Graz University of Technology

Institute for Computer Graphics and Vision

Master's Thesis

Glasses detection and segmentation from face portrait images

Paul Urthaler

Graz, Austria, December 2008

Thesis supervisor

Univ. Prof. DI Dr. Horst Bischof

Supervisor

DI Dr. Martin Urschler

Abstract

Face detection and recognition are challenging computer vision tasks, whereas the relevance of this research field gets more important. To verify the ICAO requirements for 'Machine Readable Travel Documents' (MRTD) an 'Eyeglasses Present Classification' and an 'Eyeglasses Segmentation' is developed. For the classification the Viola and Jones classifier is used, which is trained on different databases and uses several different features. In the second part, the 'Eyeglasses Segmentation', the eyeglasses frame is localised with Snakes, which are based on the Gradient Vector Flow field. A registration of the left and right eyeglasses part allows to compare and improve the found curves. To get better results (e.g. in the eye-brows area) the shortest path algorithm from Dijkstra is implemented using a suitable cost function. As a last step the exact detection of the inner and outer frame edge is done, which leads to the exact segmentation of the eyeglasses frame. Consequential the frame thickness and the frame to eye center distances are derived. Finally, a comprehensive evaluation of the eyeglasses detection, segmentation and the determined parameters is presented, compared to manually annotated ground truth data.

Keywords. Facial images, ICAO standard, Machine Readable Travel Documents (MRTD), Eyeglasses, Viola Jones algorithm, AdaBoosting, Snakes, Gradient Vector Flow, Dijkstra algorithm

Kurzfassung

Gesichtsdetektion und Erkennung ist eine herausfordernde Aufgabe, die in der heutigen Zeit immer wichtiger wird. Um den ICAO Standard für maschinenlesbare Reisedokumente (Machine Readable Travel Documents) zu überprüfen, wird deshalb ein Brillendetektor und eine Brillensegmentierung entwickelt. Als Detektor wird der Viola Jones Klassifikator verwendet, der auf verschiedene Datenbanken trainiert wird und unterschiedliche Features verwendet. Im zweiten Teil, der Brillensegmentierung, wird der Rahmen der Brille mit Snakes lokalisiert, wobei diese auf dem 'Gradient Vector Flow' Feld basieren. Eine Registrierung der linken und rechten Brillenhälfte erlaubt, die gefundenen Kurven zu vergleichen und auf einander abzustimmen. Um bessere Resultate zu erzielen (z.B. im Bereich der Augenbrauen) wird zusätzlich der Algorithmus von Dijkstra implementiert. Schlußendlich wird mit Hilfe der groben Lokalisierung die innere und äußere Rahmenkante bestimmt. Daraus werden dann Rahmendicke und der Abstand von Augenmittelpunkt zum Brillenrahmen ermittelt. Die berechneten Daten werden in einer detaillierten und umfangreichen Evaluierung mit annotierten Daten verglichen und ausgewertet.

Keywords. Gesichtsbilder, ICAO Standard, Maschinenlesbare Reisedokumente, Brillen, Viola Jones Methode, AdaBoosting, Snakes, 'Gradient Vector Flow' Feld, Dijkstra Algorithmus

Acknowledgments

First of all I would like to thank my family. They always supported me during my studies and gave me the opportunity to get the education I wanted. I am grateful to Prof. Horst Bischof for giving me the possibility to write my master thesis at the Institute of Computer Vision and Computer Graphics at the Technical University of Graz. Special thanks goes to Dr. Martin Urschler for supervising me during my work and the time he spent, dealing with my questions.

During my studies I got to know many different people and many new friendships. Special thanks goes to Peter Brunner, who gave me useful tips and proofreading my work. Finally, I would like to thank Karin for all her patience and understanding, for sometimes having less time for her.

Contents

List of Figures

Chapter 1

Introduction

Contents

1.1 Motivation and Problem Statement

Biometrics and biometric recognition are areas of research, which become more and more important. The use of biometrics is an approach to get solutions for unresolved problems in surveillance, person identification and verification, because the technological innovation allows to perform measuring and processing of biometrical data fast and trusty. The biometric technologies can be divided into two parts - physiological and behavioral tokens ([9]).

behavioral	**physiological**
voice	DNA
signature	hand geometry and fingerprint
	iris and retina recognition
	face recognition

A lot of research does not only concentrate on one single characteristic, it heads for combinations of biometric tokens. In this work we are concerned with the face recognition and face analysis token and we focus on eyeglasses detection and segmentation as a pre-processing step for face recognition and analysis applications. For robust face recognition it is essential to have normalised pictures, which fullfill the ICAO requirements for 'Machine Readable Travel Documents' ([8], [25], also see Chapter 2.1). Normalised face images should be interference-free and should contain a frontal pose, neutral expression and 'normal' eyeglasses with a typical frame, clear glasses and no reflections. Therefore it is necessary to detect eyeglasses and - if present - to delineate and evaluate the eyeglasses. This leads to solving problems in the areas of classification and segmentation (see Chapter 1.2). Normalised images which come up with all the necessary qualifications increase the reliability of face recognition and detection software.

As described in [8] eyeglasses are allowed in face portrait images for 'Machine Readable Travel Documents' (see Chapter 2.1), but eyeglasses have to satisfy different criteria. The task in this work is to automatically detect the presence of eyeglasses and to locate the eyeglasses in the face portrait image. With the located eyeglasses several different parameters of the eyeglasses appearance can be calculated. All these parameters lead to decisions, whether a face image containing eyeglasses is suitable for a 'Machine Readable Travel Document' or not.

1.2 An Informal Definition of Classification and Segmentation

1.2.1 Classification

In computer vision the classification problem is a widely used task. A class or category is the entity of objects, which have similar features and satisfy the same constraints. To separate the single categories, class limits are set up to distinguish one class from another. A classificator is the policy maker, which decides whether an object belongs to class A or B. The whole system - class limits, entity of classes and classificator - is called classification ([4], see Figure 1.1).

There are two main concepts for getting a classificator - supervised and unsupervised. In supervised classification a set of training data, which represents a given input and output, is given. The goal of the training step is to find a function, which fullfills the conditions of the training data. In unsupervised classification the training algorithm creates a model of the input data and divides it into categories with different characteristics.

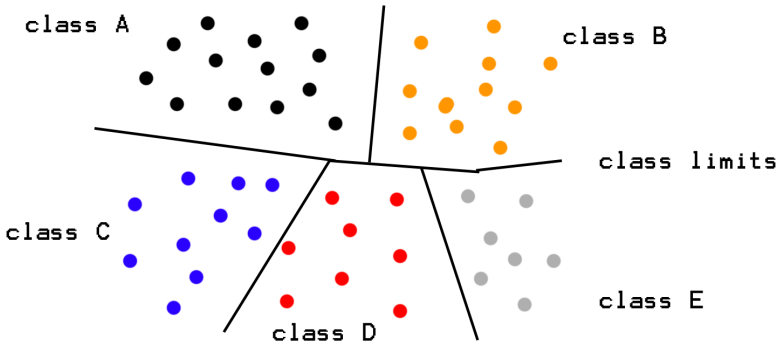


Figure 1.1: Classification problem.

1.2.2 Segmentation

The segmentation procedure is a process of partitioning an input image into different regions of connected pixels. All the pixels of one region fullfill special tokens (e.g. intensity, gradient, color, ...). The goal of a segmentation is to simplify the representation and analyse the input image. Segmentation is normally used to locate objects in images ([22], [7], [23], see Figure 1.2).

The difference between segmentation and classification is, that in classification objects are assigned to classes and in segmentation pixels are connected into regions to divide an image into several areas. But often segmentation and classification are mixed up and it is hard to decide whether one has to deal with a classification or segmentation problem.

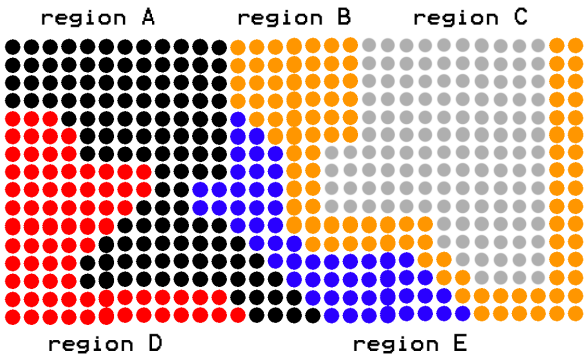


Figure 1.2: Segmentation problem.

1.3 Aims of the Thesis

While there is extensive literature describing the detection ([29], [31]) and removal ([3, 30]) of eyeglasses under the conditions such as occlusions [21], reflections [15] under different parameters, (e.g. frame thickness, position of the frame relative to the eyes) ([20]) there is little knowledge about the correct detection and exact extraction of eyeglasses. A seminal report ([20]) shows that this problem can be approached by the presence, color, frame thickness, and position of the eyeglasses relative to the eyes. The aim of this thesis is to further develop this approach into a readily usable procedure.

The procedure is divided into several parts (see Figure 1.3). First of all a classifier detects if eyeglasses are present or not. The classifier returns a value which represents the presence probability of eyeglasses in a face image. It does not give any information about the position, shape etc.

In case of existing eyeglasses, a subsequent eyeglasses segmentation step is performed. The segmentation returns an exact delineation of the left and the right glasses including the frame, which surrounds the glasses. The nosepiece and earpiece of the frame are neglected.

Via the segmentation different criteria can be derived. The first token is whether the eyeglasses are clear or tinted. Tinted eyeglasses may occlude the iris and result in a loss of biometric information. It is similar with specularities on eyeglasses, which occur due to wrong use of flashes and bad illuminations. The third characteristic describes the frame width of eyeglasses. The last criterion delineates the vertical distance between

eye center and the eyeglasses frame, where there is a difference between the upper und lower distance.

In this work we deal with the presence of eyeglasses, the segmentation, the frame width and frame distance of eyeglasses.

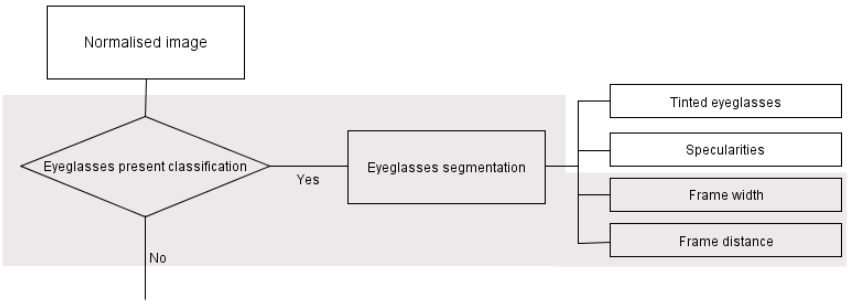


Figure 1.3: Overview of eyeglasses problems.

1.4 Structure of the Thesis

Chapter 2 gives an overview about related work. Section 2.1 will present the international standard to which our data is compared to. Section 2.4 explains the advantages and disadvantages of existing approaches that segment eyeglasses from faces. Section 2.5 presents previous work from the Advanced Computer Vision Research Lab (ACV) and Siemens and summarizes related work and literature on the detection of eyeglasses.

Chapter 3 gives a detailed overview about the developed algorithm. This chapter is structured into two parts - Section 3.2, which explains the eyeglasses detector, and Section 3.3, which gives an overview about the segmentation of eyeglasses.

Chapter 4 is devoted to experiments and their results and presents qualitative and quantitative evaluations on the accuracy of 'Glasses Present Classification' and 'Glasses Segmentation'. The qualitative evaluation shows positive and negative examples of the segmentation and discusses the advantages and disadvantages of the developed algorithm.

Chapter 5 summarises the work and discusses some potential improvements of the developed algorithm.

Chapter 2

Face Images and Eyeglasses - Overview and Related Work

Contents

2.1 ICAO Standard

The ICAO (International Civil Aviation Association *) introduced an international standard (ISO standard) for Machine-Readable Travel Documents (MRTD). A Machine-Readable Travel Document is an international travel document (e.g. passport), that contains eye- and machine-readable data. This ISO standard about 'Biometric Data Interchange Formats' consists of 8 parts, where part 5 deals with face image data (see [8]). The ISO standard defines several requirements for eyeglasses ([8, pg. 22,35]).

- The glasses should be clear and transparent.

*http://www.icao.int

- Tinted and dark eyeglasses should be removed before a photograph is taken.
- The frame should not occlude the eye pupils and eye irises.
- No reflections and lighting artefacts should be visible on the glasses.

Since the presented work adapts this ISO standard it requires ISO standard conform data. Every give face image picture has to be normalised and transformed to the ISO standard requirements (see Figure 2.1). A normalised image fullfills several criteria (see Table 2.1). We use a normalisation algorithm which was presented in [25].

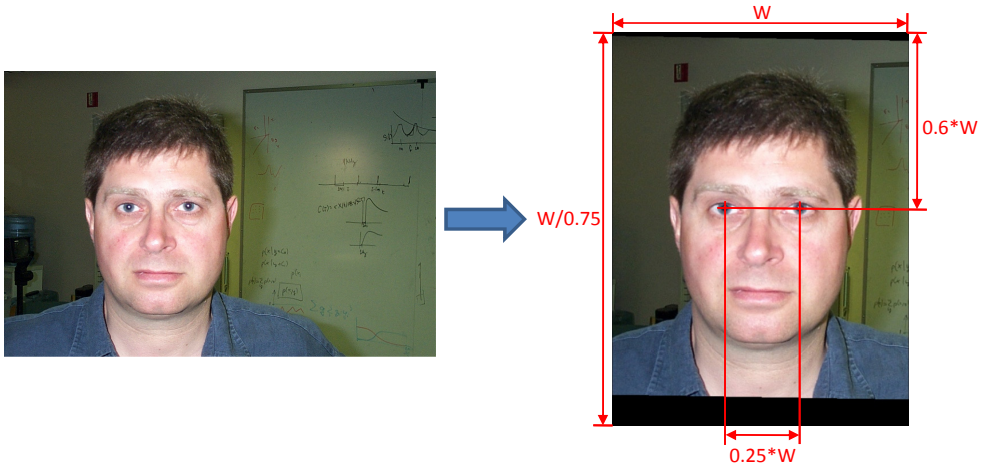


Figure 2.1: The normalisation procedure. The input (left) and the normalised image (right). Image is taken from the Caltech Faces database [1].

Parameter	Value
image width	W
image height	$W/0.75$
Y coordinate of eyes	$0.6 * W$
X coordinate of left eye	$(0.375 * W) - 1$
X coordinate of right eye	$(0.625 * W) - 1$

Table 2.1: Geometric specifications for image proportion and eye position according to ISO standard ([8]).

2.2 A Formal Definition of Classification and Segmentation

In this work two basic computer vision tasks are used. The first one is image segmentation, where a given image B with pixels $f_1, ..., f_n$ is divided into homogeneous disjoint regions. These found regions fullfill the following requirements.

$$\bigcup_{i=1,...,m} R_i = B \tag{2.1}$$

$$R_i \bigcap R_j = \varnothing \tag{2.2}$$

$$f_s \in R_i, f_t \in R_j : f_s \sim f_t \Leftrightarrow i = j \tag{2.3}$$

In literature there are a number of methods to perform a segmentation - pixel- and histogram-based methods (e.g. threshold method), edge-detection methods, region-growing methods, Watershed transformation, multi-scale segmentation, etc ([22]).

Supervised classification consists of two parts, a training and a classification step. Objects $\boldsymbol{x}$ with the features $x_1, ..., x_n$ and classes $y \in Y$ where $y \in [1, ..., m]$ are given. In the training step, objects are manually assigned to classes (Formula 2.4) and a model function f is set up from the given examples.

$$[(\boldsymbol{x}_i, y_j) | \boldsymbol{x} \in X, y_j \in Y, i = 1...n, j = 1...m] \tag{2.4}$$

$$f : X \rightarrow Y \tag{2.5}$$

There are several methods to set up the model function f. Concepts from Gauss or Bayes yield to solutions as well as artificial neural networks, more complex Support Vector machines and its hyperplanes and Boosting methods, which combine a number of weak classifiers to one strong (e.g. AdaBoosting, ...).

In the classification step the model function (Formula 2.5) is evaluated for an unseen object $\boldsymbol{x}$ to a label y.

