Machine Learning Techniques and Optical Systems for Iris Recognition from Distant Viewpoints

Dissertation

zur Erlangung des Doktorgrades
der Naturwissenschaften (Dr. rer. nat.)
der Fakultät Physik
der Universität Regensburg

vorgelegt von
Florian S. Langgartner
aus Regensburg

im Jahr 2019

Die vorliegende Dissertation entstand während einer dreijährigen Zusammenarbeit mit der Firma Continental Automotive GmbH, ansässig in der Siemensstraße 12 in 93055 Regensburg.

Das Promotionsgesuch wurde am 14.05.2019 eingereicht.
Die Arbeit wurde von Prof. Dr. Elmar Lang angeleitet.

Prüfungsausschuss:

Vorsitzender:	Prof. Dr. Josef Zweck
1. Gutachter:	Prof. Dr. Elmar Lang
2. Gutachter:	Dr. Stefan Solbrig
weiterer Prüfer:	PD Dr. Alfred Weymouth

Contents

Abstract

Vorhergehende Studien konnten zeigen, dass es im Prinzip möglich ist die Methode der Iriserkennung als biometrisches Merkmal zur Identifikation von Fahrern zu nutzen. Die vorliegende Arbeit basiert auf den Resultaten von [35], welche ebenfalls als Ausgangspunkt dienten und teilweise wiederverwendet wurden. Das Ziel dieser Dissertation war es, die Iriserkennung in einem automotiven Umfeld zu etablieren. Das einzigartige Muster der Iris, welches sich im Laufe der Zeit nicht verändert, ist der Grund, warum die Methode der Iriserkennung eine der robustesten biometrischen Erkennungsmethoden darstellt.

Um eine Datenbasis für die Leistungsfähigkeit der entwickelten Lösung zu schaffen, wurde eine *automotive Kamera* benutzt, die mit passenden NIR-LEDs vervollständigt wurde, weil Iriserkennung am Besten im nahinfraroten Bereich (NIR) durchgeführt wird.

Da es nicht immer möglich ist, die aufgenommenen Bilder direkt weiter zu verabeiten, werden zu Beginn einige Techniken zur Vorverarbeitung diskutiert. Diese verfolgen sowohl das Ziel die Qualität der Bilder zu erhöhen, als auch sicher zu stellen, dass lediglich Bilder mit einer akzeptablen Qualität verarbeitet werden. Um die Iris zu segmentieren wurden drei verschiedene Algorithmen implementiert. Dabei wurde auch eine neu entwickelte Methode zur *Segmentierung in der polaren Repräsentierung* eingeführt. Zusätzlich können die drei Techniken von einem *"Snake Algorithmus"*, einer aktiven Kontur Meth-

ode, unterstützt werden. Für die Entfernung der Augenlider und Wimpern aus dem segmentierten Bereich werden vier Ansätze präsentiert. Um abzusichern, dass keine Segmentierungsfehler unerkannt bleiben, sind zwei Optionen eines Segmentierungsqualitätschecks angegeben. Nach der Normalisierung mittels *"Rubber Sheet Model"* werden die Merkmale der Iris extrahiert. Zu diesem Zweck werden die Ergebnisse zweier *Gabor Filter* verglichen. Der Schlüssel zu erfolgreicher Iriserkennung ist ein Test der statistischen Unabhängigkeit. Dabei dient die *Hamming Distanz* als Maß für die Unterschiedlichkeit zwischen der Phaseninformation zweier Muster. Die besten Resultate für die benutzte Datenbasis werden erreicht, indem die Bilder zunächst einer *Schärfeprüfung* unterzogen werden, bevor die Iris mittels der neu eingeführten *Segmentierung in der polaren Repräsentierung* lokalisiert wird und die Merkmale mit einem *2D-Gabor Filter* extrahiert werden.

Die zweite biometrische Methode, die in dieser Arbeit betrachtet wird, benutzt die Merkmale im Bereich der die Iris umgibt (periokular) zur Identifikation. Daher wurden mehrere Techniken für die Extraktion von Merkmalen und deren Klassifikation miteinander verglichen. Die Erkennungsleistung der Iriserkennung und der periokularen Erkennung, sowie die Fusion der beiden Methoden werden mittels Quervergleichen der aufgenommenen Datenbank gemessen und übertreffen dabei deutlich die Ausgangswerte aus [35].

Da es immer nötig ist biometrische Systeme gegen Manipulation zu schützen, wird zum Abschluss eine Technik vorgestellt, die es erlaubt, Betrugsversuche mittels eines Ausdrucks zu erkennen.

Die Ergebnisse der vorliegenden Arbeit zeigen, dass es zukünftig möglich ist biometrische Merkmale anstelle von Autoschlüsseln einzusetzen. Auch wegen dieses großen Erfolges wurden die Ergebnisse bereits auf der *Consumer Electronics Show (CES)* im Jahr 2018 in Las Vegas vorgestellt.

Abstract

Previous research has shown that it is principally possible to use iris recognition as a biometric technique for driver identification. This thesis is based upon the results of [35], which served as a starting point and was partly reused for this thesis. The goal of this dissertation is to make iris recognition available in an *Automotive Environment*. Iris recognition is one of the most robust biometrics to identify a person, as the iris pattern is unique and does not alter its appearance during aging.

In order to create the database, which was used for the performance evaluations in this thesis, an *Automotive Camera* was utilized. As iris recognition is best executed in the near infrared (NIR) spectral range, due to the fact that even the darkest irises reveal a rich texture at these frequencies, the optical system is combined with suitable near infrared LEDs.

