicable
Lyons et al., 2000	Computer vision	2D Gabor wavelets and LDA was used for gender, ethnicity and facial expression classification	No	Elastic graph matching (Lades et al., 1993), Gabor wavelets, PCA, LDA	Non-public, 106 male and 76 female faces	92
Moghaddam and Yang, 2000(/2002)	Computer vision	Gender classification of SVM was compared to several other classifiers using FERET (Phillips et al., 1998) images	Yes	Maximum-likelihood estimation, SVM, RBF network, FLD, Nearest neighbor classifier, linear classifier, Quadratic classifier	FERET database (Phillips et al., 1998), 1044 males and 711 females, 84*48 face-prints, 21*12 low-resolution face-prints	96.62 for the 21*12 low-resolution face-prints and 1 percentage difference to the 84*48 face-prints
Cheng et al., 2001	Psychology	Differences in gender classification between children's and adults' faces were studied	No	PCA, perceptron	Non-public, 50 males, 50 females, 50 Caucasian adults, 50 Caucasian children, 256*256 face image size	90 for the adults and 85 for the children

Graf and Wichmann, 2002	Computer vision	PCA and LLE with SVM were used for gender classification	No	PCA, LLE, SVM	Non-public (created by Blanz and Vetter, 1999), laser scans, 100 males and 100 females, 256*256 image size	94.84
Hayashi et al., 2002	Computer vision	Gender and age estimation based on face color and wrinkles	No	Hough transform, look-up table	Commercial HOIP – FACE-DB (2002), Asians, 150 males and 150 females	83
Shakhnarovich et al., 2002	Computer vision	A common framework with Adaboost was used for face detection, gender classification and ethnicity classification	Yes	Adaboost, cascade classifier, SVM	Non-public images collected from the WWW, 3500 images, 24*24 face image size	79.0
Sun et al., 2002a	Computer vision	Genetic algorithms were used to select useful eigenvectors for gender classification	Yes	PCA, GA, Bayesian classifier, neural network	Non-public, 200 male and 200 female images, 50*50 face image size	89.7
Sun et al., 2002b	Computer vision	Genetic algorithms were used to select useful eigenvectors for gender classification	Yes	PCA, GA, LDA, Bayesian classifier, neural network, SVM	Non-public, 200 male and 200 female images, 48*40 face image size	95.3 obtained with SVM
Castrillón et al., 2003	Computer vision	Experiments with SVM and NNC for still images and video	Yes	PCA, NNC, SVM	Non-public, 450 still images; 48 video stream sequences and 1000	98.57

		stream			faces for training	
Iga et al., 2003	Computer vision	Faces are detected based on skin color and facial features are extracted and used as input for an SVM classifier	Yes	Gabor wavelets, SVM	Non-public, 150 male and 150 female faces	93.1
Walavalkar et al., 2003	Computer vision	Facial and audio data were used separately for gender classification Classification for the facial data was done with SVM using linear, polynomial and RBF kernels, and three different data representations	Yes	PCA, NMF, SVM	Non-public database where FERET database (Phillips et al., 1998) was a part and Stanford Medical Student database (Diaco et al., 2000), 4000 male and 4000 female faces from the non-public database, 200 male and 200 female faces from the Stanford Medical Student database, 20*20 face image size	92.5 for the Stanford database while using PCA and RBF kernel with SVM, 84.81 for the non-public database when using PCA and RBF kernel with SVM
Wu et al., 2003a	Computer vision	LUT Adaboost was introduced and used for gender classification	No	LUT Adaboost, Threshold Adaboost, SVM	FERET (Phillips et al., 1998) and non-public images collected from the WWW, 6800 males and 6800 females, 24*24 and 36*36 face image sizes	90.55 with SVM and 88.00 with LUT Adaboost