2.3 Related Work on Eyeglasses Present Classification

There is extensive literature on how to verify the existence of glasses in face images using various different approaches.

Zhong Jing and Robert Mariani [12] developed an eyeglasses detection algorithm, where they analyze the nosepiece in the face image. They did several experiments and demonstrated that the area between the two eyes is the most important criterion to determine if glasses exist or not. After calculating several features of this area, the presence of glasses is determined. The evaluation database contains 419 face images, where 151 people wear eyeglasses. All the face images are frontal view and the eyeglasses have various shapes and colours. The correct detection rate on their database of their classifier is 99.52 %.

In [31] Chenyu Wu et al. present an eyeglasses classifier based on a Support Vector machine (SVM). To get features, the intensity image and calculated orientation image are combined with a PCA. The dimensionality of the resulting feature space is 2752. The application of a Support Vector machine with a polynomial kernel function was trained with 63 images (with and without eyeglasses). A test with 37 images resulted in a detection rate of 81 %. In eyeglasses analysis, where the feature selection is a difficult challenge, the here used orientation image is proven to be a good feature for eyeglasses.

In a further paper Chenyu Wu et al. [30] use a boosting classifier to check the presence of eyeglasses. They define the eye area as a patch of 30x12 pixel. Furthermore they use a cascade of KLBoosting classifier, which they train with 1.271 face images. For evaluating a test data set of 1.386 face images, which are not in the training set, are used. The result showed a correct detection rate of 96 %. Compared to the previous paper ([31]) the same basic features for eyeglasses are used, only the type is changed to a KLBoosting classifier.

Another approach by Bo Wu et. al. [29] uses a boosting classifier with different wavelet features (Haar and Gabor). Results on a database with 3.000 face images from the Feret database and World-Wide-Web show, that the Haar feature has a correct detection rate of 98,4 % and an execution time of 0,09 ms/sample. The Gabor boost has similar results (98,9 % and 1,30 ms/sample), on the other hand the evaluated SVM classifier only achieves 95,5 % and needs 19,1 ms per sample execution time.

2.4 Related Work on Eyeglasses Segmentation

There is extensive literature on how to segment eyeglasses. The aims for the various approaches are quite different. Some approaches aim to remove the eyeglasses and neglect the extraction of the frame. Other approaches aim to find interferences (glasses reflections, shadows, ...). Only very few address the exact frame segmentation.

For frame segmentation there are different basic approaches. In [30] the approach is to remove eyeglasses from a face. An intermediate step localizes the key points on the eyeglasses. For this step Wu et al. use an active shape model (ASM), which contains the main geometric information of eyeglasses - position, size and shape reduced to 15 key points. Using training data and likelihood learning Wu et al. calculate a mean shape, which they use for the localization procedure later.

A Markov-chain Monte Carlo technique finds the best solution for which the initial localization of the eyeglasses is close to the real one. With this solution the assumed eyeglasses region is around the eyes, which can then be easily found.

Wu et al. used 264 samples for training, which where labeled manually. 95 % of the 40 testing images were localized well.

A second approach for segmentation is similar to the first one. In [12] Jing and Mariani first calculate an edge map with a Canny edge detector in the eyeglasses region around the left and right eye. Then they find the initial contour, sector for sector with respect to the distance between eye and edge in symmetry to the left and right eye.

A directed graph to link the 'edge parts', results in the best solution by computing the shortest path between the found edge parts.

The result shows, that 50 % of the eyeglasses are exactly extracted while 30 % are extracted to a satisfactory level. Incorrect detections result from eyebrows crossing the edges of the glasses and the color of the eyeglasses being similar to the color of the skin.

The focus of the remaining literature on how to segment eyeglasses is more on how to remove eyeglasses from a face image. In these cases it is not always necessary to

detect the exact position of the eyeglasses frame. The approach in [3] projects face images into an eigenspace. Face images with eyeglasses and corresponding images without eyeglasses construct this eigenspace. The difference between the original and the reconstructed image constitutes the 'frame' and interferences (reflections, ...).

2.5 Related Work on Eyeglasses Analysis

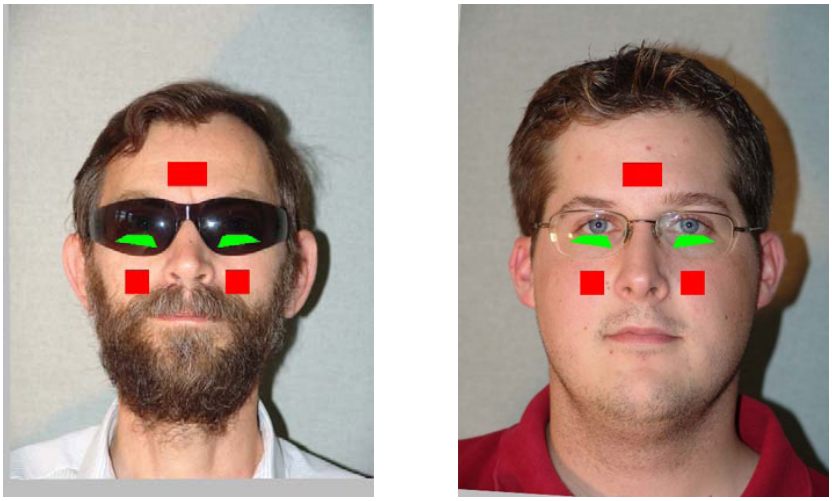
The previous work in eyeglasses analysis from the Advanced Computer Vision (ACV) and Siemens [20] is structured into four parts.

- Tinted glass detection
- Specularity detection
- Eye to frame distance measurement
- Frame width measurement

The following paragraph provides a short description of the used methods for each single part.

The '*tinted glass detection*' uses the color values of a predefined area within the eyeglasses, and compares the values of this area with the values of 'face areas'. After a transformation into the L*a*b color space and a calculation of the Euclidean distance, a decision criterion is derived. The problem with this approach is, that it requires the eyeglasses to be present within the predefined region, which is not mandated by the ISO standard (see Figure 2.2).

The '*specularity detection*' finds reflections on the eyeglasses. The approach is based on color space transformations and detecting specularities in a 'specularity channel' of the transformed color space. By combining the specularity channel with the saturation channel a region of this two-dimensional distribution is learned and used for classification.



(a) **(b)**

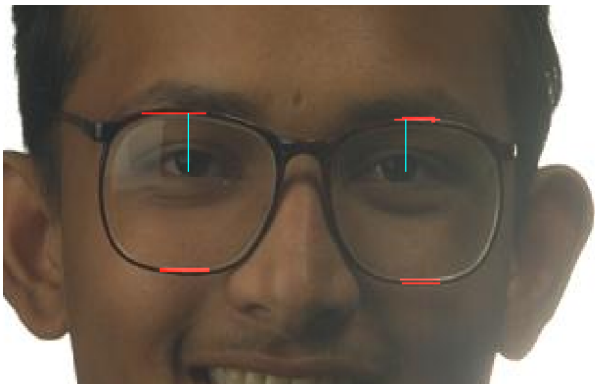
Figure 2.2: Fixed placement of the face (red) and eyeglasses (green) area.

The detailed approach on how specularities are detected would go beyond the scope of this thesis, thus in brief the specularity detection from the ACV and Siemens may be improved when the exact position of the eyeglasses is known.

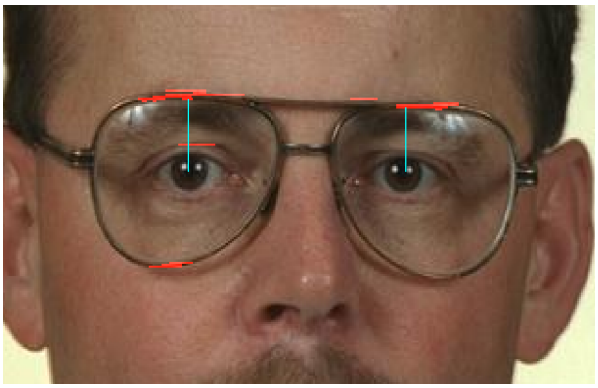
For '*Eye to frame distance measurement*' it is necessary to find the inner edge of the eyeglasses frame. Since the ICAO requirements forbid head rotation, the closed distance from the eye center to the frame can only be above and below the eye center. In this area the eyeglass frame is nearly horizontal. This allows computing the first and second derivation of angles and edges in this eye region. Applying a threshold to these derivatives finds and merges the closest 'edges' above and below the eyes.

The '*distance measurement*' works on a wide variety of types. Problems are mainly with frameless eyeglasses, where the 'frame' is hardly visible, and tinted eyeglasses, where the method cannot detect the edge of the frame properly (see figure 2.3)

The '*frame width measurement*' is structured in three steps: The first step finds candidate frame regions, where candidate regions are near the maxima of the absolute of the second derivative, and removes small regions. The second step analyzes the thickness in the candidate regions. The third step performs a statistical analysis of the



(a)



(b)

Figure 2.3: Distance measurement between eye center and closest eyeglasses frame. Images are taken from the Feret Faces database [18].

thickness distribution to yield the frame width. The results in figure 2.4 show, that the method works fine for a number of different eyeglasses. Negative detections result from reflections, bad image quality and eyeglasses frames, that contain edges.

(a)

(b)

Figure 2.4: Candidate frame regions derived with a filtered second derivative image.

Chapter 3

Implementation

Contents

3.1 Overview

The whole eyeglasses detection and analysis procedure is a complex task, which is structured into an eyeglasses detector and an eyeglasses segmentation algorithm (see Figure 3.1). The input data are normalised face images, which run through the steps shown in Figure 3.1.

The first step, the 'Eyeglasses Present Classification' (Chapter 3.2) requires a fast and exact detection of the presence of eyeglasses. Related work shows that approaches based on AdaBoost show excellent accuracy combined with fast execution times. The Viola Jones method, which was originally developed for face recognition, is a powerful and fast object detection algorithm, which can be trained to determine different object classes. Therefore a large set of annotated training data is needed. Sometimes other methods (Chapter 2.3) provide

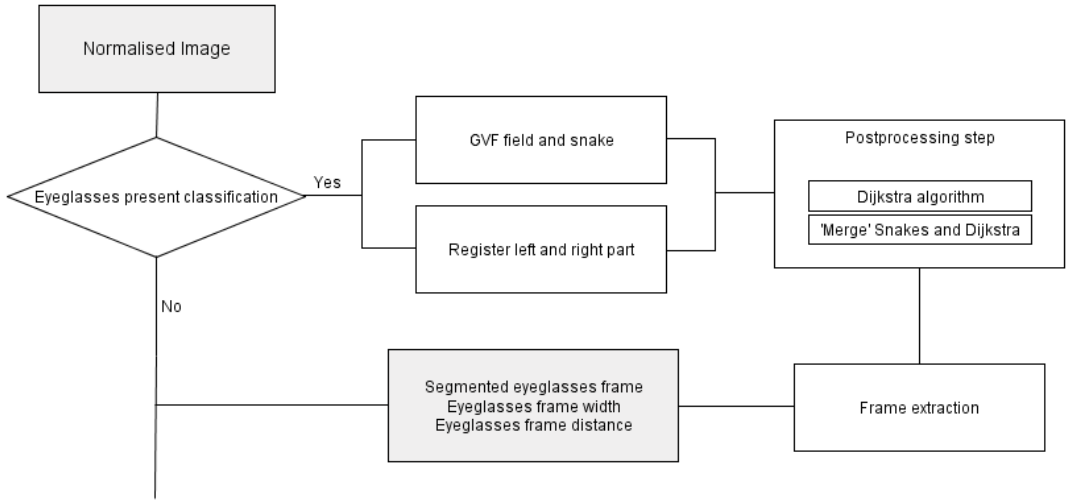


Figure 3.1: Overview of the developed methods.

a good classification rate, but the execution time is worse than the Viola Jones classifier.

There are several approaches in literature to localise eyeglasses (see Chapter 2.4), few of them work properly. A new approach for eyeglasses localisation is the Snake algorithm, which is a parametric curve to describe object boundaries. To get the exact contour internal parameters, which define the Snake properties (like elasticity, rigidity and viscosity), and external parameters, which result from the underlying image, get the Snake into the correct form. Therefore no training data is necessary and it allows to find all contours, which can be achieved with the chosen internal parameters.

Another possibility and an extension of Snakes is the 'Active Shape Model' (see [30]), which trains a mean shape and uses it in a localisation step to find the unknown object. The disadvantage of the 'Active Shape Model' (compared to Snakes) is that a large set of different and annotated eyeglasses is needed to get a suitable mean shape.

As it is possible to specify a rough geometric model with the internal parameters of the Snake, a restriction of eyeglasses types in the form of a training step is avoided and the classic Snake method is used. Section 3.3 introduces the proper eyeglasses segmentation utilizing a Snake algorithm (Chapter 3.3.1) to find the left and right eyeglasses in a facial image. Two independent Snakes are derived, which are compared and analysed in the next step.

To further improve the segmentation result an algorithm to register the left and right eyeglasses frame (Chapter 3.3.2) is necessary. Here a cost function with several elements is set up to compare the left and right half.

In the special eye-brow area the Snakes do not always provide exact solutions. There the shortest path algorithm from Dijkstra is implemented, which uses a weighted graph (Chapter 3.3.3). Graph-theoretic approaches for edge detection were already done by Falcao et al. [5] with their 'LiveWire and Live Lane' segmentation paradigms and Mortensen et al. [16] with the Intelligent Scissors concept.

Finally a comparison of the left and right Snake and the Dijkstra curve is done, where one final curve is found. Afterwards an exact frame extraction procedure is investigated (Chapter 3.3.5).

Due to the wide variety and different types of eyeglasses the chosen methods and algorithms get along with no previous knowledge. Except the 'Eyeglasses Present Classification' no substep requires a training step or annotated training data.

3.2 Eyeglasses Present Classification based on Viola-Jones Approach

The method presented in this thesis is based on 'Robust Real-Time Face Detection' from Viola and Jones, due to its good performance (real-time) and its high accuracy.

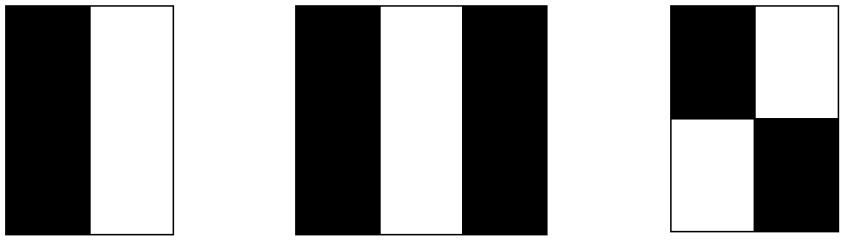
3.2.1 'Robust Real-Time Face Detection'

Viola and Jones developed a face detection algorithm that uses a boosted cascade of simple features. The aim of this algorithm is to detect faces in real-time and with high accuracy.

The algorithm has two phases - the learning phase and detection phase. The learning phase trains the classifier on two sets of images. One including and one excluding the searched object.

The algorithm works with three types of features - a two-rectangle feature, a three-rectangle feature and a four-rectangle feature. The computational complexity of these

three features is low, since each feature is the difference of the sum between two rectangular regions (see Figure 3.2).



(a) **(b)** **(c)**

Figure 3.2: Three different rectangle features (black area is subtracted from the white area).

With the detector resolution of 24 x 24 pixel, this method yields a very large set of features ($\tilde{1}$70.000). To rapidly compute this set of rectangle features, Viola and Jones use an integral image.

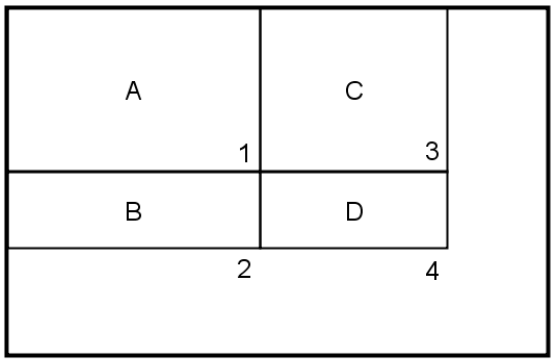


Figure 3.3: Integral image

The integral image says, that the value of the integral sum in the area A is stored in point 1, the integral of A and B is stored in point 2 etc. The integral value of rectangle D is calculated by adding point 1 and 4 and subtracting point 2 and 3 (see 3.3).