As the recorded images cannot always be processed right away, several preprocessing techniques are discussed with the goal of enhancing the image quality as well as processing only images that have an acceptable quality. In order to segment the iris, three different algorithms were implemented. Thereby, a newly developed *Segmentation in the Polar Representation* is introduced. In addition, the three techniques can be enhanced by a *Snake Algorithm*, which is an active contour approach. For removing the eyelids and eyelashes from the segmented area, four *noise removal* approaches are presented. For the goal of

ensuring that no fatal segmentations slip through, two options for a *segmentation quality check* are given. After the normalization with the *rubber sheet model*, the feature extraction is responsible for collecting the iris information, therefore, the results using a *1D-Log Gabor Filter* or a *2D-Gabor Filter* are compared. In the end, the key to iris recognition is a test of statistical independence. For this reason, the *Hamming Distance* serves well as a measure of dissimilarity between the phase information of two patterns. The best results for the database in use are gained by checking the image with a *Sharpness Check* before segmenting the iris by utilizing the newly introduced *Segmentation in a Polar Representation* and the *2D-Gabor Filter* as feature extractor.
The second biometric technique that is considered in this thesis is periocular recognition. Thereby, the features in the area surrounding the iris are exploited for identification. Therefore, a variety of techniques for the feature extraction and the classification are compared to each other. The performances of iris recognition and periocular recognition as well as the fusion of the two biometrics are measured with cross comparisons of the recorded database and greatly exceed the initial values from [35].
Finally, it is always required to secure biometric systems against spoofing. In the course of this thesis a printout attack served as the scenario that should be prevented, wherefore a working countermeasure is presented.
The results of this thesis points to the possibility of utilizing biometrics as a personalized car key in the future. Due to this huge success, the findings were also presented at the *Consumer Electronics Show (CES)* in 2018 at Las Vegas, yielding a great amount of feedback.

Chapter 1

Introduction

In the automotive industry four *Megatrends* can be identified. These are *Safety*, *Environment*, *Affordable Cars* and *Information*.

At first, the *Safety Megatrend* unites all those technologies that aim to increase the vehicle safety. The long term vision of automated driving, connected with the vision zero – zero fatalities, zero injuries, zero accidents – corresponds to this domain. It is currently most driven by Google and Tesla, the leaders concerning automated driving.

The *Environment Megatrend* tries to reach the goal of zero emissions, for example by using fewer fossil fuels. The carbon dioxide emissions shall be reduced in order to make the automotive world greener and cleaner, as well as less climate harming. Recent developments especially in relation with, as well as due to, the exhaust gas scandal show intensified advances towards fully electrified vehicles and fuel cell engines.

The third *Megatrend* is to make the existing technologies available in *Affordable Cars*. Of course, prosperity gaps and diverse expectations on cars require varying definitions for different parts of the world. In Western Europe and the United States of America the price limit for affordable cars is about $10,000$ Eu-

ros, whereas people in other parts of the world could never afford this amount of money, nor would they label this as cheap.

Last but not least, the *Information Megatrend* deals with gathering and using more and more pieces of information. It aims at optimizing the selection of presented data in order to adequately inform the driver. Moreover, it is supposed to prevent overcharging the attention of the driver for allowing relaxed and secure driving.

This thesis aims to contribute to the progress in the *Information Megatrend*. Since the possibility to robustly identify persons using biometric technology has been around for some years, the automotive industry wants to integrate these technologies in their environment, too. There are several possibilities to recognize individuals by biometric aspects, for example optically, thermally, capacitively or electrophorensicly. All of of them have different costs and security levels.

Generally, the methods can be differentiated between static and dynamic. Dynamic or behavioral biometric characteristics for example include gait analysis, voice analysis, signature analysis and keystroke dynamics. Common static or physiological biometric technology covers fingerprint recognition, face recognition, DNA sequence analysis, retinal scans and many more.

One of the most secure methods with comparatively low costs and the benefit of a relatively low time consumption is iris recognition. It is ideal for an integration in vehicles, as the required optical systems are already existing or will be available soon for modern cars.

This thesis is based upon the results of [35], which made the first steps towards iris recognition in an automotive environment from distant viewpoints, with realtime recognition and as little cooperation from the subjects as possible. All this with the goal of enhancing theft protection by only allowing autho-

rized people to start the engine. Therefore, [35] served as a starting point and was partially reused in this thesis. Similarly, the existing implementation using *Python* and *OpenCV* [29] was reutilized as a base for the comprehensive advances that will be presented in the following chapters. On top of the presented approaches, plenty other techniques were tried that will not be described, as this would vastly increment the size of this thesis, though adding only little additional information. The idea was to catch up with the open issues and ideas of [35] and finally implement the system in a car demonstrator, always keeping the long-term vision of completely replacing the car key by biometric technology in mind, which requires excellent recognition rates.

Chapter 2

Techniques

The following chapter introduces some general computer vision and machine learning techniques that were used throughout this thesis. *Histogram Equalization* (see 2.1) is utilized to optimize the usage of the full range of allowed values in an image. Thereby, contrast is enhanced globally. *Contrast Limited Adaptive Histogram Equalization (CLAHE)* (see 2.2) is an extension to *Histogram Equalization*. It enhances contrast not only globally but also locally. The *Z-Score Transform* (see 2.3) allows the standardization of distributions in a way that they become comparable to other distributions. In order to sharpen an image, two different *Sharpening Filters* (see 2.4) are presented, namely *Laplacian Filters* and a method named *Unsharp Masking*. *Median Filtering* (see 2.5) is a possibility to remove noise without smoothing the edges of an image. Finally the *Hough Transform* (see 2.6) is a tool that can be used in order to find simple geometric shapes, such as lines or circles. Thereby, usually a *Canny Edge Detector* (see 2.6.1) is utilized, which is a commonly used technique for edge detection, based on *Sobel Filtering*.