Wu et al., 2003b	Computer vision	Face detection, alignment and gender classification were combined and experiments with still images and video stream were carried out. Face detection was done with cascaded LUT Adaboost, classification with LUT Adaboost and alignment with SDAM	Yes	LUT Adaboost, cascade classifier, SDAM, Threshold Adaboost	FERET (Phillips et al., 1998) and non-public images collected from the WWW, 6800 males and 6800 females, 24*24 and 36*36 face image sizes; 37 video clips of 20 males and 17 females	88.00 for the still images and 88.04 for the video clips
Buchala et al., 2004	Computer vision	Curvlinear Component Analysis (CCA) was shown to reduce data dimensionality more efficiently than PCA	No	PCA, CCA, MLP, SVM	Several public and one non-public, 250 males, 250 females, 60*90 and 100*100 face image size	93.75 with SVM
Clutterbuck and Johnston, 2004	Psychology	Speed of the gender classification in humans for the familiar (famous celebrities), learned, and unfamiliar faces was studied	Not applicable	Not applicable	12 unfamiliar, 12 familiar (very famous), and 12 learned faces	Not applicable
Costen et al., 2004	Computer vision	Sparse models were applied to gender classification	No	EBPC, LDA, SVM	Non-public, 300 Japanese face images	94.42 with SVM
Jain and Huang, 2004a	Computer vision	ICA with FLD was used for gender classification	No	ICA, FLD	FERET database (Phillips et al., 1998), 250 male and 250 female faces, 64*96 face	99.3

					image size	
Jain and Huang, 2004b	Computer vision	ICA with SVM, LDA and cosine classifier used for gender classification	No	ICA, cosine classifier, LDA, SVM	FERET database (Phillips et al., 1998), 250 male and 250 female faces, 64*96 face image size	95.67 with SVM
Kawano et al., 2004	Computer vision	Usefulness of the different parts of the face were considered for gender classification while using FDF	No	FDF, LDA, cluster discriminant analysis	Commercial HOIP – FACE-DB (2002), Asians, 150 males and 150 females	93.7 for the whole face
Nishino et al., 2004	Computer vision	Thermography was used for gender classification	No	Emphasized variance value, Mahalanobis distance	The first experiment: thermography data from 3 males and 3 females, 8 measurements per person during 105 seconds The second experiment: thermography data from 2 males and 2 females, measurement during 10 minutes while temperature was increased from 24°C to 27°C	77.5

Ueki et al., 2004	Computer vision	Facial, hair and clothing images were used for gender classification separately and together	No	PCA, GMM, likelihood-based integration of classifications	Non-public, varying amounts of images for each image type, 7432 images for the test with all image types integrated, 32*32 face image size without hair, 32*32 image size with the hair excluding the face, 24*24 image size for the clothing images	92.2 when using facial, hair and clothing images together
BenAbdelkader and Griffin, 2005	Computer vision	A novel local region-based approach for gender classification	Yes	FLD, SVM	Several public, 12,964 faces, face image size 50*40	94.2 with SVM
Buchala et al., 2005	Computer vision and psychology	Whole face, eye area and mouth area used for gender classification separately and as composite The classifiers' and human performance compared	No	PCA, CCA, SOM, SVM	200 male and 200 female images from several public databases, 128*128 and 64*64 size for the face, 32*64 size for eye and mouth images	92.25 for composite classifier using PCA with SVM
Wilhelm et al., 2005	Computer vision	Two feature extraction methods, ICA and AAM based, are used with various classifiers for gender, age, facial expression, and identity	No	ICA, PCA, AAM, NNC, MLP, RBF network, LVQ network	Non-public, 70 people, 7 images per person, face image size 60*70 pixels	92 for the MLP classifier with AAM feature

		recognition				extraction
Baudouin and Humphreys, 2006	Psychology	Gender classification done by humans was studied where upper part of the face was combined with the lower part of the different face possibly of different gender Gender classification of upside-down faces was studied	Not applicable	Not applicable	10 female faces and 10 male faces, 500*381 face-prints	Not applicable
Castrillón et al., 2006	Computer vision	Evolving of the system that learns while it is on-line is studied	Yes	PCA, SVM	Mixture of 6000 images taken from the WWW and public databases, 59*65 face image size ; 900 non-public video streams	83 for the off-line classifier and 75 for the on-line classifier
Kim et al., 2006	Computer vision	GPC was shown to outperform SVM in gender classification GPC can be used to improve classification with SVM	No	Nearest neighbor classifier, LDA, EM-EP with GPC, EM-EP with SVM, SVM with cross-validation	Non-public database and AR database (Martinez and Benavente, 1998), 53 males and 50 females from the non-public database, 70 males and 56 females from the AR database; 256*256, 28*23 and 20*26 face image	99.975 or 97.5 (the used scale not stated) achieved with GPC