After calculating the single features the classifier training starts, where each feature represents a classifier. The classifier with the lowermost error gets a weight and is combined with the previously used classifiers. In the end the training provides one strong classifier, which consists of weighted weak classifiers. The way to find a final classifier is called 'AdaBoosting'.

The advantage of the 'AdaBoosting' is the detection phase and its cascade system. The strong classifier, which consists of weak classifiers, is applied to a sub-window. The first weak classifier gives a first decision. If it returns a positive answer, the next weak classifier is used. If one weak classifier rejects, the whole sub-window will be rejected, thus resulting in high detection performance (see Figure 3.4).

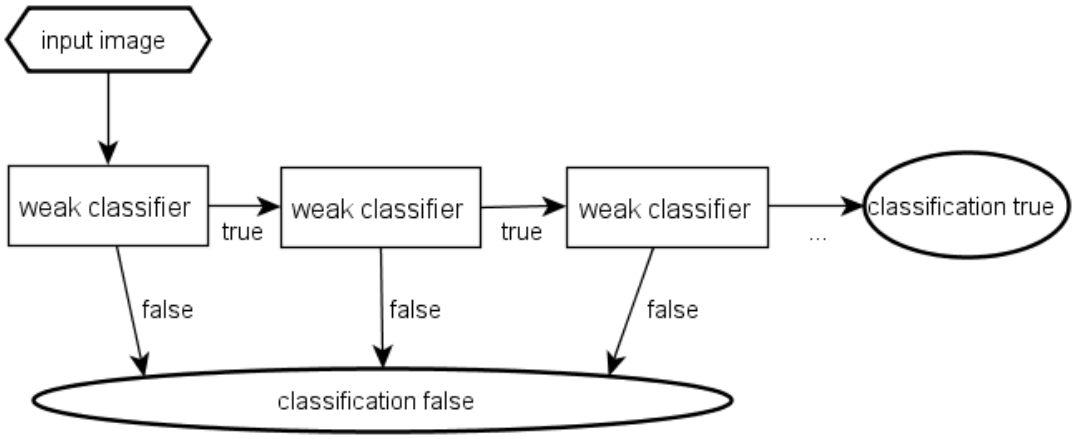


Figure 3.4: Detection cascade of the Viola Jones detector.

3.2.2 Eyeglasses Detection

The basis for the presented eyeglasses detection is the Viola Jones method explained in Chapter 3.2.1. Instead of face patches, images with eyeglasses (see Figure 3.5) and without eyeglasses (see Figure 3.6) are used here. The patches are of 240x120 pixel in size and are extracted from the left and right eye of a face portrait image.

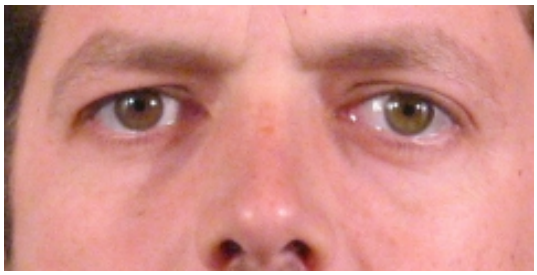
The classifier requires a training step to determine weak classifiers, which combine to one strong classifier.

(a)

(b)

Figure 3.5: 'Glasses patches' for the Viola-Jones algorithm.

(a)



(b)

Figure 3.6: Patches without the object for the Viola-Jones algorithm.

3.3 Eyeglasses Segmentation

This chapter shows how to segment eyeglasses from normalised face portrait images. The segmentation is a procedure with five parts. In chapter Chapter 3.3.1 the main localisation is done with a Snake and an underlying Gradient Vector Flow field. Furthermore a registration process of the left and right eyeglasses (Chapter 3.3.2) part allows to use the symmetric feature of eyeglasses. In the eyebrows area the shortest path algorithm of Dijkstra is implemented to improve the Snake results (Chapter 3.3.3). After comparing the left-eye and right-eye Snake (Chapter 3.3.4), the exact eyeglasses frame is extracted (Chapter 3.3.5) and the frame-width and frame-distance are derived.

3.3.1 Find Eyeglasses with Snakes

The presented method is based on the paper 'Snakes: Active contour models' [13] and 'Gradient Vector Flow: A New External Force for Snakes' [32]. In these papers a parametric Snake model is used to locate object boundaries. In [32] Xu et al. use an advanced external force for Snakes, they replace the traditional potential field with a new Gradient Vector Flow field.

3.3.1.1 Snakes

Snakes are curves of the form

$$\mathbf{x}(\mathbf{s}) = [x(s), y(s)], s \in [0, 1]; \tag{3.1}$$

These curves move through the image and try to minimize an energy function.

$$\int_0^1 \frac{1}{2}(\alpha|\mathbf{x}'(\mathbf{s})|^2 + \beta|\mathbf{x}''(\mathbf{s})|^2) + E_{\text{ext}}(\mathbf{x}(\mathbf{s}))ds \tag{3.2}$$

The internal and external forces constitute the energy function. The internal force describes the tension and rigidity of the Snake. α and β are weighting parameters of the internal force. The external force is derived from the underlying image and is also called the potential field and represents the domain, in which the Snake moves.

Typically the Snake should draw towards the edges in an image. Therefore the external force is based on the gradient of the image.

The algorithm applies a blurring filter before computing the potential field. In this case the boundaries and edges in the image become smoother. This results in the positive effect that the 'range' of the Snake increases.

To get the minimum of the energy function (3.2), one has to form its derivative and set it to zero, thus leading to the Euler equation.

$$\alpha \mathbf{x}''(\mathbf{s}) - \beta \mathbf{x}''''(\mathbf{s}) - \nabla E_{\text{ext}} = 0 \tag{3.3}$$

To solve this Euler equation, an iterative method is necessary. The function will be seen as a function of time t and the parameterisation on s, i.e. (s,t) [32, page 1-2].

In their paper Xu and Prince replace the classic edge-based potential field with a Gradient Vector Flow (GVF) field $\mathbf{v}(\mathbf{x}, \mathbf{y}) = (\mathbf{u}(\mathbf{x}, \mathbf{y}), \mathbf{v}(\mathbf{x}, \mathbf{y}))$ and the normal Snake with a GVF Snake.

The goal of the Gradient Vector Flow field is to increase the 'streamlines' of the vector field. Especially in homogeneous regions, where the gradients of the edge map $f(x, y)$ are zero, information from the boundaries is utilized to find a reasonable result in the homogeneous parts of the image.

In the Snake equation the potential force $-\nabla E_{\text{ext}}$ is replaced by $\mathbf{v}(\mathbf{x}, \mathbf{y})$. Again the solution is found in an iterative manner.

$$\alpha \mathbf{x_t}(\mathbf{s}, \mathbf{t}) = \alpha \mathbf{x}''(\mathbf{s}, \mathbf{t}) - \beta \mathbf{x}''''(\mathbf{s}, \mathbf{t}) + \mathbf{v} \tag{3.4}$$

As an advantage the GVF field increases the 'range' of the Snake, thus the GVF Snake is also able to move into concave regions of the object. This allows the Snake to find and capture the edge from both sides of the boundary. [32, pg. 3], [33]

The images in Figure 3.7 and 3.8 are examples for a normal potential field and a Gradient Vector Flow field.

In a further step Xu and Prince extend the internal forces of the Snake with a

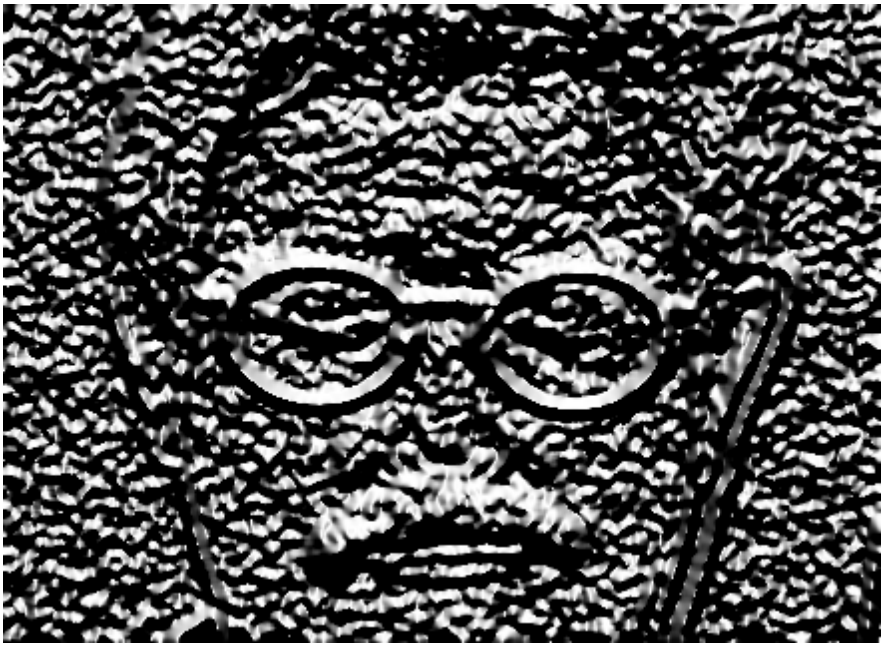


Figure 3.7: Example of a normal potential vector flow field.

Figure 3.8: Example of a Gradient Vector Flow field.

pressure force. The pressure force prevents the Snake from decreasing its size. Thus the Snake moving through the vector field tends to grow rather than to shrink.

3.3.1.2 Snakes and Eyeglasses

To yield a good result with Snakes it is necessary to have some a-priori knowledge. The closer the initial Snake to the 'real' glasses is, the better the final solution will be. For that reason the eyeglasses are assumed to occlude the left and the right eyes. In general the eyeglasses shape is more a ellipse than a circle, that's why the initial Snake is a small ellipse, which is centered in the eye and should be smaller than the eyeglasses (see Figure 3.9 and in [33]). In this case the position of the ellipse depends on the center of the left and right eye. As the input data are normalised face images (see Section 2.1), the eye center coordinates for the Snake initialisation are fixed values (see Table 2.1).

Figure 3.9: Face image with initial Snake.

To utilize Snakes to find glasses, it is first necessary to compute an edge map of the image using a Laplacian filter (3.10a). This edge map is blurred with a Gaussian filter (3.10b). The Gaussian filter removes residual interferences and should point out the important edges of the eyeglasses.

This algorithm detects the position of eyeglasses well. Problems occur with frameless and barely framed eyeglasses. Neglecting further pre-processing steps avoids potential data loss.

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \qquad \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

(a) 2D Laplace filter

(b) Gaussian filter $g(x, \sigma) = e^{-\frac{x^2}{2*\sigma^2}}$

Figure 3.10: Filters used.

The algorithm computes the Gradient Vector Flow field with an iteration number of 80 using the blurred edge map (see Figure 3.8). After these 80 iteration steps the internal and external forces yield to the correct position of the Snake (see Figure 3.11). The Snake method provides two different Snakes - one for the left and one for the right eye. Thus, the algorithm computes the Snakes completely independently, which means that the form, size and properties need not be identical.

Figure 3.11: Face image with found Snake.

3.3.2 Register left and right Eyeglasses

So far the algorithm dealt with the left and right eyeframe independently. In order to check the correctness and to give a reliability of the detection, it is necessary to compare the left and the right Snake. Therefore a few preparation steps are essential. Normalising the face portrait results in a symmetric left and right eye (see [8]).

It is uncertain that the left and right parts of the eyeglasses frame are always exactly symmetric to the y-axis. Thus, to compare the left and right Snake, the left and right framepart need to be registered (see Figure 3.14a).

This process takes a small patch of the whole image that contains the eyeglasses only. To find a correct registration, the parameters of the three degrees of freedom of the rigid transformation have to be determined. Due to the normalisation of the face images the translation in y direction can be neglected, which reduces the three degress of freedom to two (rotation ϕ and translation x).

Figure 3.12 shows three examples with different free parameters, where the right eye part is subtracted from the left eye area.

Under the assumption that the previous eye normalisation worked well and the eyeglasses are aligned to the horizontal and vertical axis of the face, our experiments showed that the range of translation and rotation goes from -10 pixels to 10 pixels and -2.5 to 2.5 degrees, respectively.

This registration uses a 2-dimensional cost function. The cost function contains weighted features of the patch, which can be easily calculated using the following image features:

- intensity value $P(x, y)$
- edge map (Sobel operator) $Q(x, y)$
- gradient direction $G(x, y)$

Figure 3.13 presents results from this cost function with all possible combinations of transformation and rotation in the predefined ranges.

The lowermost value of the cost-function yields the best registration results. We find the minimum by an exhaustive search over the translation and rotation range with a discretisation of the cost function in steps of 0.5 degree and 1 pixel.

(a) rot = -2.5, trans = -8 px **(b)** rot = 1.0, trans = 6 px

(c) rot = 1.5, trans = 0 px **(d)** rot = 1.0, trans = -4 px

Figure 3.12: Registration process with different free parameters.

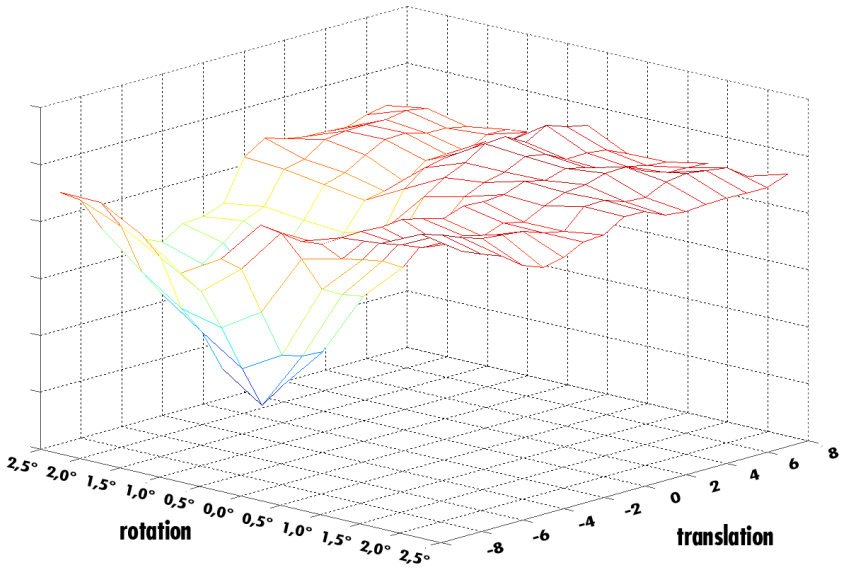
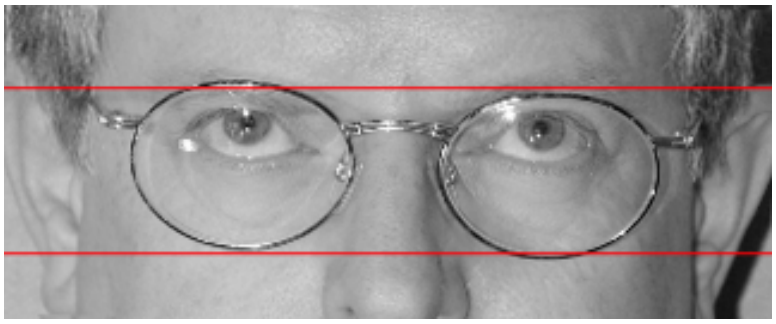
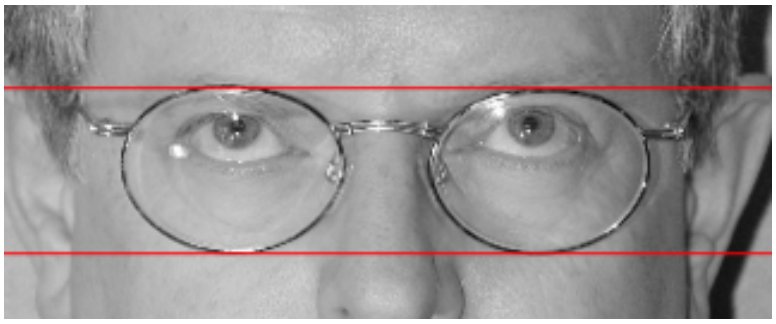


Figure 3.13: Visualisation of the registration cost function. The minimum shows the optimal alignment.