2.1 Histogram Equalization

Histogram Equalization [51, 62] is a technique for adjusting intensities in order to enhance the overall image contrast by stretching the intensity range. The distribution of a histogram is mapped to a more uniform and wide distribution of intensity values, making use of the maximum possible range. The first step for an 8 bit monochrome image with 256 possible intensities is to calculate the probability $p(i)$, with which each intensity value occurs with. This is done by

$$p(i) = \frac{n_i}{N}\,, \tag{2.1}$$

with n_i as the number of pixels with intensity $i \in [0, i_{max}]$, $i_{max} = 255$, and N the total number of pixels N. The cumulative distribution function F_{cd} is given by

$$F_{\mathrm{cd}}(i) = \sum_{j=0}^{i} p(j)\,. \tag{2.2}$$

It has to be multiplied with the size of the intensity range $i_{\max}$, in order to obtain the complete transformation function $T(i)$ [35]

$$T(i) = i_{\max} \sum_{j=0}^{i} \left(\frac{n_j}{N}\right)\,. \tag{2.3}$$

2.2 CLAHE

Contrast Limited Adaptive Histogram Equalization (CLAHE) [62] is a way to increase the contrast of an image. *Adaptive Histogram Equalization (AHE)* is an extension to the normal *Histogram Equalization* (see 2.1), which computes multiple histograms for different neighborhoods of the image. By equalizing the respective section's histograms, the local contrast is enhanced and illumination effects are evenly distributed. As *AHE* tends to amplify noise in images too much, the contrast is limited by clipping histogram bins, if they exceed a given value, the clip limit. Subsequently, the clipped parts are redistributed equally among all bins. This is then called *Contrast Limited AHE* or *CLAHE* [50]. As the calculation of multiple histograms is computationally expensive, it is possible to reduce complexity by interpolation or by a sliding window approach [61].

2.3 Z-Score Transform

The *Z-Score Transform* [33, 62] is used to standardize distributions in order to being able to compare differently distributed random variables. The *Z-Score Transform* Z of a value x from a sample $\mathbf{X}$ is given by

$$Z(x) = \frac{x - \bar{\mathbf{X}}}{\sigma_{\mathbf{X}}} \tag{2.4}$$

with $\bar{\mathbf{X}}$ being the sample's mean value and $\sigma_{\mathbf{X}}$ the sample's standard deviation. In image processing, the application of the *Z-Score Transform* results in an increased invariance to different illuminations. Important thereby is that due to the resulting distribution's mean value of zero and standard deviation of one, the values will not be integers and might as well be negative, preventing an

interpretation as image intensity. Therefore, these values need to be rescaled back to the original 8-bit image space by normalization to the range of $[0, 255]$.

2.4 Sharpening Filters

In order to sharpen an image, *Sharpening Filters* [51, 62] can be applied. For creating such filters, the *Laplacian Operator* $L(x, y)$ of an image $I(x, y)$

$$L(x, y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} \tag{2.5}$$

can be utilized. As images are represented by discrete pixel intensities, it is required to use discrete convolution kernels, which are approximations of the second derivatives in the *Laplacian Operator* $L(x, y)$. The two most commonly used *Laplacian Filters* are

$$L_4 = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix} \quad \textit{and} \quad L_8 = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix}. \tag{2.6}$$

Applying these filters results in the enhancement of discontinuities and edges of an image on a featureless background. For finally obtaining the filtered image, the resulting *Laplacian image* is added to the original image. It is also possible to design the filters in a way that they perform both steps at once, in order to simplify the computation:

$$L_{4+} = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{pmatrix} \quad \textit{and} \quad L_{8+} = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{pmatrix}. \tag{2.7}$$

Another possibility to sharpen an image is given by a technique called *Unsharp Masking*. Thereby, the image is enhanced by

$$I_{\text{enhanced}} = I_{\text{original}} + a \cdot (I_{\text{original}} - I_{\text{blurred}}) \tag{2.8}$$

with a as a value that adjusts the enhancement potency. The blurred image I_{blurred} is obtained by either averaging the original image or using a *Gaussian Filter*. An example for the resulting *Unsharp Masking* filters M_{us} with a size of 9×9 is given by

$$M_{\text{us},9\times9} = \frac{1}{81} \begin{pmatrix} -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & 161 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \end{pmatrix} . \tag{2.9}$$

2.5 Median Filtering

In order to perform *Median Filtering* [51, 62], the median of the pixels in a defined neighborhood area is taken as the new pixel value. Thereby, noise is effectively removed, without smoothing the edges, which makes it a suitable tool for facilitating the segmentation (see 5.2) of images. Bigger kernels can even remove larger distortions, such as reflections. Figure 2.1 shows the impact of *Median Filtering* on an image from the self-recorded database (see chapter 4). The left image is the original. The other two pictures were filtered with kernel sizes of 7 for the middle one and 29 for the right one, respectively. In the slightly filtered image the smoothing is visible and eases the segmentation. The heavily filtered picture is even smoother and does no longer show the reflections from the NIR LEDs (see Figure 3.5).