					size	
Lapedriza et al., 2006	Computer vision	A system to extract facial features in uncontrolled environments that can be used for gender classification was presented	No	Adaboost, JointBoost	FRGC database (2006), 3440 face images taken under controlled environment and 1886 images taken in cluttered conditions	96.77
Lian and Lu, 2006	Computer vision	SVM with LBP was used for multi-view gender classification	No	LBP, SVM	CAS-PEAL database (Gao et al., 2004), 14,384 images of 1040 people, 150*130 face image size	96.75
Lu et al., 2006	Computer vision	Intensity and range images were used separately and together for gender and ethnicity classification	No	SVM	University of Notre Dame Biometrics Database (Chang et al., 2003) and a non-public database, 944 scans of 276 people from UND database and 296 scans of 100 subject from the non-public database	91.0
Saatci and Town 2006	Computer vision	Gender and expression classification using AAM and SVM were studied so that both genders had separate expression classifiers and vice	No	AAM, SVM	Several public, 809 images, 60-dimensional feature vectors created of the AAM control parameters	97.6 for stand-alone gender classifier, 94.8 with preceding

		versa Cascade structure where one expression recognizer precedes the other was studied				expression classification
Sun et al., 2006	Computer vision	LBP was used with Adaboost and with SOM for gender classification	No	LBP, Chi square distance, Adaboost, SOM	FERET database (Phillips et al., 1998), 1400 male faces and 1000 female faces, 144*120 face image size	95.75% with Adaboost
Tivive and Bouzerdoum, 2006	Computer vision	Toepliz-connected and binary-connected convolutional neural networks were used for gender classification	No	Toepliz-connected and binary-connected convolutional neural network, multi-layer perceptron	FERET database (Phillips et al., 1998) and a non-public database containg mostly images collected from the WWW, 1152 male and 610 female face images from the FERET database, 4000 male and 4000 female face images from the non-public database, 32*32 face image size	97.1 for the FERET images with the binary-connected convolutional neural network, 88.8 for the non-public database with the Toepliz-connected neural network

Yang et al., 2006b	Computer vision	Experiments with varying combinations, of face detection, normalization and gender classification were carried out	Yes	Nested cascade classifier, real Adaboost, Delaunay triangulation, affine mapping, PCA, SVM, FLD	Non-public Chinese face database, 7000 males and 4500 females; 3529 FERET (Phillips et al., 1998) images	97.16 with Chinese faces and 93.8 with FERET images
Baluja and Rowley, 2007	Computer vision	Adaboost and SVM based gender classifiers are compared by varying face normalization steps Pixel comparison operators are used with Adaboost Sensitivity of the classifiers to variations in scale, rotation, and translation is studied	No	Adaboost, SVM	1495 male and 914 female faces from the FERET database (Phillips et al., 1998), 20*20 and 12*21 face image size, a face mask used in part of the experiments	94.3 for Adaboost classifier with 20*20 size face images without mask and without normalized intensities

Appendix 2

Effects of face rotation, scale and translation for the classification rates with classifiers that have been trained with properly aligned face images.