(a) Eyeglasses area BEFORE registration.



(b) Eyeglasses area AFTER registration.

Figure 3.14: Registration of the eyeglasses area.

The algorithm adapts the face image and Snakes, according to the rotation and translation values, to compare the left with the right Snake (see Figure 3.14).

3.3.3 Dijkstra Algorithm for Eyebrows Area

The area near the eyebrows proves itself to be tricky. Often the eye frame 'disappears' in the eyebrows, the eyebrows cross the frame etc (see Figure 3.15, 3.16). For that reason a Snake based approach sometimes has problems to find the correct edge at the upper frame of the eyeglasses. Whether the curve part retrieved by the Dijkstra algorithm is used or rejected is decided in Section 3.3.4.

Figure 3.15: Face image where eyeglasses frame overlaps eyebrows (example 1). Image taken from the Feret Faces database [18].

It is not only here, in image analysis there are often situations, when automatic techniques fail or don't provide adequate results. Therefore graph-theoretic approaches for edge detection were already done by Falcao et al. [5] with their 'LiveWire and Live Lane' segmentation paradigms and Mortensen et al. [16] with the Intelligent Scissors concept.

That's why in this area the Snake is supported by a shortest-path-algorithm to find a correct solution. A graph $G = (V, W)$ with the edge weights $W \in \mathbf{R}+$ and $v \in V$ is given. The nodes v represent the image pixels and the edge weights W are derived from the underlying image and a cost function. Solving the shortest-path problem is

Figure 3.16: Face image where eyeglasses frame overlaps eyebrows (example 2). Image taken from Feret Faces database [18].

equivalent to finding the path that minimizes the cost function.

This can be done by the Dijkstra algorithm (see [2]), which needs an edge-based 'graph' with non negative values. With a given start $s \in V$ and defined end point $e \in V$ the algorithm finds the shortest path $w, v_{1..n}, e$ in G, where $n \in \mathbf{N}$ and $v \in V$. The 'shortest' path features the smallest resulting sum of weighted edges.

In the previous previous we showed the registration of the left and the right part of the eyeglasses. The computed graph for the Dijkstra algorithm is a combination of the left and right part of the eyeglasses.

3.3.3.1 Edge-weighted Graph

The algorithm uses a 2-dimensional image to construct the edge-weighted graph. Every pixel in this image represents a node in the graph and every node has 8 neighbors. That means that the graph edges are simply the connections between a pixel and its 8 neighbors.

The shortest path should ideally represent the edge of the eyeglasses. That is why the used features are gradient features, to strengthen the edge properties of the image. The algorithm uses the following features for the cost function

- intensity value at pixel p, $f_{\mathrm{I}}(p)$
- zero-crossing at pixel p, $f_{\mathrm{Z}}(p)$
- gradient value at pixel p, $f_{\mathrm{G}}(p)$
- gradient direction at pixel p and q, $f_{\mathrm{D}}(p,q)$

The cost function $l(p,q)$ can be computed like this

$$l(p,q) = w_{\mathrm{I}} f_{\mathrm{I}}(p) + w_{\mathrm{Z}} f_{\mathrm{Z}}(p) + w_{\mathrm{G}} f_{\mathrm{G}}(p) + w_{\mathrm{D}} f_{\mathrm{D}}(p,q) \tag{3.5}$$

As can be seen, the cost function $l(p,q)$ consists of two basic elements. The intensity, zero-crossing and gradient value can be computed independently, while the gradient direction needs the information of the 'neighbors'.

The weights of the cost function here are derived empirical ($w_I = 0.05, w_Z = 0.15, w_G = 0.8, w_D = 0.2$).

3.3.3.2 Dijkstra Algorithm

The Dijkstra algorithm takes a graph with nodes and connections between nodes. Based on a starting point it computes the shortest path to the end node.

```
1 // variables
2 L is an active nodes list
3 C is a cost_map with size(L)
4 P is a list with the parent of each node
5 S is a status map with size(L), where the status can be
      (notprocessed, processed, visited)
6
7 // initialization
8 C = infinite
9 C(s) = 0   // cost map is 0 at startpoint
```

```
P(s) = 0   // start point has no parent
S = notprocessed
L = s;

// main loop
while L != empty
  find node p with minimum in cost map
  remove node p from L
  S(p) = processed

  for each neighbor q
     if S(q) != processed
       cost = C(p) + l(p,q)
       if S(q) == visited AND cost < C(q)
         remove node q from L
         S(q) = notprocessed
       end
       if S(q) == notprocessed
         C(q) = cost
         P(q) = p
         S(q) = visited
         insert node q in L
       end
     end
  end for
end while
```

Listing 3.1: Calculate the shortest path to each node.

When the Dijkstra algorithm is finished, it is straight forward to find the path from the end point to the start point.

```
// variables
C is the cost_map with size(L)
P is the list with the parent of each node

// initialization
q = e
path = []
cost = 0

while P(q) != 0
  cost = cost + C(q)
  path = [q, path]
  q = P(q)
end while
```

Listing 3.2: Reconstruct path from start to end point

3.3.3.3 Dijkstra algorithm in the Eyebrows Area

Figure 3.17 shows the result of the Snake method. The left Snake is too big and includes the eyebrows area.

Figure 3.17: Face image with detected Snake (red). Image is taken from the Feret Face database [18].

Therefore the Dijkstra shortest path algorithm is computed for this special frame piece. This part of the frame is fixed and marked with two vertical bars in figure 3.17. As starting and end point of the Dijkstra method the Snake points at the vertical bars are chosen.

3.3.3.4 Results

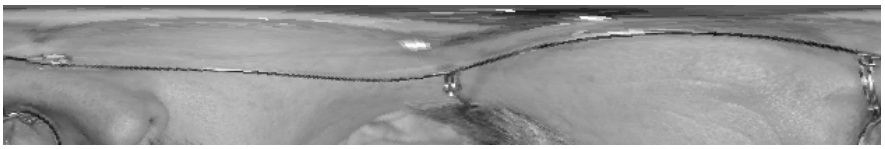
The image (Figure 3.18) below shows the qualitative results of the Dijkstra algorithm. In Section 3.3.4, where the postprocessing step is described, a comparison of the left and right Snake and the Dijkstra curve is performed. Thereby a detailed analysis of the found curves is done and the best solution regarding the symmetry of the left and right eyeglasses is determined.

Figure 3.18: Face image with detected Snake (red) and computed Dijkstra curve (blue). Image is taken from the Feret Face database [18].

3.3.4 'Merge' the left-eye and right-eye Snake with Dijkstra curve

Transforming the two Snakes into polar coordinates allows comparing and analyzing the face image. This step requires two different transformations, one for the left and right eye each. Each transformation moves the center of the coordinate system to the left/right eye center and translates all points from the image within a radius of 100 pixels into the new coordinate system.

(a) Original image with eye centers.



(b) Transformation of the left eye area.

(c) Transformation of the right eye area.

Figure 3.19: Transformation of the left and right eye areas.

Figures 3.19b and 3.19c show that the Snakes are not a 'circle' anymore. The new curves are located in a transformed 'eye-image' of the size 629x100 pixels. A detailed analysis separates the curves into four parts - horizontal part of the frame at the top, horizontal part at the bottom, vertical parts on the left and right side of the eyeglasses frame (see Figure 3.20).

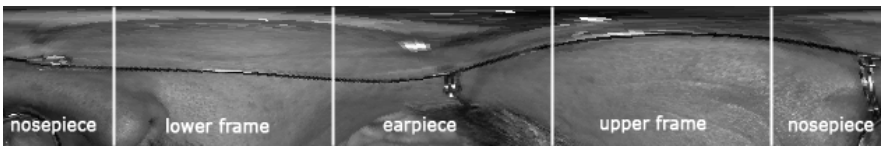


Figure 3.20: Transformed eyeglasses parts - nosepiece, upper and lower frame, earpiece.

First it is checked if the curves are in a 'restricted area'. Due to all kind of eyeglasses having a similar form, there are a few parts in the transformed image, where the eyeglasses can not be located (see Figure 3.21).

Figure 3.21: 'Restricted area'

To compare the parts of the Snake and the Dijkstra curve, a cost function $l(x, y)$ is set-up, which is derived from the transformed 'eye-images'. The cost function consists of several weighted features, which are determined empirically.

- intensity value $f_I(x, y)$
- zero-crossing $f_Z(x, y)$
- gradient value $f_G(x, y)$

The cost function $l(x, y)$ provides a 'reliability' value for a chosen curve and can be computed like this

$$l(x, y) = w_I * f_I(x, y) + w_Z * f_Z(x, y) + w_G * f_G(x, y) \tag{3.6}$$

The following comparison is done for every eyeglasses part (as in Figure 3.20). First the left and right Snake parts are subtracted and the variance and maximum error distance are determined. If the variance is lower than 4.0 pixel and the maximum distance error is 5.0 pixel, the average of the two curves is computed. If not, the cost

function for the two curves is calculated and a 'realiability' value is retrieved. The curve part with the higher 'reliability' is chosen.

In the upper frame area, where the eyebrows area is situated, the two Snake parts are compared additionally with the found Dijkstra curve, which is computed in Section 3.3.3. If all of the three curves are not similar - see above -, the one with the highest 'reliability' is taken. The cost function applied on the whole curve describes the correctness probability of the localisation.

The final result is one single curve, which is valid for the left as well as for the right eyeglasses part (see Figure 3.22). The curve may not be continuous at the linking points, wherefore the area around the linking points is replaced with the median. Furthermore if the curve is a little bit noisy, a smoothing filter is applied. The final curve represents the eyeglasses localisation for the left and right half.

Figure 3.22: 'Merged' Snakes and Dijkstra curves.

3.3.5 Frame Extraction

The last part of the whole segmentation process is the exact detection of the eyeglasses frame, which normally has an inner and outer border (except frameless eyeglasses). An eyeglasses frame is a complex part, which is not easy to detect. Especially the ear- and nosepiece of the eyeglasses have completely different appearances and no underlying basic structure. As a consequence these parts cannot always be detected exactly and properly.

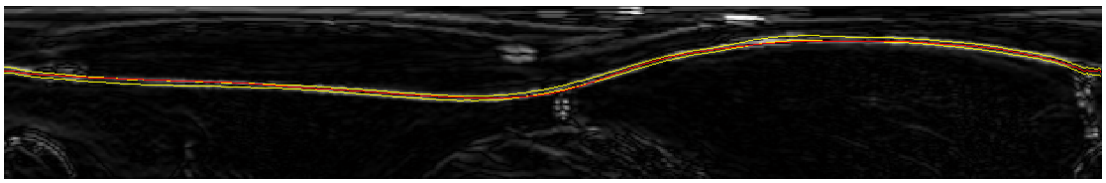


Figure 3.23: Exact eyeglasses frame edges (yellow) and transformed Snake (red).

The algorithm uses the two transformed presentations of the eye areas to get the exact eyeglasses frame. The used a-priori knowledges are that the found curve (see Section 3.3.4) is located on the eyeglasses frame and that the frame thickness is almost steady.

First the horizontal gradient images of the transformed eye-images are computed. The strongest edges near the localisation curve are connected to an upper and lower frame curve regarding the thickness. Sometimes it happens, that the localisation curve is not exactly on the frame. In these cases a variance of 2 pixels is permitted.

A further step removes peaks in the upper and lower curve and determines if the two found frame edges correspond to each other. The resulting curves should be smooth curves without peaks.

As a result of the two frame edges the average frame distance for the upper and lower frame part can be derived (see Figure 3.20). The final step transforms the inner and outer edge back into the original space (see Figure 3.24).

Figure 3.24: Final segmentation of eyeglasses frame.

3.4 Chapter Summary

In this chapter our algorithms for 'Eyeglasses Present Classification' and 'Eyeglasses Segmentation' were described. The first part dealt with the classification. As shown in Chapter 2.3 the boosting classifiers provide better results than comparable machine learning algorithms. That's why our used approach is based on the widely used and powerful AdaBoosting algorithm from Viola and Jones. The Boosting is combined with different types of features and leads to results, which can be found in Chapter 4.1.

The second part of the Chapter dealt with the Eyeglasses Segmentation, which is divided into several parts. Due to the wide variety of eyeglasses a method without previous knowledge in form of training steps is avoided. The extended Gradient Vector Flow field Snake leads to a first basic localisation of the left and right eyeglasses. After a registration process of the two eyeglasses halves, the Dijkstra algorithm optimizes the found Snake in the eyebrows area. Then the curves for the left and right part are compared and a final solution is found. This solution is evaluated with 100 manual annotated eyeglasses (see Chapter 4.2). Beside that two eyeglasses parameter - frame width and frame distance between eye center and frame - are evaluated with 170 manually marked images.

Chapter 4

Experiments and Results

Contents

In this section we are going to describe the evaluation setup experiments we performed using our developed algorithms.

4.1 Eyeglasses Present Classification

4.1.1 Training Step

In the training phase 1200 eyeglasses patches (600 with eyeglasses and 600 without eyeglasses) are used to compute the classifier (see Figures 3.5 and 3.6). The patches have a size of 240x120 pixels and are derived from normalised face portrait images (see Figure 4.1). The training step yields to 157 weak classifiers, which combine to one strong classifier. Some examples for the determined sub-windows are shown in Figure 4.2.

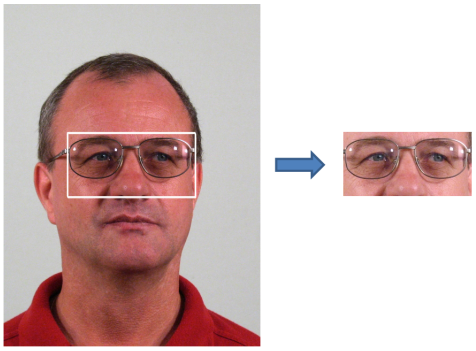
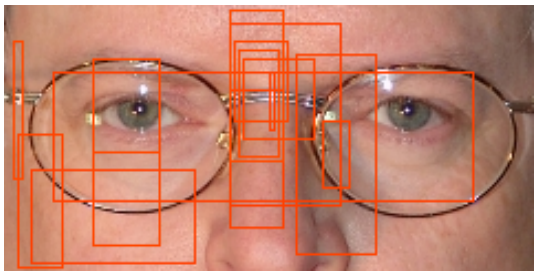
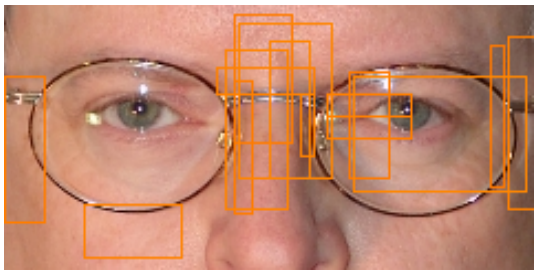


Figure 4.1: Eyeglasses patches for the Eyeglasses Present Classification.



(a)



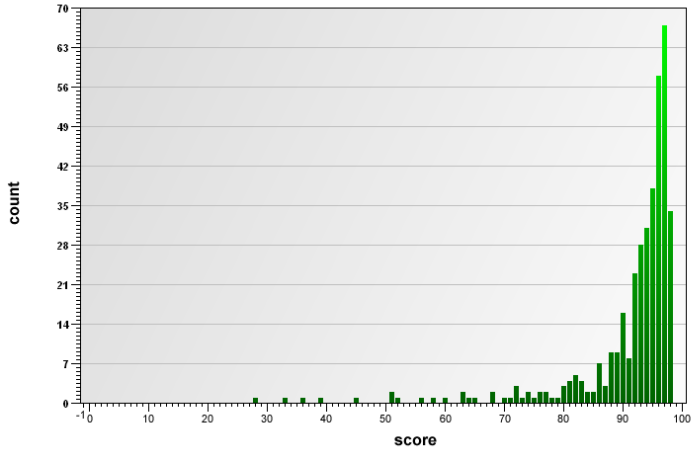
(b)

Figure 4.2: Weak classifiers samples.