(a) original image (b) image filtered with kernel size 7 (c) image filtered with kernel size 29

Figure 2.1: The two images on the right depict the effect of *Median Filtering* with kernel sizes of 7 and 29 respectively, on an image from the self-recorded database (see chapter 4). In the slightly filtered image the smoothing is visible and eases the segmentation. The heavily filtered picture is even smoother and does no longer show the reflections from the NIR LEDs (see Figure 3.5).

2.6 Hough Transform

The *Hough Transform* [62] is a standard technique in the fields of image analysis, computer vision and digital image processing. It was invented by P. Hough [27] and enhanced by R. Duda and P. Hart [20] and can be used to determine the parameters of lines or circles in an image. Even more complex structures could be found by applying the *Hough Transform*, too, but the more complex a structure is, the higher the storage and computational requirements become. If it is assumed that the iris and the pupil can be approximated by circles, the *Hough Circle Transform* can be used to find their radius and center coordinates [32][37][63][66]. Similarly, the eyelids can be approximated by lines. Therefore, the *Hough Line Transform* constitutes a very simple approach to detect them [35].

2.6.1 Canny Edge Detection

For both, the *Hough Line Transform* and the *Hough Circle Transform*, only the relevant edge information of a picture should be used, which also drastically decreases the computational load. To this end, the *Hough Gradient Method* [49][69] generates an edge map before performing the *Hough Transform*. This may be achieved by application of the *Canny Edge Detector*, which is a technique for edge detection invented by J. Canny [7]. It extracts useful edge information from a picture and thereby reduces the amount of data for the following computational steps. The algorithm consists of four stages:

- At first, a 5×5 *Gaussian Filter* is applied in order to reduce noise, as all edge detection algorithms are sensitive to noise.
- Secondly, the denoised image is filtered by *Sobel kernels* in horizontal and vertical direction, in order to calculate the magnitude and the direction

of the intensity gradient of the image.

- The third step is called *non-maximum suppression*. It removes unwanted pixels that are not the local maxima in the neighborhoods of the same gradient directions. The result is a binary image representing the thin edges of the image.

- The last step is the *hysteresis thresholding*, which decides whether edges are really edges or not. Therefore, upper and lower thresholds for the intensity gradient are introduced. All values that are smaller than the lower threshold are discarded and all values that are bigger than the upper threshold are considered to be proven edges. For the values in between the two thresholds it is checked if they are connected to a pixel that is assured to be an edge. If so, they are treated as edges, otherwise they are discarded. Thereby, it is also possible to remove the remaining small noise pixels by assuming that edges are always long lines [35].

2.6.2 Hough Line Transform

The simplest case of a *Hough Transform* is the detection of straight lines, which can be described by

$$r = x\ cos\ \theta + y\ sin\ \theta\,, \tag{2.10}$$

in a polar coordinate system (r, θ), with the distance r to the closest point on the line and its angle θ to the x-axis (see Figure 2.2). Every line can be assigned to a specific point in the two-dimensional *Hough space* (r, θ). Every single point (x_0, y_0) corresponds to a unique sinusoidal curve in the *Hough space* (r, θ), since it could be traversed by many straight lines with different angles θ.

The curves of points (x, y) forming a straight line will cross in the point (r_0, θ_0) representing that line. As lines consist of many points (x, y), it is possible to set a threshold indicating how many crossings are needed to decide whether a crossing point in the *Hough space* really represents a line [35].

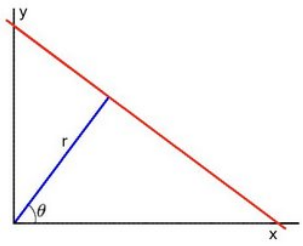


Figure 2.2: Representation of lines in polar coordinates (r, θ) for the *Hough Line Transform*: r is the distance to the closest point on the line and θ is the angle between r and the x-axis [35].

2.6.3 Hough Circle Transform

The *Hough Circle Transform* uses the same principle as the *Hough Line Transform*. The only difference is that the *Hough space* $(r, x_{\text{center}}, y_{\text{center}})$ is now three-dimensional, as three variables are needed to describe a circle:

$$r^2 = (x - x_{\text{center}})^2 + (y - y_{\text{center}})^2 \tag{2.11}$$

with $(x_{\text{center}}, y_{\text{center}})$ being the center of the circle and r its radius. Usually the radius is fixed to a certain value in order to find the optimum center for the circle in the two-dimensional *Hough space* $(x_{\text{center}}, y_{\text{center}})$. This is repeated for several radii. Subsequently, the variable combination with the most crossings in the *Hough space* is chosen as the final result [35].

Chapter 3

Optical Systems

Different camera systems produce pictures with different properties. As a consequence, cross comparisons between different cameras are far from being meaningful. At the point when the research on iris recognition started [35], no appropriate *Automotive Camera* was available. Therefore, the *Foscam FI8918W IR Night Vision Camera* (see 3.1) provided an initial temporary solution. It is still worth mentioning at this point to emphasize that iris recognition could theoretically also be done using such a simple consumer night vision camera. Soon thereafter, the algorithm was optimized for the *Basler Automotive Camera* (see 3.2) in combination with a zoom objective and external IR diodes [35]. In the course of this thesis several other components are used in addition thereto and will be addressed in the following.