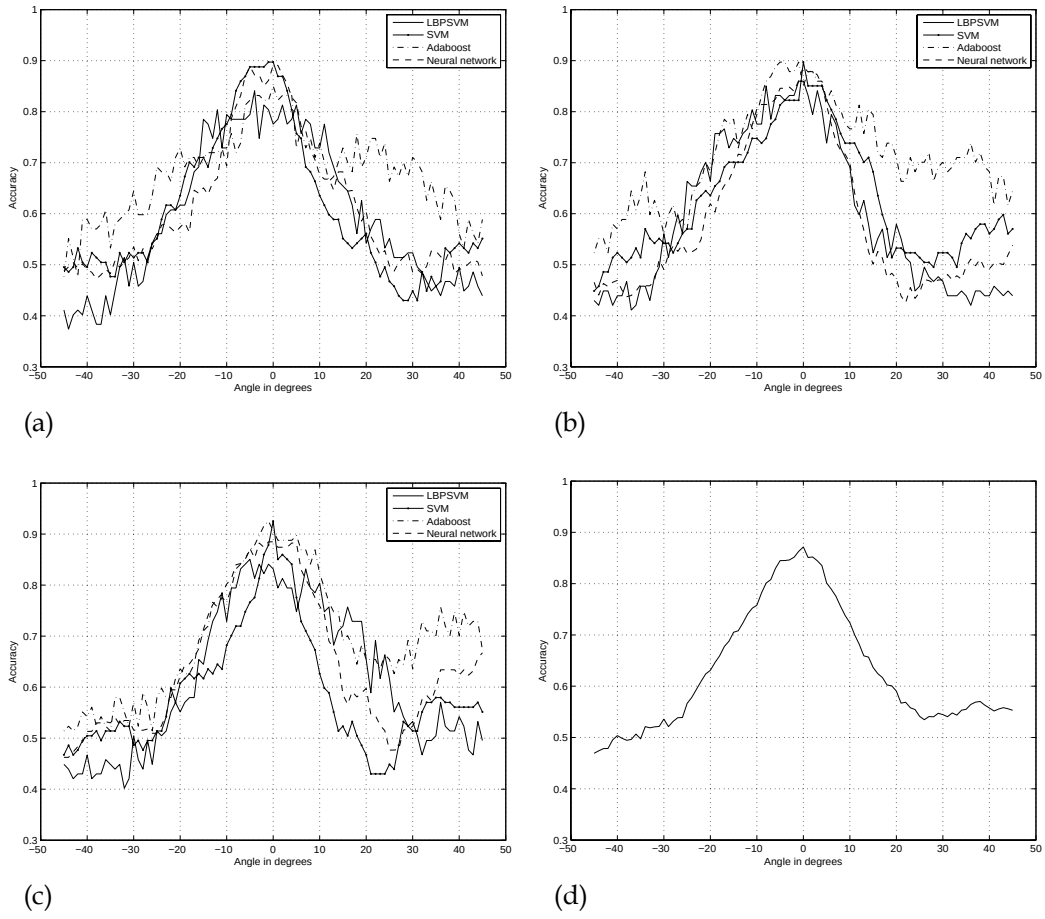


Figure 1. Effect of rotation on classification accuracies with (a) 24*24 size images, (b) 36*36 size images, (c) 48*48 size images, and (d) overall effect of rotation over all image sizes and methods.