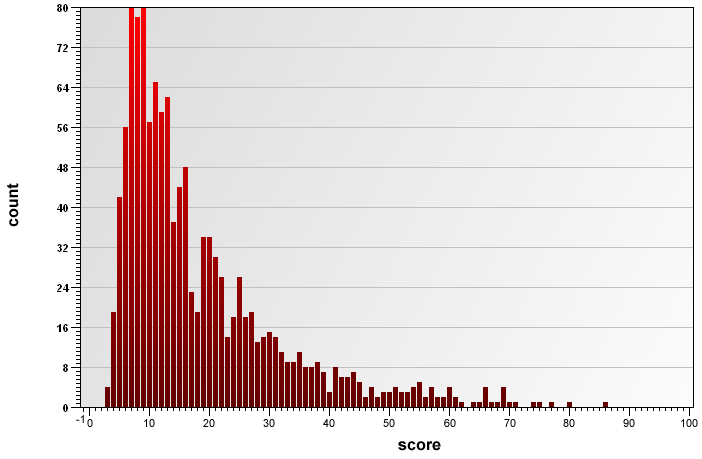
4.1.2 Classification Step

For the evaluation of the 'Glasses Present Classification' algorithm four different databases of annotated face images are used. The first database contains 1607 images, the second one has 1576 normalised images, the third one 412 and the last one 1333 face images. The data sets were provided by Siemens and from different sources.

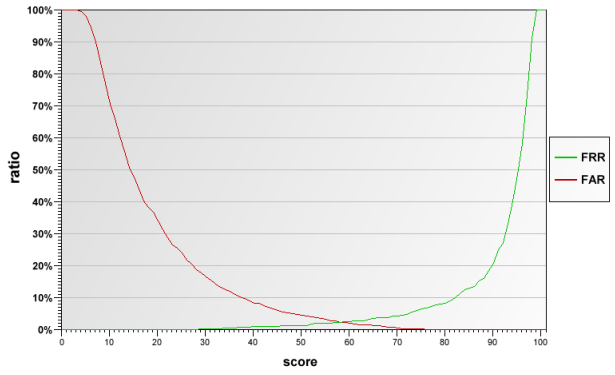
During the evalution process, each image returns a parameter, which gives the probability of the decision whether the patch contains eyeglasses or not. It's range is from 0 to 100.



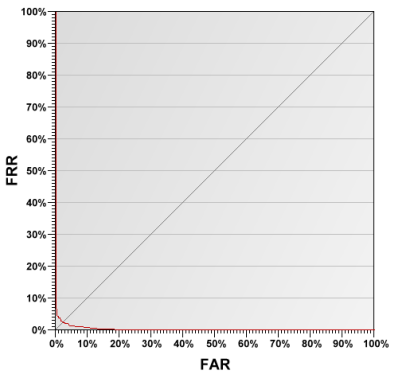
(a) Compliant scores.



(b) Non-compliant scores.

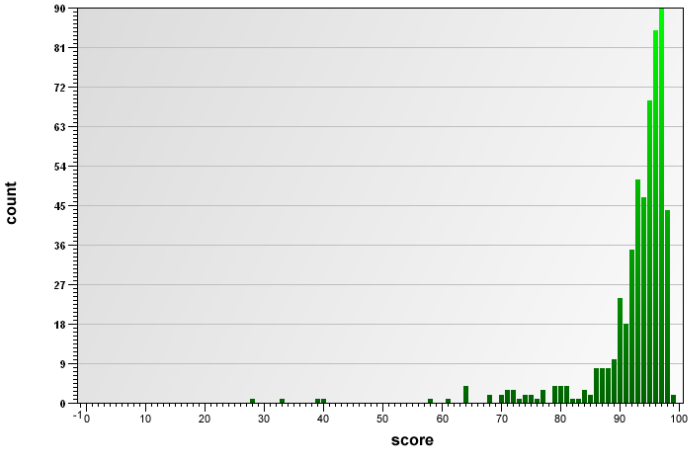


(c) FAR/FRR distribution.

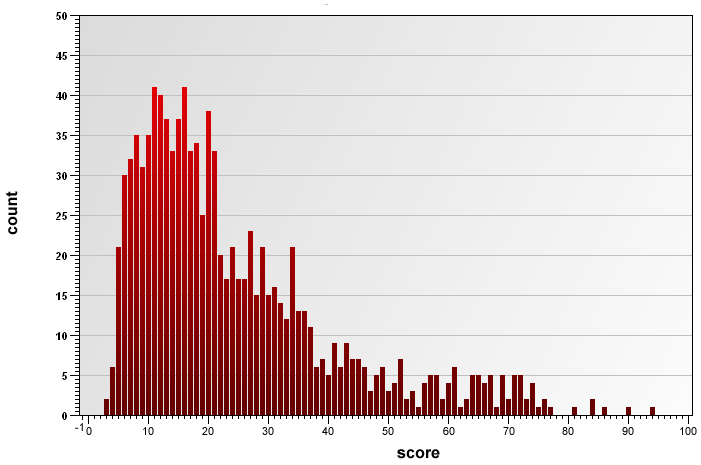


(d) ROC curve.

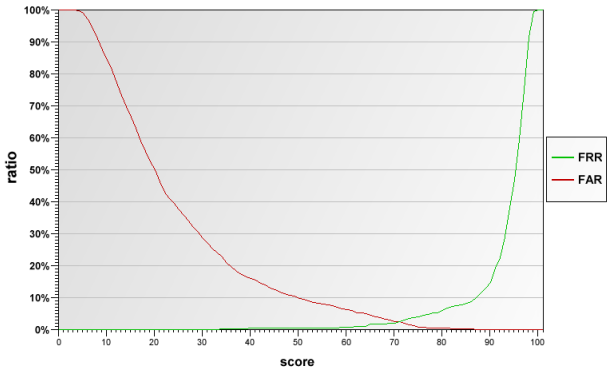
Figure 4.3: Evalutions on database 1 (1607 images).



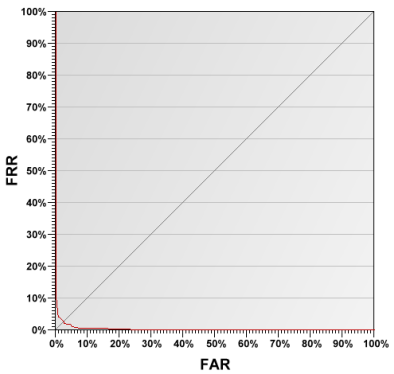
(a) Compliant scores.



(b) Non-compliant scores.

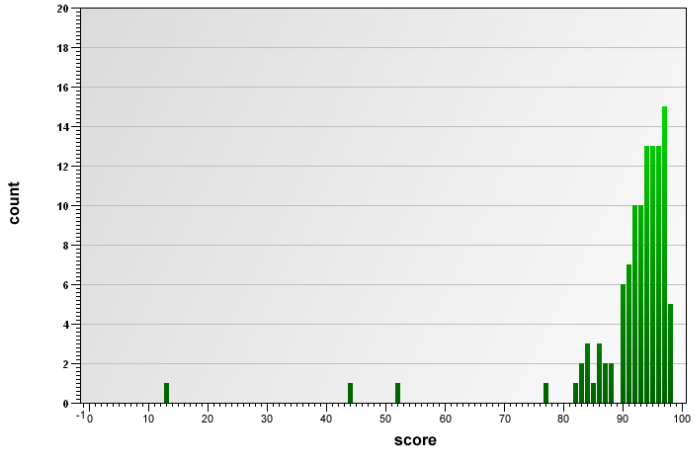


(c) FAR/FRR distribution.

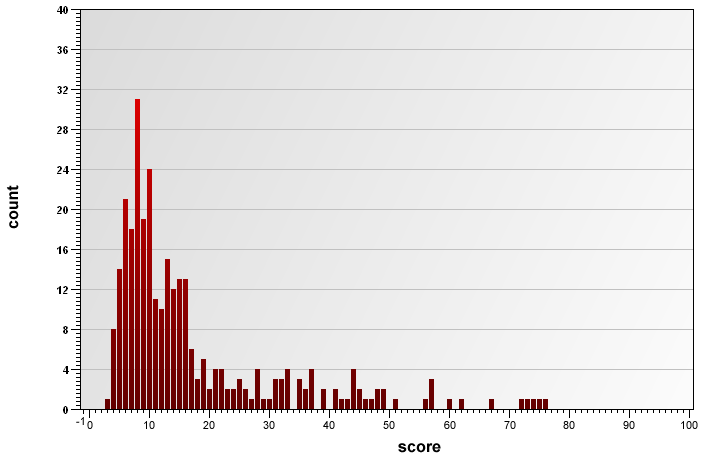


(d) ROC curve.

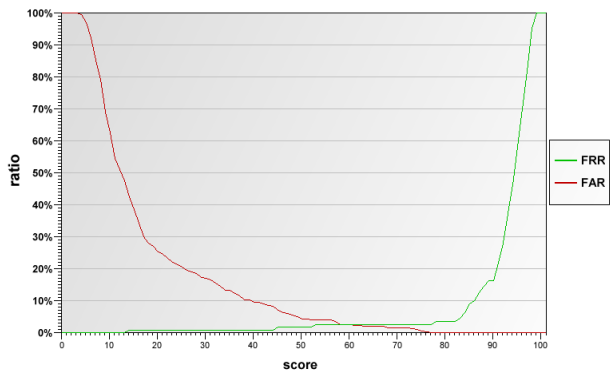
Figure 4.4: Evalutions on database 2 (1576 images).



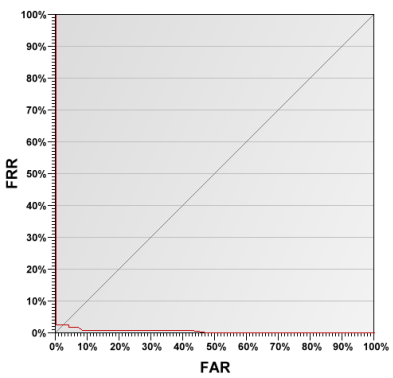
(a) Compliant scores.



(b) Non-compliant scores.

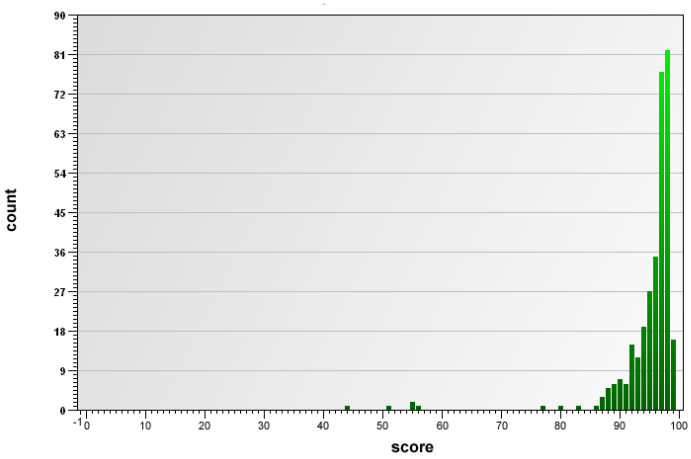


(c) FAR/FRR distribution.

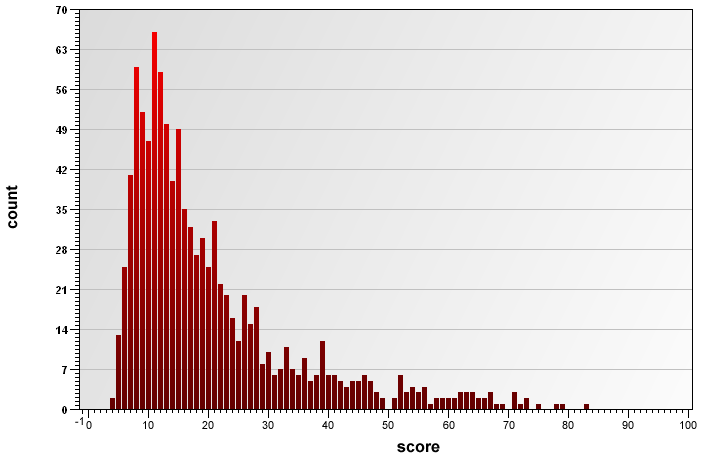


(d) ROC curve.

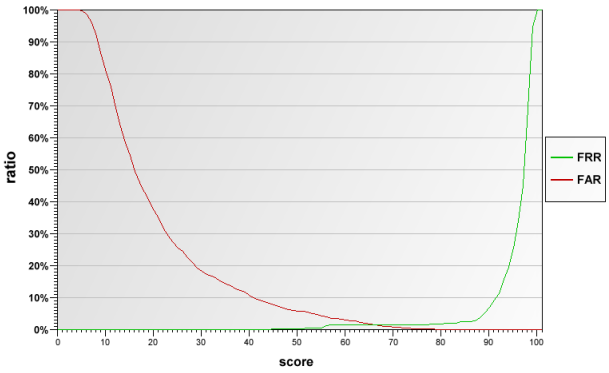
Figure 4.5: Evalutions on database 3 (412 images).



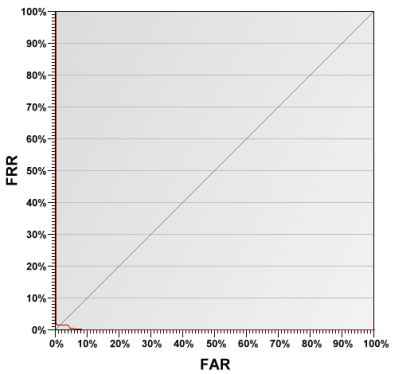
(a) Compliant scores.



(b) Non-compliant scores.



(c) FAR/FRR distribution.



(d) ROC curve.

Figure 4.6: Evalutions on database 4 (1333 images).

The final and important results are shown in the table below.

	EER (FAR/FRR)	EER score
Database 1 (1067 images)	2.0 %	58
Database 2 (1576 images)	3.0 %	71
Database 3 (412 images)	2.0 %	61
Database 4 (1333 images)	1.5 %	66

Overall the extensive evalutions of the 'Glasses present' method, shows that it works well on various different databases. On all databases the maximum error of wrong detection is at 3 %.

4.1.3 Discussion

All in all nearly 5000 facial images were used during the evaluation process. The result of this detector is very satisfying. On independent databases the classifier has a maximum error of 3 % and a mean error of 1.9 %. The required 99 % correctness for an independent test data set is not achieved.

The question is, how to increase performance even further. One possible attempt would be to train the classificator with more different databases, because every single has its own special patterns. In this case the classifier was trained with the Siemens and Feret ([18]) database, which maybe has problems with different skin colors, different eye-region characteristics, etc.

4.2 Eyeglasses Segmentation

The evaluation on eyeglasses segmentation is divided into two sections. In 4.2.1 several examples of a segmentation are shown and in 4.2.2 a comparison with manually annotated eyeglasses is done.

4.2.1 Examples for Eyeglasses Segmentation

The following pictures are examples, how a good and how a wrong segmentation of the eyeglasses can look like. Figure 4.7, 4.8, 4.10, 4.11, 4.12 and 4.13 show typical face images with 'normal' eyeglasses, where the segmentation worked well. Figures 4.9 and 4.14 contain correct detected eyeglasses where the lower frame is nearly invisible. Figures 4.10 and 4.12 show face images out of the Feret database, which have a darker skin color, but the eyeglasses can be segmented correctly. In Figure 4.15 the algorithmus detected that the eyeglasses are frameless (red line), whereas the eyeglasses in Example 13 could not be detected correctly. The same happenend in Figure 4.16, where the frame did not find the 'eyeglasses'.

Figure 4.7: Eyeglasses segmentation example 1.

Figure 4.8: Eyeglasses segmentation example 2.

Figure 4.9: Eyeglasses segmentation example 3.

Figure 4.10: Eyeglasses segmentation example 4. Image is taken from the Feret Faces database [18].