3.1 Foscam IR Night Vision Camera

3.1.1 Camera

The *Foscam FI8918W IR Night Vision Camera*, depicted in Figure 3.1, is a consumer night vision camera that is commonly used for video surveillance of private property. It has an objective lens with a focal length of 2.8 mm and is able to record RGB images with a resolution of 640 × 480 pixels. It also provides a ring of integrated 850 nm NIR LEDs. Due to the limited resolution,

Figure 3.1: *Foscam FI8918W IR Night Vision Camera*, 640 × 480 pixels, RGB, focal length of 2.8 mm, integrated 850 nm NIR LEDs, optimum native distance: 1 cm [35].

the optimum distance for image recording is about 1 cm. This distance assures the 70 pixels on the radius of the iris, which are at least needed for a proper iris recognition (see chapter 5). The image recording and the camera settings, like contrast, brightness and control of the IR diodes have to be set in a browser interface [35].

3.1.2 Pictures

The small recording distance enabled the relatively weak built-in IR LEDs to be sufficient to illuminate the iris thoroughly, which is not possible for larger distances. Even though the recording in such a small distance is challenging, it is still possible to produce usable images. Figure 3.2 is a high-quality example for an image of an eye that was recorded with the *Foscam camera*. The reflection of the IR LEDs is visible through white points in the image. All the recordings were conducted in the dark, in order to have constant exposure conditions and to maximize the proportion of the IR light. Nevertheless, the pictures kept showing a hint of green, which was caused by the green status LED of the camera.

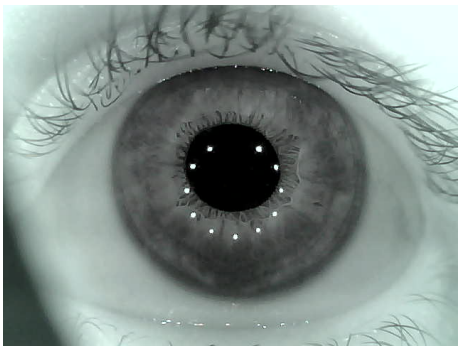
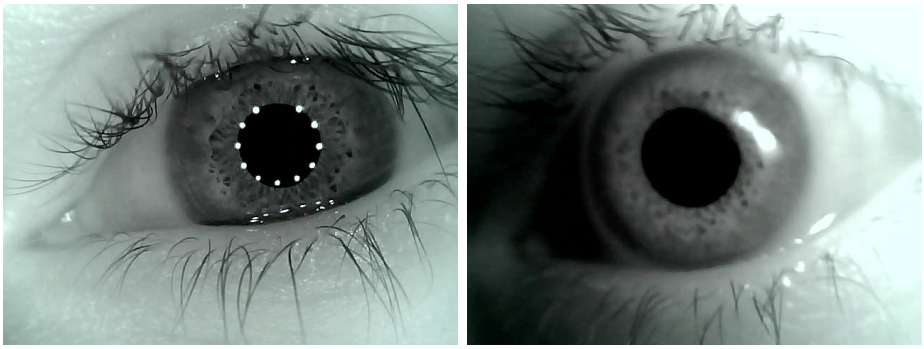


Figure 3.2: High-quality example for a picture of an eye that was recorded with the *Foscam camera*. The ring of IR LEDs is visible through white points in the image. The camera's status LED causes the hint of green [35].

The impact of the IR illumination is very strong if the *Foscam camera* is used, as Figure 3.3 illustrates. It shows pictures of the same eye, recorded with the *Foscam camera*, with and without the built-in IR LEDs. Again, a ring of white points is visible in the left Figure 3.3(a) and proves that the IR LEDs were

switched on. Here, the iris pattern appears very clearly, the contrast is good and the brightness is evenly spread. Figure 3.3(b) on the right side still shows parts of the iris pattern, but the whole picture is blurry and noisy. It features an unbalanced illumination: a huge shadow on the left part of the image and a very bright region on the opposite side. Further pictures that were recorded using the Foscam night vision camera are provided in the Appendix A.1 [35].



(a) An eye recorded with the *Foscam camera* with IR illumination

(b) An eye recorded with the *Foscam camera* without IR illumination

Figure 3.3: Pictures of the same eye, recorded with the *Foscam camera*, with and without the built-in IR LEDs. The impact of the IR illumination is nicely visible. Unlike the right image, the left image shows a very clear iris pattern with good contrast and an evenly spread brightness. The ring of white points proves that the IR illumination was switched on [35].

3.2 Basler Automotive Camera

3.2.1 Camera, Sensor, Objective Lens and NIR LEDs

The first automotive camera used in the course of the creation of this thesis was the *Basler Automotive Camera daA1280-54um*, which is a camera that uses the same *Aptina AR0134 CMOS Sensor* that is currently being used in the cameras that are built into modern cars. It provides a resolution of 1280×960 pixels and captures monochrome images. In order to allow more flexibility in comparison to a fixed focal length lens, a *Tamron Mega-Pixel M12VM412 zoom lens* with an adjustable focal length from 4 to 12 mm was used to gather the first experiences and in order to be able to quickly change the aperture, the focus and the zoom by hand. In Figure 3.4 the lens is depicted mounted on the camera. In order to increase the system's maximum distance that allows iris recognition, lenses with higher focal lengths were used later on. Using a lens with a fixed focal length of 25 mm, it became possible to increase the distance from 20 to 30 cm

Figure 3.4: *Basler daA1280-54um Automotive Camera* with *Aptina AR0134 CMOS Sensor* and *Tamron Mega-Pixel M12VM412 zoom lens*, 1280×960 pixels, monochrome, focal length from 4 to 12 mm, and optimum native distance from 20 to 30 cm [35].