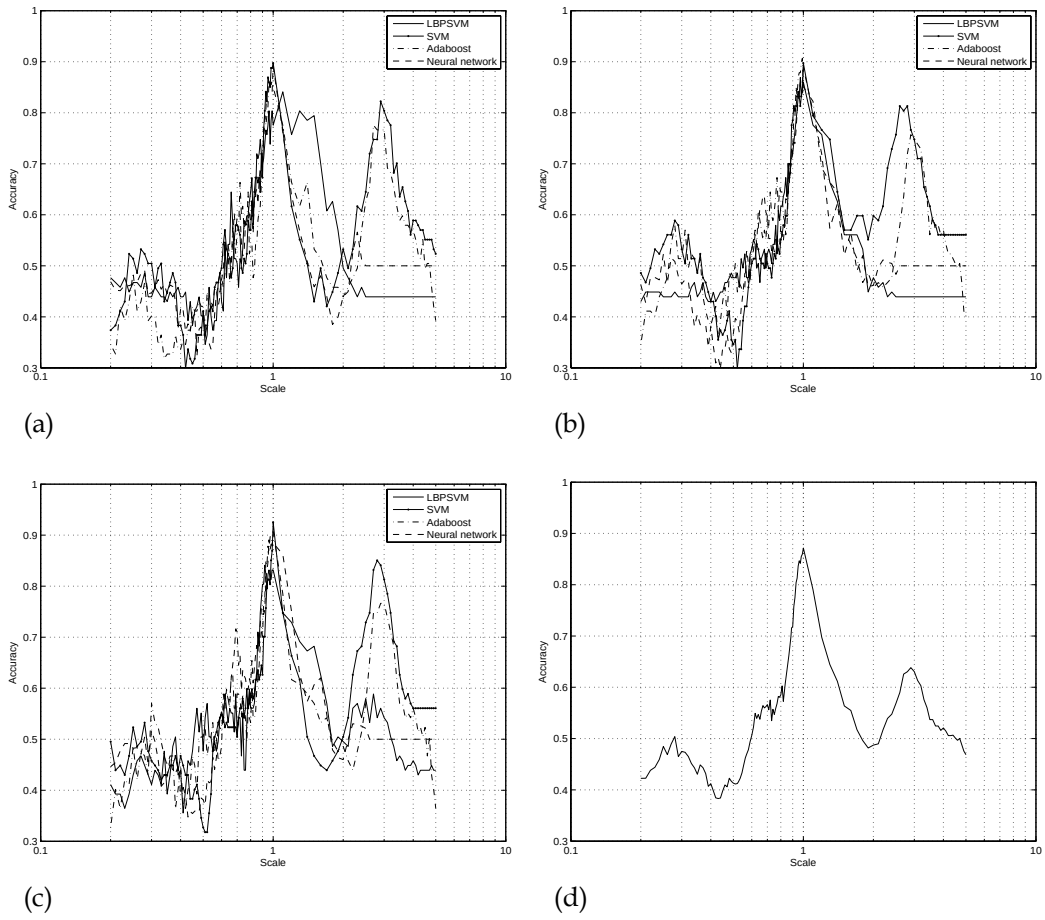


Figure 2. Effect of scale on classification accuracies with (a) 24*24 size images, (b) 36*36 size images, (c) 48*48 size images, and (d) overall effect of scale over all image sizes and methods.