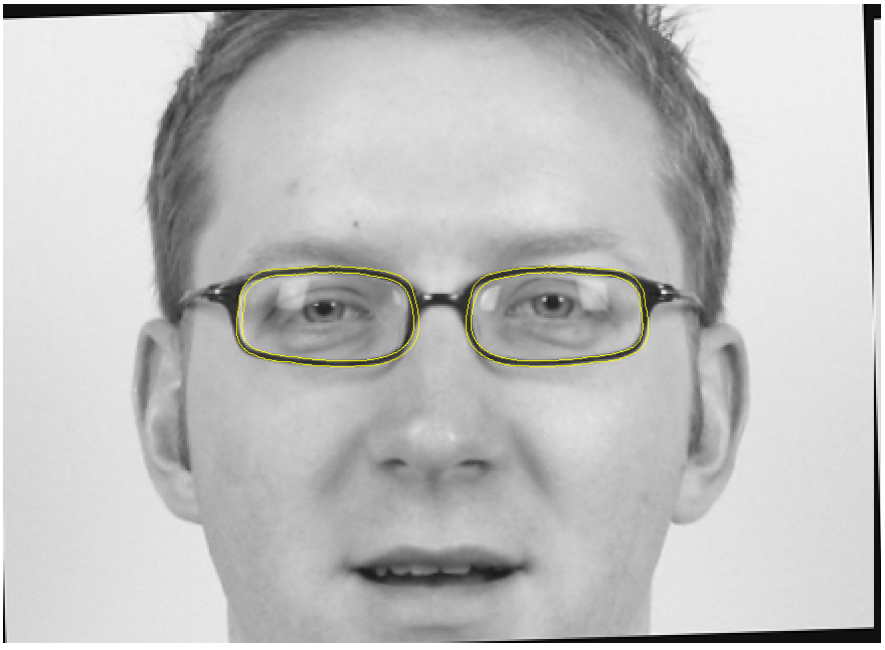


Figure 4.11: Eyeglasses segmentation example 5.

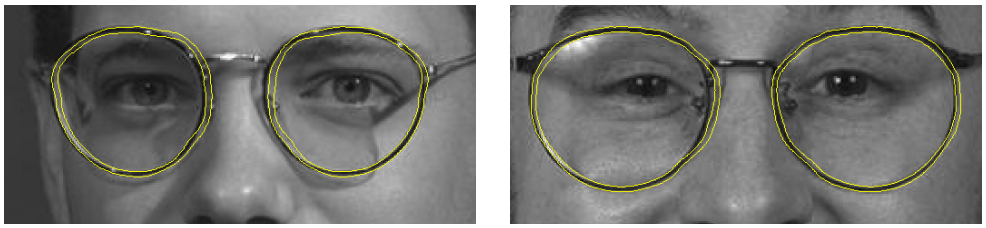


Figure 4.12: Eyeglasses segmentation example 6 and 7. Images are taken from the Feret Faces database [18].

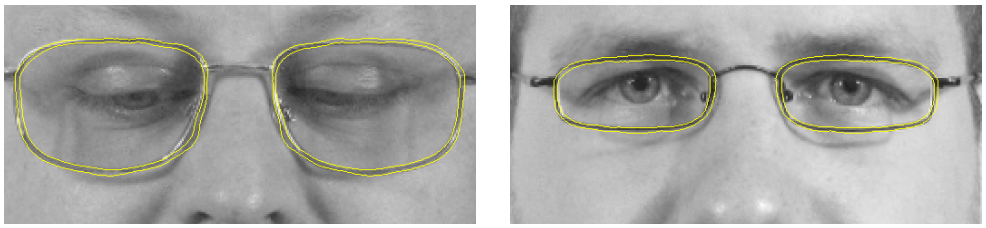


Figure 4.13: Eyeglasses segmentation example 8 and 9.

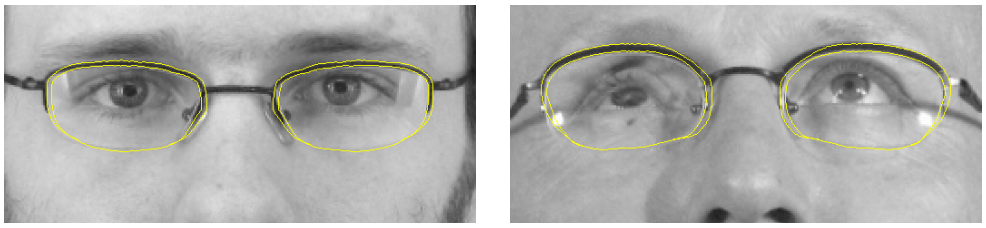


Figure 4.14: Eyeglasses segmentation example 10 and 11.

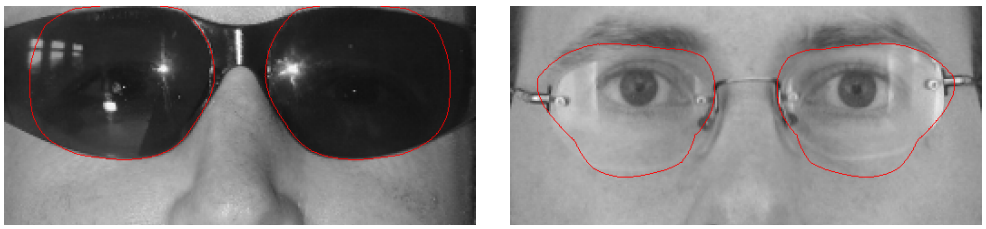


Figure 4.15: Eyeglasses segmentation example 12 and 13.

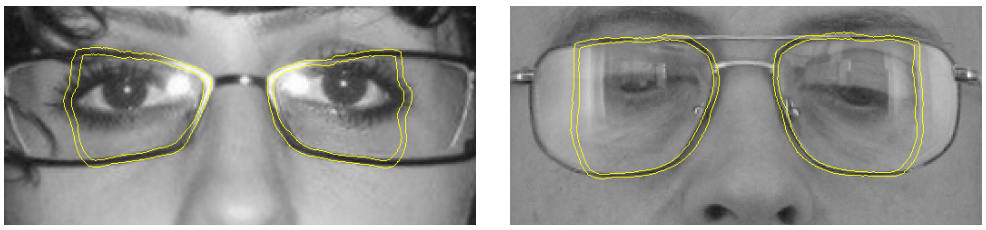


Figure 4.16: Eyeglasses segmentation example 14 and 15.

4.2.2 Comparison with manually annotated Eyeglasses

The evaluation on the eyeglasses segmentation is a complex task, because the manual eyeglasses segmentation takes time and is not always trivial. Often not even a human being can detect the eyeglasses frame correctly. For this evaluation 100 face images were used, where the eyeglasses were annotated manually.

To calculate the correctness of the segmentation the eyeglasses are divided into eight parts - four for the left frame, four for the right frame.

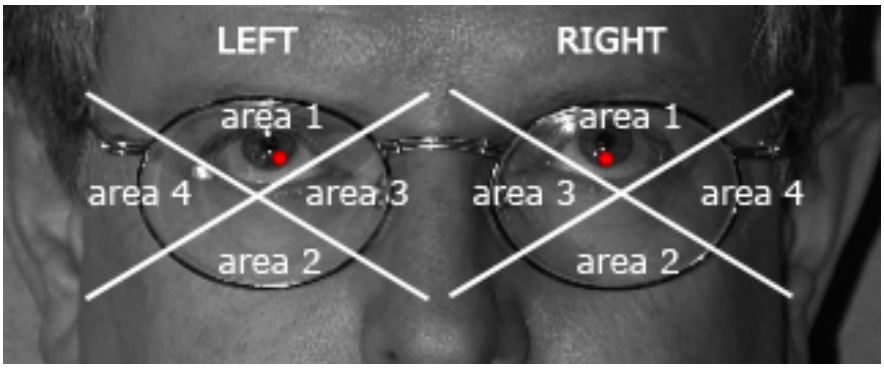


Figure 4.17: Eyeglasses area.

For the first evaluation we have four curves - the inner and outer frame edge for the left and right part. In each area we compute the mean error of the distance between the detected curve and the manually segmented one. The computed curve is subtracted from the manually annotated one, which means that positive values represent a 'too small' segmentation, whereas negative distance values yield to a 'too big' detection. In the Figures 4.19, 4.21, 4.23 and 4.25 the histograms of the absolute error distances are shown, whereas in 4.20, 4.22, 4.24 and 4.26 the histograms of the error distances are presented.

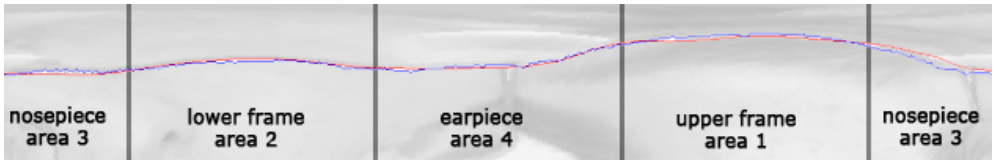
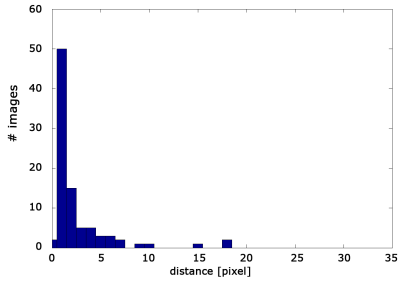
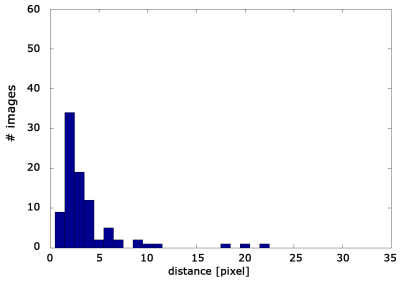


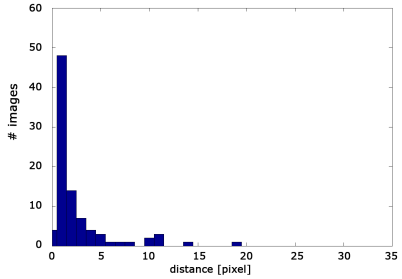
Figure 4.18: Comparison of the deteted curve (red) and manually annotated one (blue).



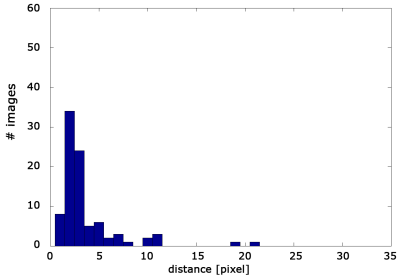
(a) Outer-left curve.



(b) Inner-left curve.

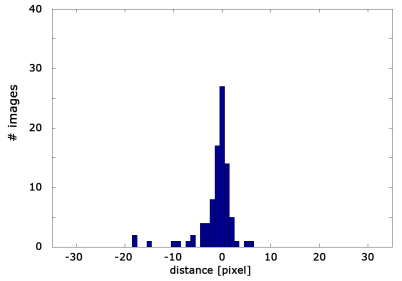


(c) Outer-right curve.

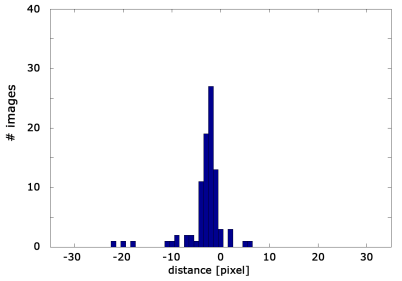


(d) Inner-right curve.

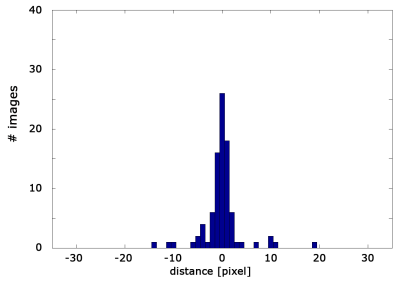
Figure 4.19: Absolute error distance histogramms for area 1.



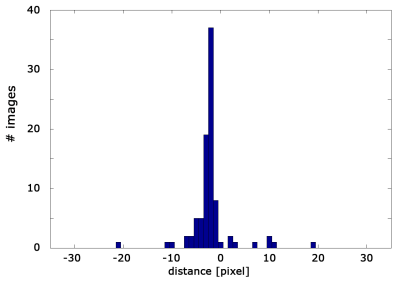
(a) Outer-left curve.



(b) Inner-left curve.

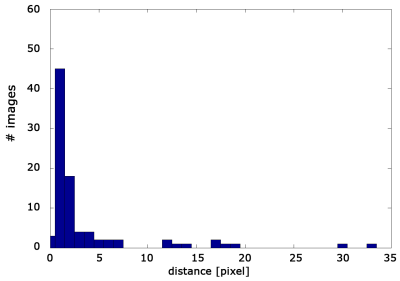


(c) Outer-right curve.

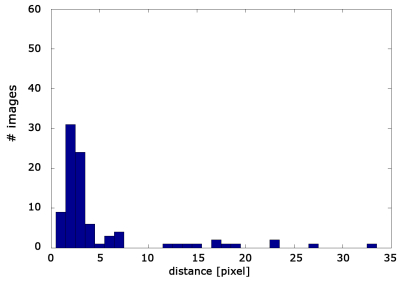


(d) Inner-right curve.

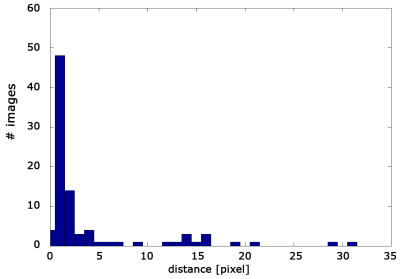
Figure 4.20: Error distance histogramms for area 1.



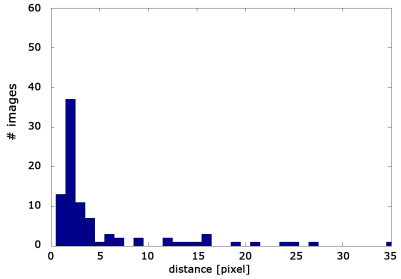
(a) Outer-left curve.



(b) Inner-left curve.

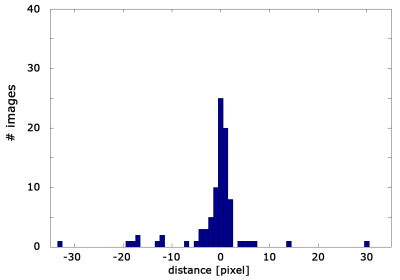


(c) Outer-right curve.

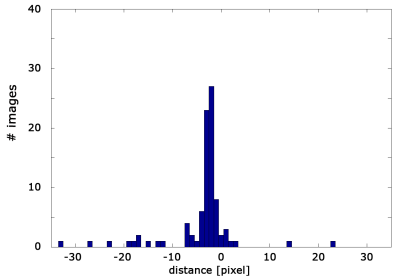


(d) Inner-right curve.

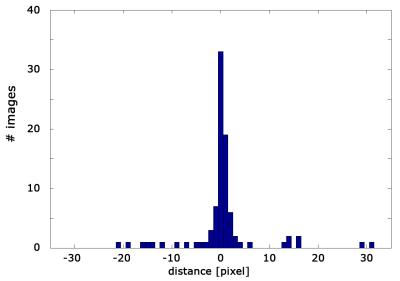
Figure 4.21: Absolute error distance histogramms for area 2.



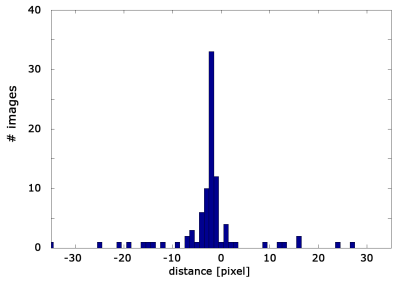
(a) Outer-left curve.



(b) Inner-left curve.

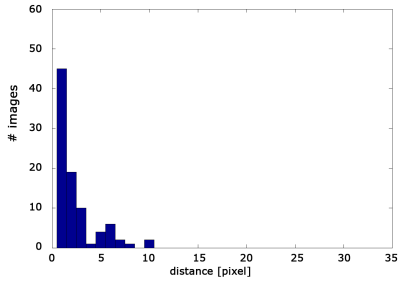


(c) Outer-right curve.

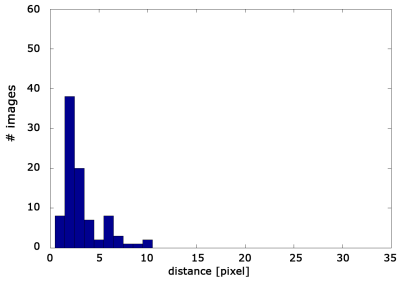


(d) Inner-right curve.

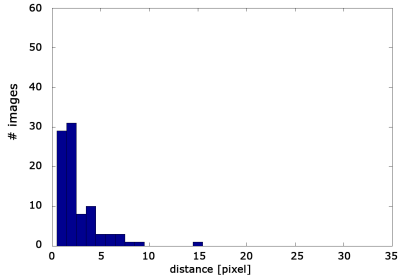
Figure 4.22: Error distance histogramms for area 2.