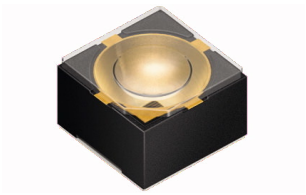


Figure 3.5: *OSRAM High Power IR LED SFH 4780S*, especially designed for iris recognition, centroid wavelength 810 nm, narrow half angle of ±10°, operating in constant mode with 500 mA current [35] or camera synchronized pulsed mode with 2 A current.

up to 1 m, while maintaining the condition of having the minimum of 70 pixels on the iris's radius (see chapter 5). This even sufficed to integrate the camera in the interior mirror of a car and therefrom allows recording the iris data of the person sitting in the driver seat. *Auto focus* helped to enlarge the degree of freedom, which the driver has, for proper positioning in front of the camera. This will be addressed in section 3.3. Later on, there were cameras from Basler available with up to 5 megapixels, too. Those were also tried out, but since there was no actual need for switching to higher resolutions, since the maximum distance of 1 m sufficed for the application in the automotive field, their capabilities were not explored further. Furthermore, it is unlikely that such resolutions will be used in any vehicles for the next decade. In order to reveal the iris textures of dark irises [11] and to allow the system to operate at night, two *OSRAM High Power IR LEDs SFH 4780S* (see Figure 3.5) were utilized. These diodes are especially designed for iris recognition and emit light with a centroid wavelength of 810 nm. With a narrow half angle of ±10°, they have a very straight light cone compared to other LEDs. They were first operated by a constant current LED driver with 500 mA current. Later on the driver was replaced by a pulsed LED driver, to be able to employ a more powerful camera synchronized pulsed mode, with 2 A current, in order to

Figure 3.6: Example for a face recorded with the first *Basler camera* setup at the maximum distance. There are approximately 70 pixels on the iris radius. The IR illumination was switched on. This is visible by the reflections in both pupils [35].

allow brighter illumination at higher distances. With the goal of making the illumination as uniformly as possible, an IR filter, which eliminates light from the visible range, was used. Figure 3.6 shows an example of a face that was recorded using the *Basler camera* with the *Tamron objective* at the maximum distance of 30 cm. The iris possesses about 70 pixels on its radius. The picture was taken using the IR filter and the IR illumination from the OSRAM LEDs [35].

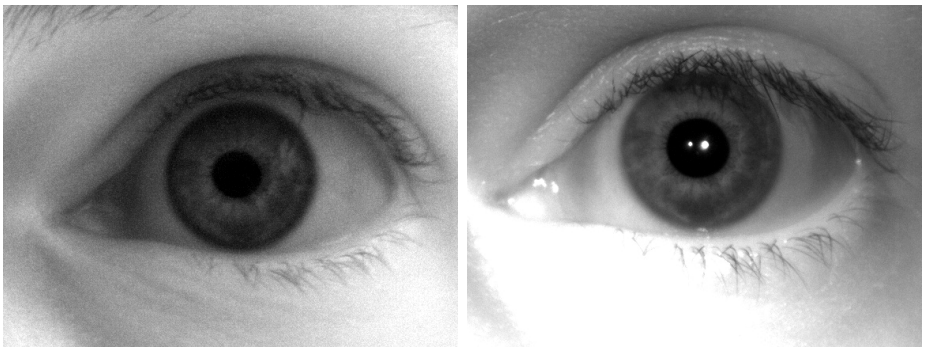
3.2.2 Pictures

At the very beginning it was cumbersome to record high-quality images with the *Basler camera*, since it was difficult to align the camera settings with the aperture, the focus and the zoom of the *Tamron lens* in order to maintain the focus at the correct distance and still obtain a well exposed image. During these struggles, the idea for the latter usage of *auto focus* (see 3.3) was born. Nevertheless, the manual setup makes iris recognition possible. Figure 3.7 is a good example for a picture of an eye that was recorded with the *Basler camera*. It is a sub-picture of Figure 3.6. As in the pictures from the *Foscam camera* (see 3.1.2) the IR LEDs are visible as white points. Again, the impact

Figure 3.7: This picture is a sub-picture of Figure 3.6. It is a good example of a picture of an eye that was recorded with the Basler camera. As in the pictures from the *Foscam camera* (see 3.1.2) the IR illumination is visible through the two reflections at the pupil [35].

of the IR illumination (see chapter 5), in this case coming from the *OSRAM High Power LEDs*, is very strong for the Basler camera. In order to emphasize this, Figure 3.8 consists of two pictures of the same eye, with and without the *OSRAM IR LEDs* switched on. In order to suppress the reflections caused of the visible light, the right picture was taken utilizing an IR filter. Both pictures were enhanced using *Histogram Equalization* (see 2.1). Otherwise, the pictures

would appear much darker, especially the one without the IR illumination on the left side (see Figure 3.8(a)). Nevertheless, this image is blurry and noisy. The iris looks glazed and shows reflections on the right side. The iris pattern is partly visible but does not offer such a clear look as the one on the other side (see Figure 3.8(b)). Even though the IR LEDs were not perfectly directed at the iris, the iris pattern appears very nicely and shows a quite high contrast. Furthermore, the two reflections at the pupil prove that the IR LEDs were switched on, while the image was recorded. More pictures that were recorded with the *Basler camera* can be found in the Appendix B [35].