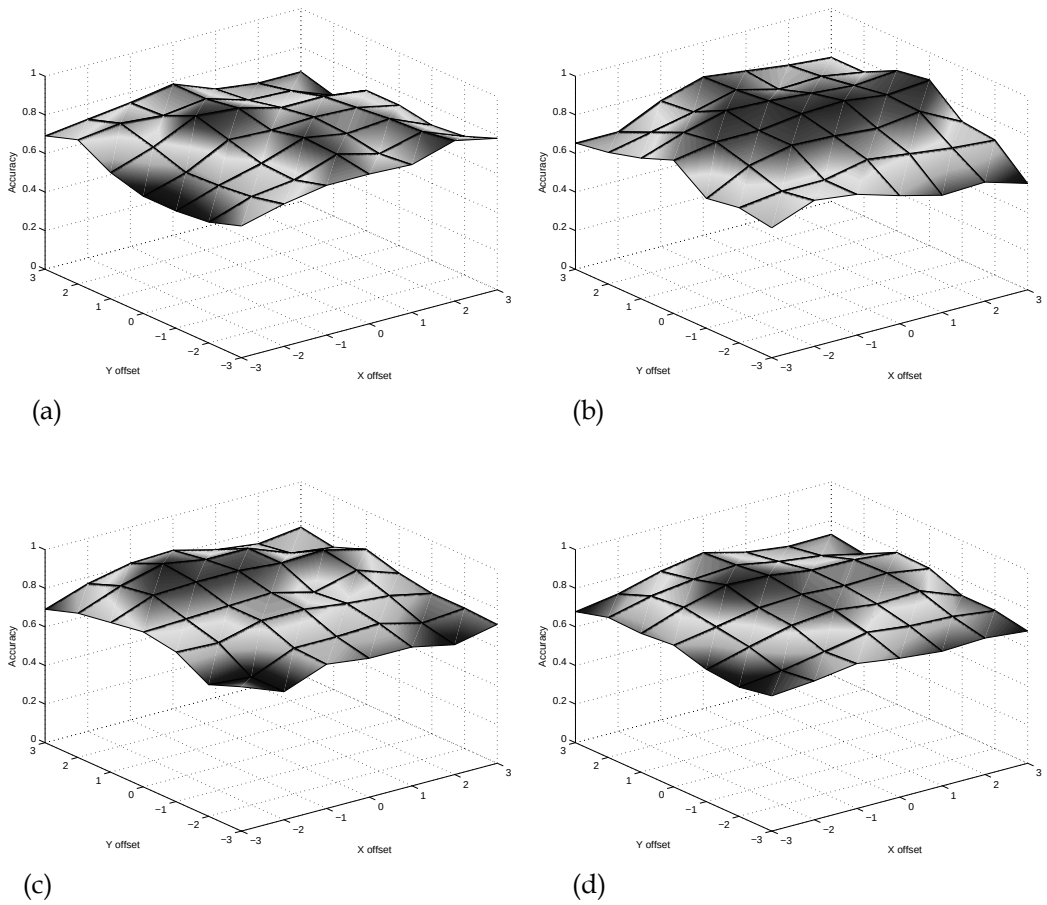


Figure 3. Effect of translation on classification accuracy of SVM with LBP features with (a) 24*24 size images, (b) 36*36 size images, (c) 48*48 size images, and (d) average effect of translation calculated from all image sizes.

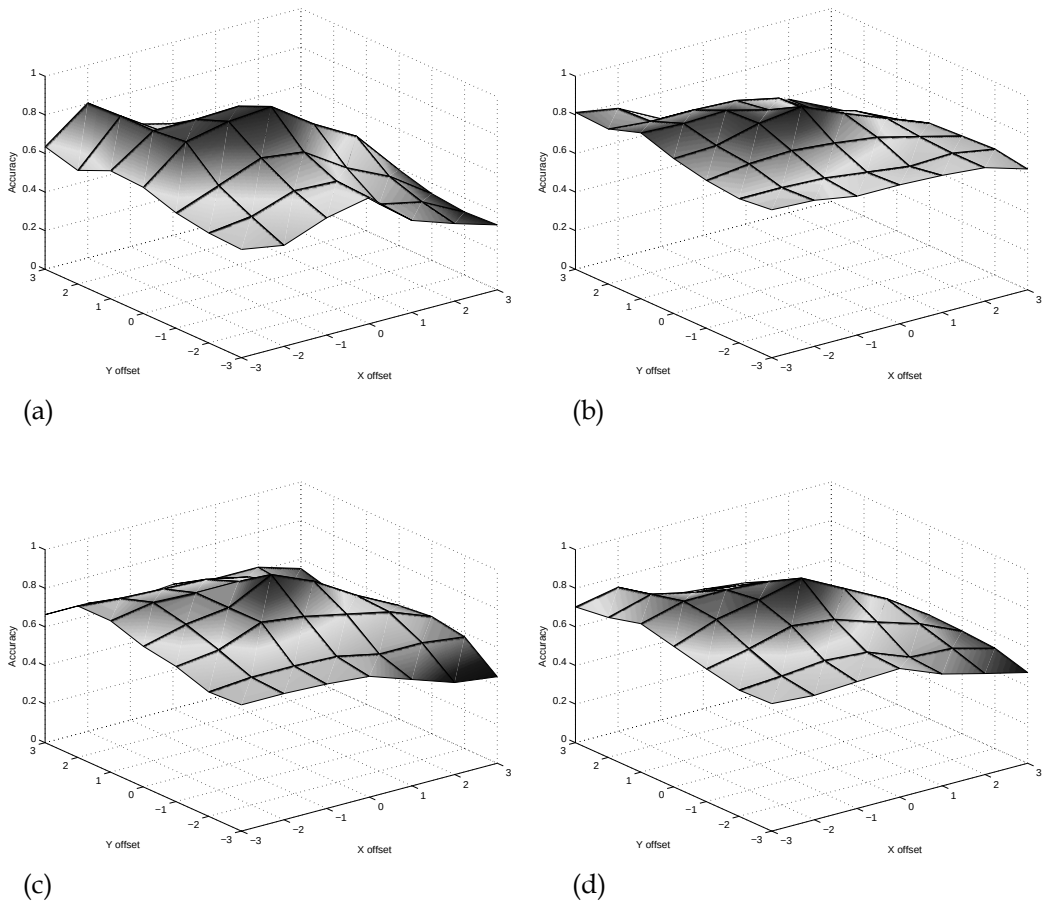


Figure 4. Effect of translation on classification accuracy of the SVM with pixel based input with (a) 24*24 size images, (b) 36*36 size images, (c) 48*48 size images, and (d) average effect of translation calculated from all image sizes.