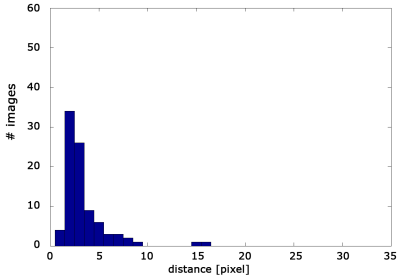
(a) Outer-left curve.



(b) Inner-left curve.

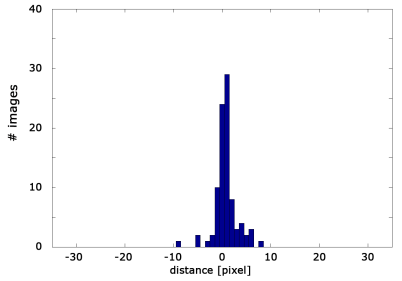


(c) Outer-right curve.

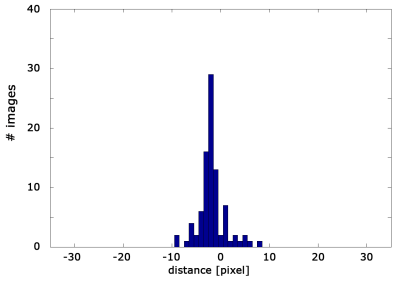


(d) Inner-right curve.

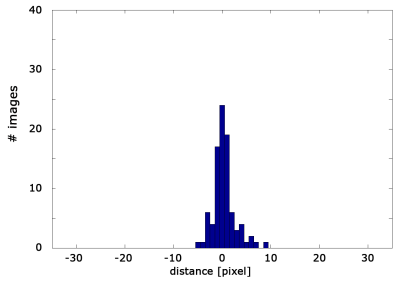
Figure 4.23: Absolute error distance histogramms for area 3.



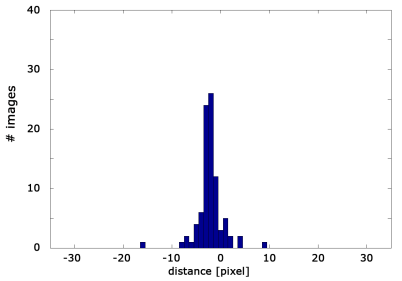
(a) Outer-left curve.



(b) Inner-left curve.

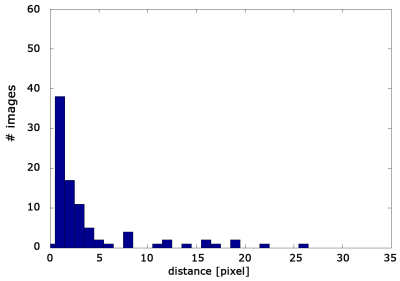


(c) Outer-right curve.

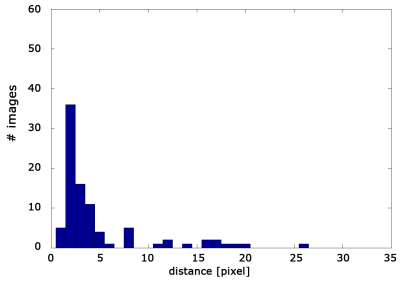


(d) Inner-right curve.

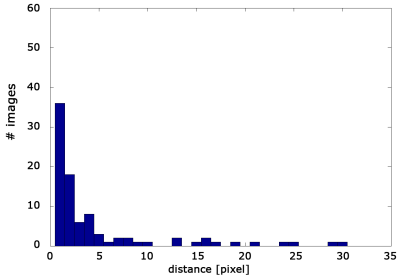
Figure 4.24: Error distance histogramms for area 3.



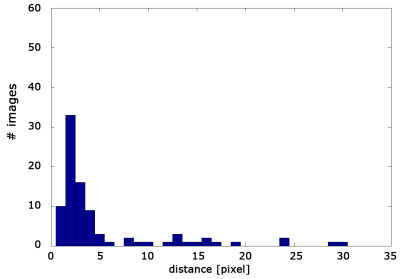
(a) Outer-left curve.



(b) Inner-left curve.

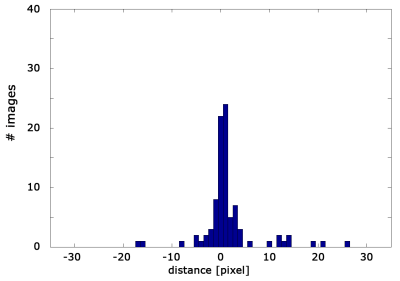


(c) Outer-right curve.

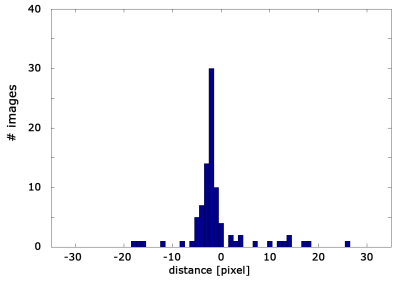


(d) Inner-right curve.

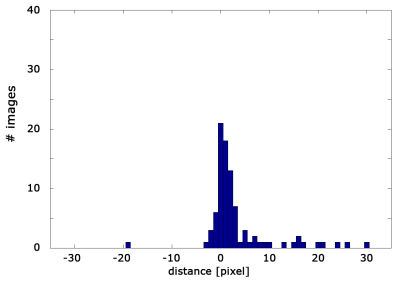
Figure 4.25: Absolute error distance histogramms for area 4.



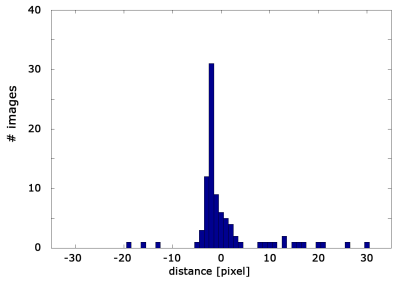
(a) Outer-left curve.



(b) Inner-left curve.



(c) Outer-right curve.



(d) Inner-right curve.

Figure 4.26: Error distance histogramms for area 4.

4.2.3 Discussion

The first part of the '*Eyeglasses Segmentation*' shows different segmented images. Most of the eyeglasses are segmented correctly, but the problem is that if the primary localisation process does not get an approximate localisation, the further steps yield to a wrong segmentation. In principle the occured difficulties can be summarised to a few major problems. There is the eyebrows area, which is discussed separately by the Dijkstra algorithm. Secondly there are eyeglasses without a frame (mainly the lower frame is missing), where it is tricky for the Snake to find the correct border of the glasses, which leads to a wrong eye-center to frame distance (see Figure 4.14 and Figure 4.15). Another 'problem' is that the frame often generates a shadow, which results in a wrong detection of the eyeglasses frame. A wrong segmentation of the frame in area three and four has no effect on the two computed eyeglasses parameters. In Figure 4.16 the eyeglasses frames are not localised correctly, which leads to a wrong segmentation.

The evaluation of the eyeglasses frame borders is divided into four parts, whereas the distance evaluations are done for each part. As the histograms show, the error distances of all curves and areas are normally distributed and the mean error distances are around zero. The distributions in area one and four show a few more outliers than the others which result from the shadow problem of the eyeglasses frame.

4.3 Eyeglasses Parameter

4.3.1 Evaluations

In this part two parameters of the eyeglasses are analysed - first the thickness of the frame (see Figure 4.28) and second the distance between eye center and upper/lower frame (see Figure 4.27). For evaluation 150 annotated images are used, where the eye-frame distance and the frame width are annotated manually.

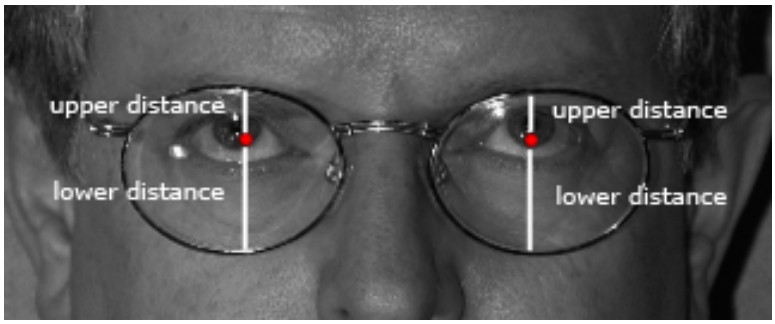


Figure 4.27: Distances between eye center and frame.

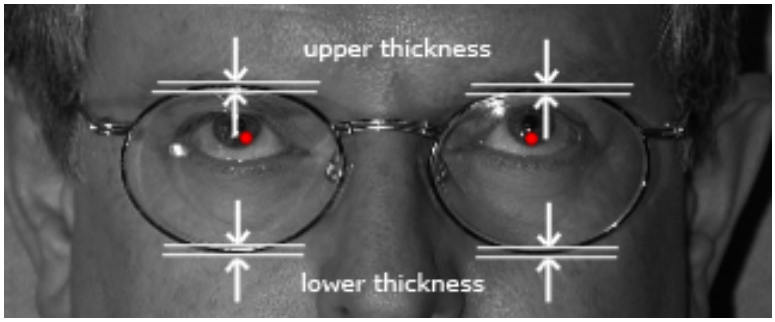
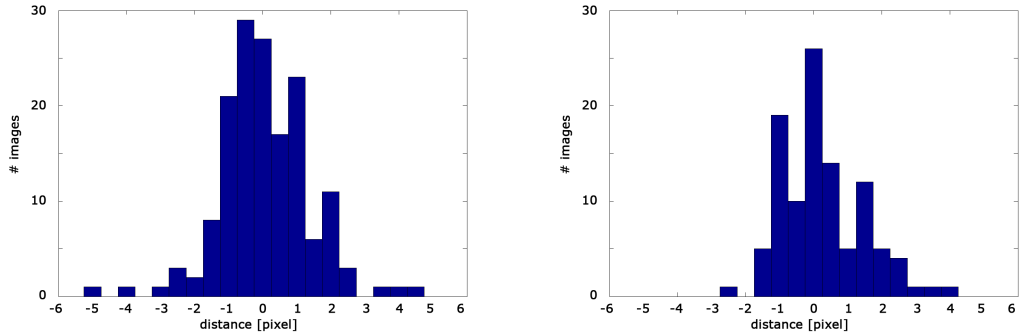


Figure 4.28: Thickness of the eyeglasses frame.

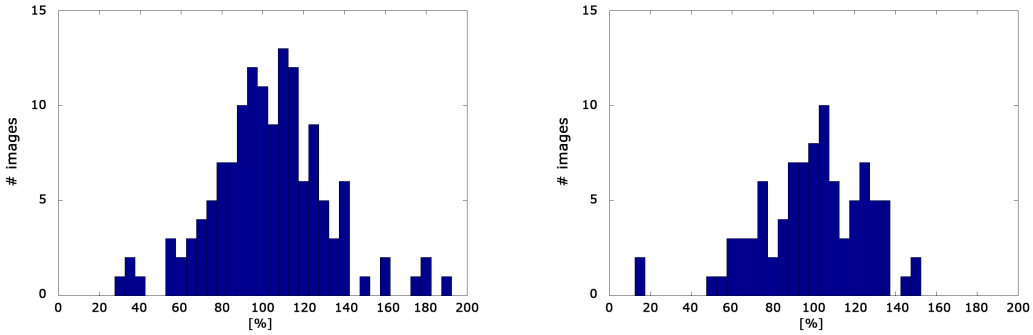
Only images are used, where the eyeglasses are detected correctly. Malfunction detections are discarded, because in this step the precision of the frame thickness and frame distance is the main goal. The calculated values are compared with the manually retrieved values. The difference values are then drawn in an histogram plot in the Figures 4.29, 4.32 and 4.35. In 4.30, 4.33 and 4.36 the percentual errors are shown, which are computed as follows (Formula 4.1).

$$Value_{Relative} = \frac{100 * Value_{Computed}}{Value_{Annotated}} \% \tag{4.1}$$



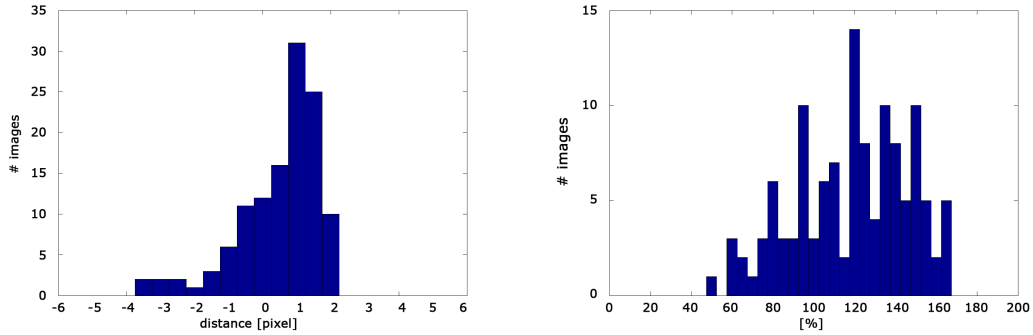
(a) Upper thickness. **(b)** Lower thickness.

Figure 4.29: Thickness error.



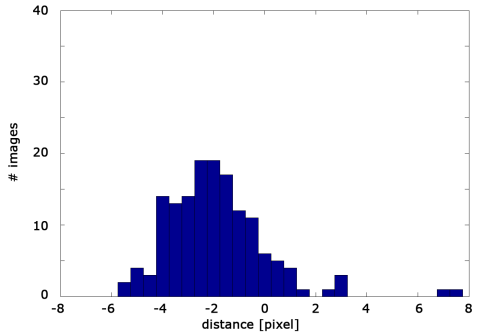
(a) Upper thickness. **(b)** Lower thickness.

Figure 4.30: Relative thickness error.

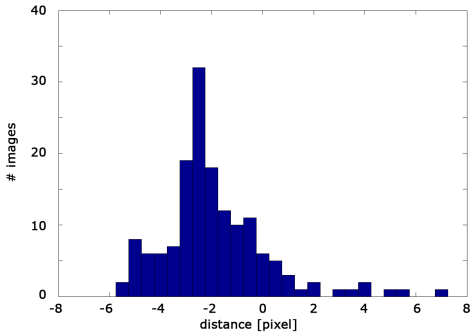


(a) Thickness error. **(b)** Relative thickness error.

Figure 4.31: Combined thickness error.

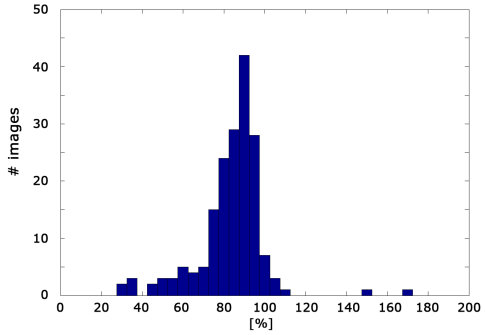


(a) Upper distance for left eye.

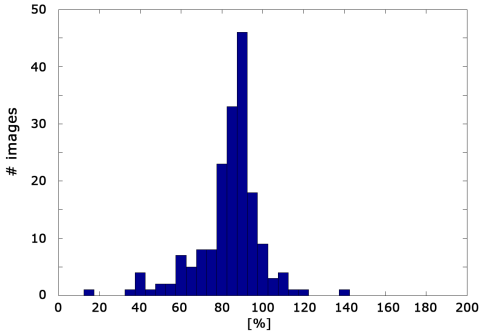


(b) Upper distance for right eye.

Figure 4.32: Upper distance error.

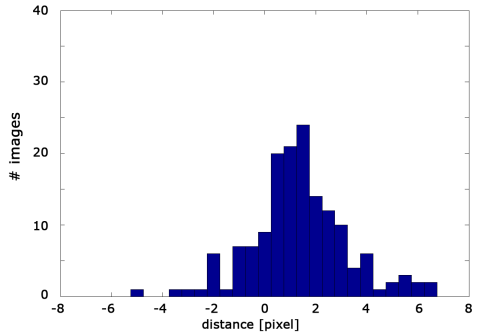


(a) Upper distance for left eye.