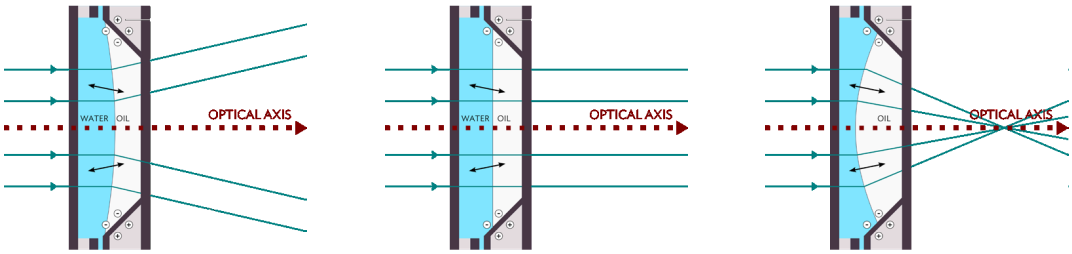
(a) without IR illumination (b) with IR illumination

Figure 3.8: *Histogram equalized* (see 2.1) pictures of the same eye, recorded with the *Basler camera*, with and without the IR illumination from the *OSRAM IR LEDs*. The impact of the IR illumination is nicely visible. The image on the left side is blurry and noisy and shows a glazed iris with reflections. On the other hand side, an IR filter suppresses the reflections of visible light, whereas the IR LEDs allow a clear look on the iris pattern with sharp contrast, although they were not perfectly directed at the eye [35].

3.3 Auto Focus

Lenses with fixed focal lengths have a limited depth of field. To capture sharp images, the eye has to be positioned within a small region that is determined by the lens parameters. To ensure optimum sharpness, regardless of the driver's positioning, an *auto focus* can be used. In the automotive world movable objects are used as rarely as possible, since they are more probable to break than fixed objects. For this reason a mechanical auto focus lens is not an option. A quite interesting solution is offered by liquid lenses. These allow controlling their curvature by changing the applied voltage, without requiring any mechanical movement, resulting in a variable focus. A *Corning Varioptic C-C-39N0-250 lens* (see Figure 3.9) was chosen as it was promising a wider operating temperature range than the lenses from other suppliers, which enlarges the probability for an automotive certification. Besides, it is robust against shock and vibration, which makes it even more suitable to be placed in vehicles. The lenses use an effect called electrowetting [43], in which an amount of

Figure 3.9: *Corning Varioptic C-C-39N0-250 auto focus lens*: offers a wide operating temperature range and is robust against shock and vibration. This enables it to be placed in vehicles. The lens structure is depicted in Figure 3.10. This image was taken from [8].



(a) Divergent Lens (b) Flat Lens (c) Convergent Lens

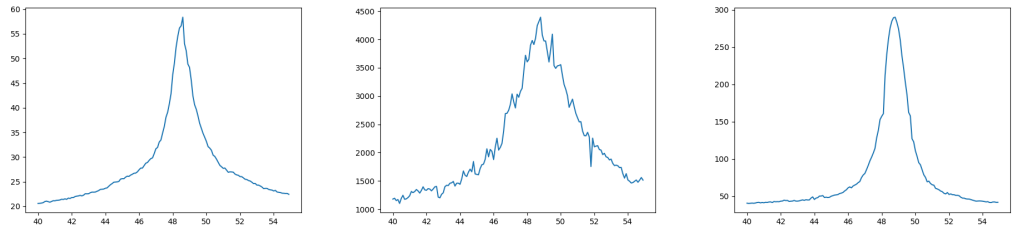
Figure 3.10: Images showing which lens structure was used to realize electrowetting inside the lens. Depending on the applied voltage the possible lens states are divergent, flat and convergent. These images were taken from [8].

insulating liquid, e.g. oil, is placed on conductive material with an insulating surface and surrounded by conductive liquid like water. The shape of the oil layer is controlled by applying voltage between the conductive substrate and the conductive liquid [8]. Figure 3.10 shows how electrowetting is realized inside the lens, as well as the possible states the lens can take up depending on the applied voltage: divergent, flat and convergent. In order to employ a working auto focus with the liquid lense, a proper voltage control algorithm had to be implemented. State of the art auto focus implementations work by optimizing the contrast by constantly shifting the voltage and searching for the maximum amount of edges in the resulting images. In order to calculate the number of edges, one of the most well known possibilities is the *Sobel Filter*. Another option is to use a *Fast Fourier Transform (FFT)* to measure the amount of high frequencies in the image. This allows to deduce its sharpness: An increased amount of high frequencies correlates with a higher sharpness. A third option is given by Daugman [11]. He stated that the *Fast Fourier Transform (FFT)* is generally the right tool to cope with the problem but suggested the use of an

-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	+3	+3	+3	+3	-1	-1
-1	-1	+3	+3	+3	+3	-1	-1
-1	-1	+3	+3	+3	+3	-1	-1
-1	-1	+3	+3	+3	+3	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1

Figure 3.11: 8×8 filter for fast focus assessment by Daugman [11].