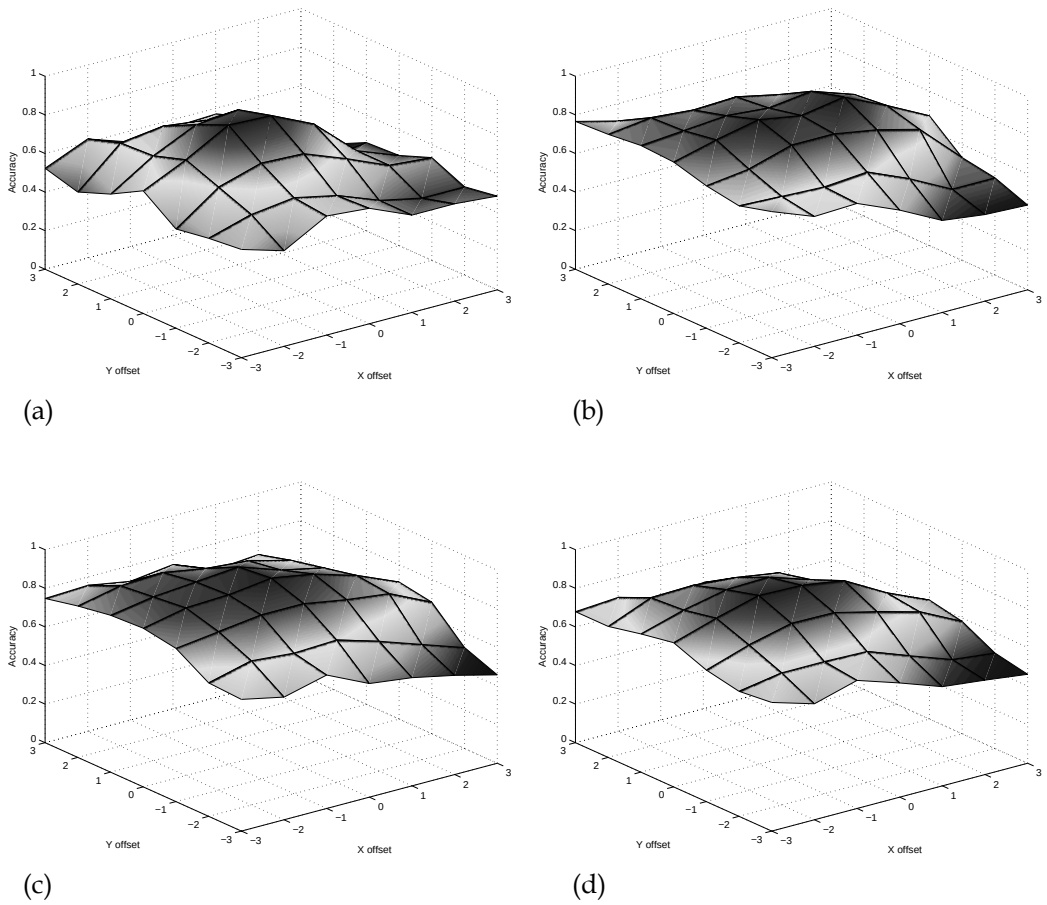


Figure 5. Effect of translation on classification accuracy of threshold Adaboost with Haar-like features with (a) 24*24 size images, (b) 36*36 size images, (c) 48*48 size images, and (d) average effect of translation calculated from all image sizes.