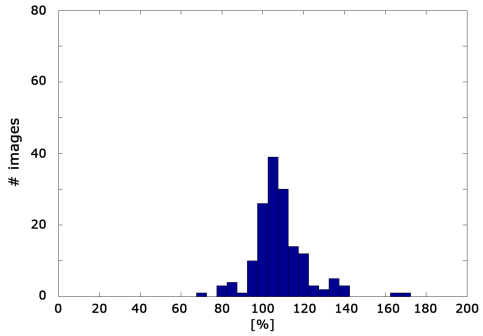


(b) Upper distance for right eye.

Figure 4.33: Relative upper distance error.

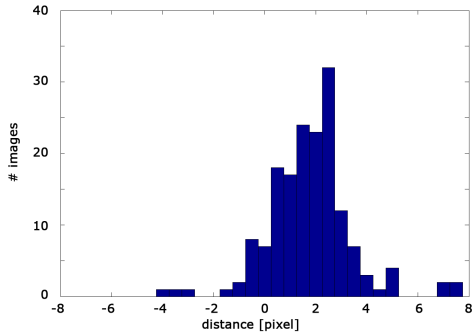


(a) Upper distance error.

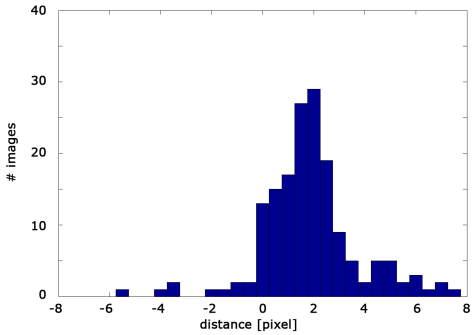


(b) Relative upper distance error

Figure 4.34: Combined upper distance error.

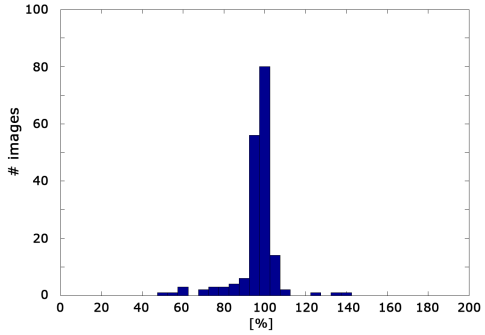


(a) Lower distance for left eye.

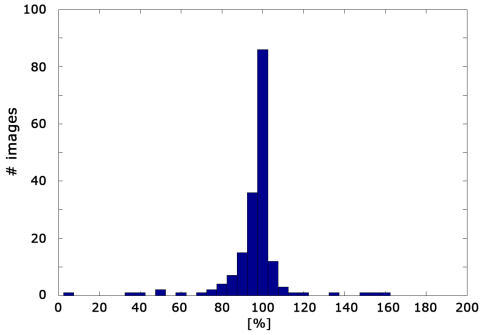


(b) Lower distance for right eye.

Figure 4.35: Lower distance error.

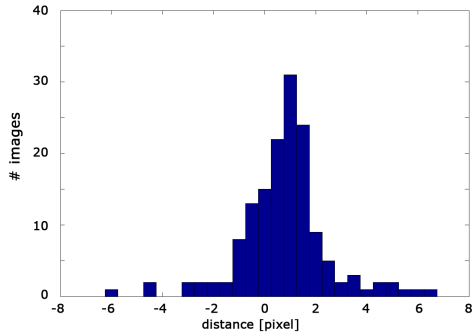


(a) Lower distance for left eye.

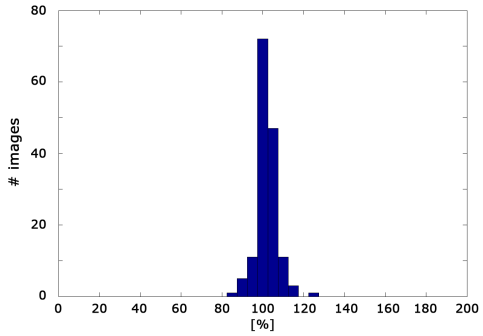


(b) Lower distance for right eye.

Figure 4.36: Relative lower distance error.



(a) Lower distance error.



(b) Relative lower distance error

Figure 4.37: Combined lower distance error.

4.3.2 Discussion

As required by Siemens the maximum variance of the frame thickness is 25 %, the minimum variance is 1 pixel. The histograms in Figure 4.29 and 4.30 and the chart in 4.1 show that a majority of the evaluated eyeglasses fullfill this requirement. For the eye-center to frame distance the requirements demand a maximum variance of 10 % and allow a minimum variance of 2 pixels. Compared with the results of the evaluation, a large part of the eyeglasses perform these conditions.

In Figure 4.31 the lower and upper thickness are combined whereas the thicker rating is taken. The same is done with the upper and lower distance (see Figures 4.34 and 4.37), whereas here the lower of the two distances is chosen. The combination is done to get single parameters for a further decision and validation of the eyeglasses.

	Relative error		Absolute error	
	μ [%]	σ [%]	μ [pixel]	σ [pixel]
Upper thickness	102.1	26.1	0.15	1.8
Lower thickness	99.7	26.3	-0.7	1.3
Final thickness	108.6	27.9	0.5	1.2
Upper distance for left eye	88.1	12.5	-2.6	1.9
Upper distance for right eye	88.0	11.0	-2.5	2.0
Final upper distance	108.4	13.6	1.3	1.9
Lower distance for left eye	96.6	8.5	1.4	1.7
Lower distance for right eye	98.9	10.1	1.3	1.9
Final lower distance	102.1	4.8	0.8	1.8

Table 4.1: Eyeglasses parameter - Statistical analysis.

These two parameters are a result of the eyeglasses segmentation. This means, to improve the results here, it is necessary to update the whole segmentation procedure. Another tricky aspect is the manual annotation, because it is often very challenging for a human being to annotate correct and exact values. This variance in annotation should be mentioned and kept in mind for further work on this task. This is a possible reason for all the methodic errors.

4.4 Chapter Summary

In this Chapter we showed the evaluation of the presented '*Eyeglasses Present Classification*' and '*Eyeglasses Segmentation*'. Our trained classificator of the '*Eyeglasses Present Classification*' was applied to four independent and different databases. The detailed detection results and the FAR/FRR distributions are shown. In the '*Eyeglasses Segmentation*' the found frame curves were compared with manually annotated ones. As a last part two eyeglasses parameters (frame thickness and eye-center to frame distance) were evaluated and the results are listed based on the manually annotated examples.

Chapter 5

Conclusion and Outlook

Contents

5.1 Conclusion

In this thesis we presented a system, which detects the presence, locates the position and determines several parameters of eyeglasses in face portrait images. As basis we used three different databases from Siemens, Caltech ([1]) and Feret ([18]). The databases contain standardised facial images with a size of 640x480 pixels, where all images are normalised with respect to the position of the eyes.

In the first step an eyeglasses detector method was developed. This algorithm is based on the 'Viola Jones face detector', which uses an 'AdaBoosting' classifier. For our purposes the classifier was trained with 500 facial images with and without eyeglasses. The retrieved classifier's performance on each of the databases has an error rate (EER) of maximum 3.0 % and a mean error rate of 1.9 % with 5000 unknown tested images. The requested performance of 99 % is not quite achieved. To increase the performance further, we suggest to train the classifier with more different databases, which offer a wider variety of different types (skin color, eye-region characteristics, etc.) .

The second and main part is the localisation of the eyeglasses. The primary localisation is based on a Snake, which is initialised with an ellipse located around the position of the eyes. So far it is not possible to 'scan' the whole face image for the eyeglasses and find the initialisation position automatically.

The internal parameters of the Snake are set manually and should keep the Snake in a circular shape. To increase the robustness of the Snake a Gaussian blurring filter is used to remove small interferences. With the underlying image a 'Gradient Vector Flow field' is computed, through which the Snake 'moves' to satisfy the energy function. Due to the different types of face images and huge variation of eyeglasses it is tricky to find the best Snake and preprocessing parameters to get an optimal solution. A disadvantage of the method is that for a good result a 'good' initialisation is necessary. The better the initial curve, the more exact is the final Snake. The 'Snake method' provides two curves - one for the left and one for the right eye.

In general the main localisation works satisfying, but it also shows a few main problems because of the large differences of the input data (skin color, eyeglasses shape and colour, shape of face, ...). First the eyebrows area, which is investigated separately by a shortest-path alogrithm. Secondly there are frameless eyeglasses (mainly the lower frame is missing), where it is tricky to find the correct glasses border. Another main problem is the shadow on the skin generated by the eyeglasses frame, which leads to a wrong detection of the frame. But all in all the evaluation shows that the main eyeglasses types are detected and segmented correctly. There are a number of eyeglasses where the vertical parts near the nosepiece and earpiece are not exact due to the complexity of the eyeglasses frame. And there are still examples of eyeglasses which can not be detected correctly at all (e.g. frameless eyeglasses).

The following steps are post-processing steps and are necessary to get more information about the found Snakes. To compare the independent curves a registration method is implemented, because it is uncertain that the left and right part of the eyeglasses frame are symmetric to the y-axis. With a cost function based on gradients, edges and intensity a good registration is provided, which allows a comparison of the left and right Snake. Further a correctness probability for the left

and right curve are derived and if required singular parts of the curves are adapted. Especially for the eyebrows area a Dijkstra algorithm is implemented, which tries to find the edge of the eyeglasses frame with a shortest path method. This method achieves improvements in this area.

As a last step the algorithm computes the eyeglasses parameters (frame thickness and eye-center to frame distance) and the exact frame borders. The parameter computation results from the segmentation, which means the better the segmentation the better the retrieved parameters. The retrieved results from 150 annotated images show that the required maximum thickness error of 25 % and the maximum frame distance error of 10 % are fullfilled by the majority (see Table 4.1) of evaluated facial images. As mentioned before an improvement here can only result from an improvement of the previous segmentation.

5.2 Outlook

For future work the main work has to concentrate on the main localisation step, which is done with a Snake here. Due to the wide variety of eyeglasses our method used no prior knowledge and no training data. Maybe it is reasonable to exclude a few types of eyeglasses to improve the performance for the main types of eyeglasses. This can be done with more prior knowledge (e.g. training step of annotated eyeglasses), especially with more information about the frame pattern and shape. Another possibility is a hybrid approach. If the correctness probability is low, a specialised frameless eyeglasses segmentation method can be used, which is derived in a training step (e.g. an ASM dedicated solely to frameless eyeglasses).

Bibliography

[1] Caltech (1999). Caltech faces database. http://www.vision.caltech.edu/html-files/archive.html.

[2] Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (1992). *Introduction to Algorithms*. MIT Electrical and Computer Science Series, MIT Press.

[3] Du, C. and Su, G. (2005). Eyeglasses removal from facial images. *Pattern Recognition Letters*, 26:2215–2220.

[4] Duda, R. O., Hart, P. E., and Stock, D. G. (2000). *Pattern Classification*. John Wiley & Sons, 2nd edition.

[5] Falcao, A., J.K. Udupa, J., Samarasekera, S., Sharma, S., Hirsch, B., and Lotufo, R. (1998). User-steered image segmentation paradigms - live wire and live lane. *Graphical Models and Image Processing*, 4:233–260.

[6] Ferrara, M., Franco, A., and Maltoni, D. (2000). Evaluating systems assessing face-image compliance with icao/iso standards. In *C.d.L. Scienze dell'Informazione*.

[7] Gonzalez, R. C. and Woods, R. E. (2008). *Digital Image Processing*. Prentice Hall International, 3rd edition.

[8] ISO/IEC (2004). Biometric data interchange formats - part 5: Face image data. *ISO/IEC JTC 1/SC 37N 506*, CD 19794-5.

[9] Jain, A. K. and Prabhakar, S. (2004). An introduction to biometric recognition. *IEEE Transactions on circuits and systems for video technology*, 14(1).

[10] Jiang, X., Binkert, M., Achermann, B., and Bunke, H. (1998). Towards detection of glasses in facial images. In *Proc. Fourteenth International Conference on Pattern Recognition*, volume 2, pages 1071–1073.

[11] Jiang, X., Binkert, M., Achermann, B., and H., B. (1997). Detection of glasses in facial images. In *Computer Vision - ACCV'98*, volume 1352/1997, pages 726–733.

[12] Jing, Z. and Mariani, R. (2000). Glasses detection and extraction by deformable contour. In *Proc. 15th International Conference on Pattern Recognition*, volume 2, pages 933–936 vol.2.

[13] Kass, M., Witkin, A., and Terzopoulos, D. (1987). Snakes: Active contour models. In *Int. J. Computer Vision*, volume 1(4), pages 321–331.

[14] McHenry, K. and Ponce, J. (2006). A geodesic active contour framework for finding glass. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 1038–1044.

[15] McHenry, K., Ponce, J., and Forsyth, D. (2005). Finding glass. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR 2005*, volume 2, pages 973–979 vol. 2.

[16] Mortensen, E. N. and Barret, W. A. (1998). Interactive segmentation with intelligent scissors. *Graphical Models and Image Processing*, 60:349–384.

[17] Park, J.-S., Oh, Y. H., Ahn, S. C., and Lee, S.-W. (May 2005). Glasses removal from facial image using recursive error compensation. *IEEE Transactions On Pattern Analysis and Machine Intelligence*, 27/5.

[18] Phillips, P. J., Wechsler, H., Huang, J., and J. Rauss, P. (1998a). The feret database and evaluation procedure for face-recognition algorithms. *Image Vision Comput.*, 16(5):295–306.

[19] Phillips, P. J., Wechsler, H., Huang, J., and Rauss, P. J. (1998b). The feret database and evaluation procedure for face-recognition algorithms. In *Image Vision Comput.*, volume 16(5), pages 295–306.

[20] Ramoser, H. (2005). Eyeglass detection in passport images. Technical report, Advanced Computer Vision - Technical Report.

[21] Saito, Y., Kenmochi, Y., and Kotani, K. (1999). Estimation of eyeglassless facial images using principal component analysis. In *Proc. International Conference on Image Processing ICIP 99*, volume 4, pages 197–201 vol.4.

[22] Sonka, M., Hlavac, V., and Boyle, R. (2007). *Image Processing, Analysis, and Machine Vision*. Thomson Learning, 3rd edition.

[23] Stockmann, G. and Shapiro, L. G. (2001). *Computer vision*. Prentice-Hall, New Jersey.

[24] Storer, M., Urschler, M., Bischof, H., and Birchbauer, J. A. (2008a). Classifier fusion for robust icao compliant face analysis. In *Proc. IEEE Int Conf on Automatic Face and Gesture Recognition (FG 2008)*, Amsterdam, The Netherlands.

[25] Storer, M., Urschler, M., Bischof, H., and Birchbauer, J. A. (2008b). Face image normalization and expression/pose validation for the analysis of machine readable travel documents. In *Proc. 32nd Workshop of the Austrian Association for Pattern Recognition: Challenges in the Biosciences: Image Analysis and Pattern Recognition Aspects*, pages 29–39, Linz, Austria.

[26] Subasic, M., Loncaric, S., Petkovic, T., Bogunovic, H., and Krivec, V. (2005). Face image validation system. In *Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis*.

[27] Urschler, M. (2001). Image-based verification of parametric models in heart-ventricle volumetry. Master's thesis, Institute of Computer Vision and Graphics, Technical University of Graz.

[28] Viola, P. and Jones, M. (1997). Robust real-time face detection. In *IEEE Proc. Conference on Computer Vision and Pattern Recognition*, volume 2, pages 747–747.

[29] Wu, B., Ai, H., and Liu, R. (2004a). Glasses detection by boosting simple wavelet features. In *Proc. 17th International Conference on Pattern Recognition ICPR 2004*, volume 1, pages 292–295.

[30] Wu, C., Liu, C., Shum, H.-Y., Xy, Y.-Q., and Zhang, Z. (2004b). Automatic eyeglasses removal from face images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(3):322–336.

[31] Wu, C., Liu, C., and Zhou, J. (2001). Eyeglasses verification by support vector machine. In *Proc. Second IEEE Pacific-Rim Conf. Multimedia*.

[32] Xu, C. and Prince, J. (1997). Gradient vector flow: a new external force for snakes. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 66–71.

[33] Xu, C. and Prince, J. (1998). Snakes, shapes, and gradient vector flow. *IEEE Transactions on Image Processing*, 7(3):359–369.