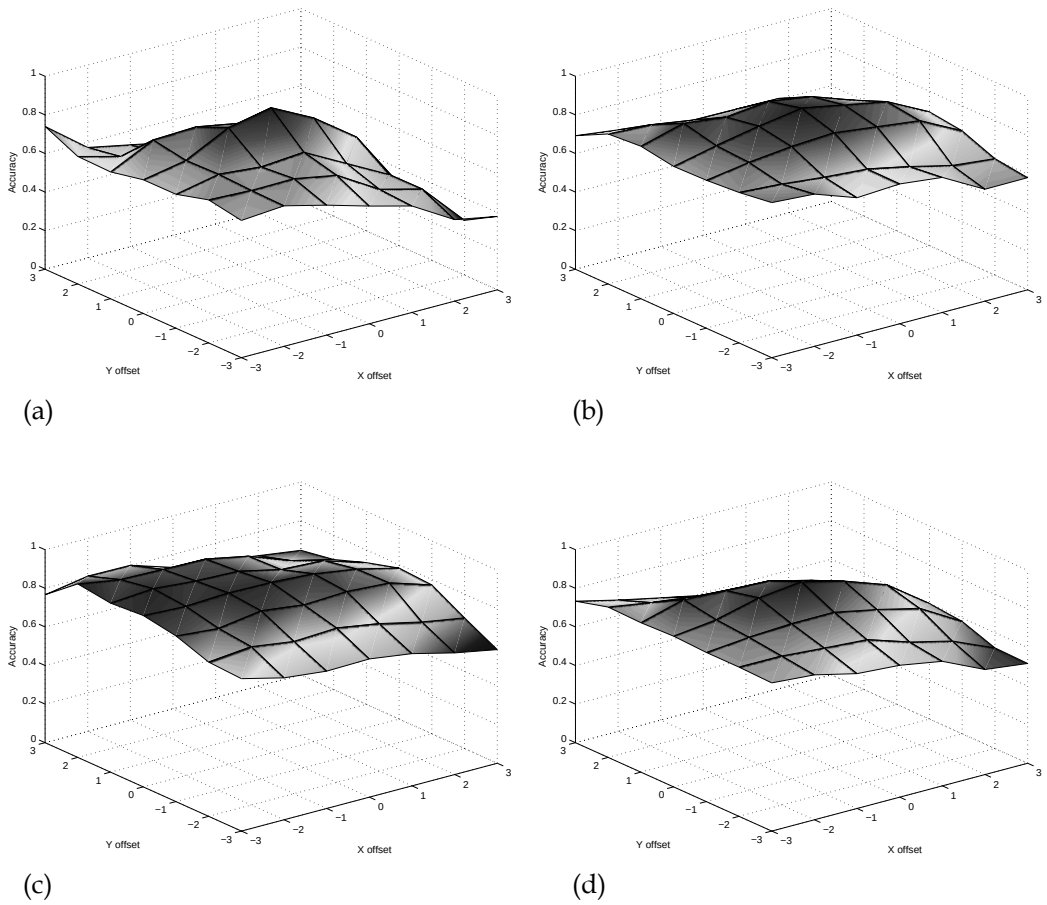


Figure 6. Effect of translation on classification accuracy of multi-layer perceptron with pixel-based input with (a) 24*24 size images, (b) 36*36 size images, (c) 48*48 size images, and (d) average effect of translation calculated from all image sizes.

Publications in the **Dissertations in Interactive Technology** series

1. **Timo Partala:** Affective Information in Human-Computer Interaction
2. **Mika Käki:** Enhancing Web Search Result Access with Automatic Categorization
3. **Anne Aula:** Studying User Strategies and Characteristics for Developing Web Search Interfaces
4. **Aulikki Hyrskykari:** Eyes in Attentive Interfaces: Experiences from Creating iDict, a Gaze-Aware Reading Aid
5. **Johanna Höysniemi:** Design and Evaluation of Physically Interactive Games
6. **Jaakko Hakulinen:** Software Tutoring in Speech User Interfaces
7. **Harri Siirtola:** Interactive Visualization of Multidimensional Data
8. **Erno Mäkinen**: Face Analysis Techniques for Human-Computer Interaction