8×8 filter (see Figure 3.11), which serves as a low computational complexity image frequency analyzer. In order to achieve a sharp image the voltage is varied and the output of the sharpness analyzer is tracked. Figure 3.12 shows graphs for the output values of a static object for the three solutions from 40 V to 55 V. For the *Sobel Filter* as well as the filter of Daugman's approach the mean values of the filter's outcome are taken as the result. For the *FFT* the value is calculated by taking the mean of the highest frequencies. Apparently, the *Sobel Filter* creates a curve with a sharp peak, whereas the *FFT* output is wider and more noisy. The filter Daugman suggested results in a flatter graph in the non-peak region, compared to the *Sobel Filter*, but does not have such a sharp peak. Table 3.1 summarizes the performances of the three solutions. The *Sobel Filter's* output values range from 20 to 60. The ratio between those values (3.0) serves as a measure of the discriminability of the peak and the base. Its *full width at half maximum (FWHM)* amounts to 1.7 V. Considering the *FFT* factor of 3.6 V it might seem that it is slightly more powerful than



(a) Sobel filter (b) FFT (c) Daugman

Figure 3.12: Graphs of the output values (with no unit) of the auto focus solutions from 40 V to 55 V of a static object. The *Sobel Filter* creates a curve with a sharp peak, the *FFT* curve is wide and noisy and the filter Daugman suggested produces a graph that is flat in the non-peak region and has a curvy peak.

the *Sobel Filter*, but with a more than doubled *FWHM* (3.6 V) this can be discarded. The most powerful solution is the filter that Daugman suggested. It features a discriminability ratio of 7.0 and a *FWHM* of 1.5 V. Both are superior values. Therefore, one is best advised to neglect the *Sobel Filter* as well as the *FFT* approaches.

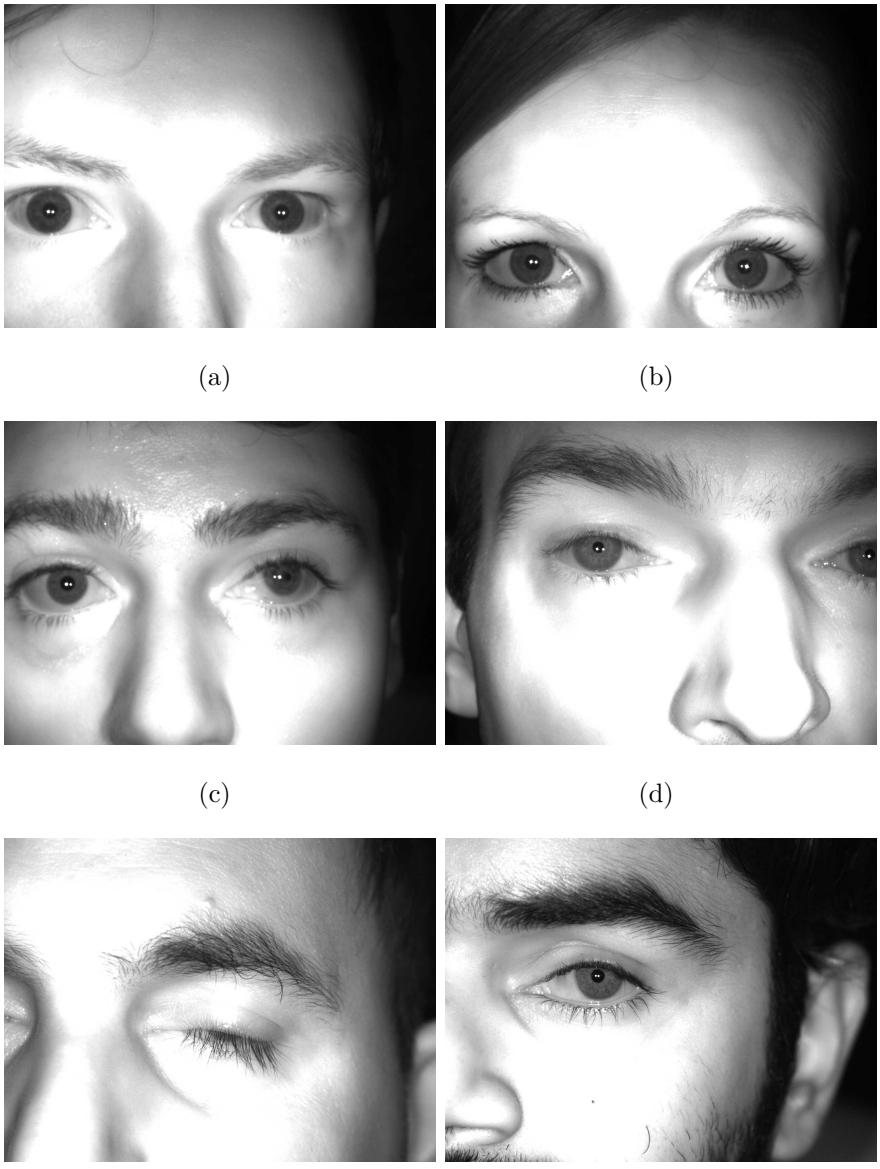
	low	high	ratio	FWHM [V]
Sobel filter	20	60	3.0	1.7
FFT	1,200	4,300	3.6	3.6
Daugman	40	280	7.0	1.5

Table 3.1: Performance values of auto focus assessment solutions. The filter Daugman suggested is clearly the best option. Its ratio of 7.0 shows the biggest discriminability of the peak and the base among the set. It also has the lowest *full width at half maximum (FWHM).*

Chapter 4

Database

The utilized self-recorded database was created using the first available *Basler Automotive Camera* (see 3.2.1) and two *OSRAM High Power IR LEDs SFH 4780S* (see Figure 3.5), in order to indicatively measure the performance of the system. For absolute performance values, it would be necessary to create such a database for any possible system setup and for a huge variance of environmental conditions. Nevertheless, this database provides the possibility to measure the impact on the performance for any change in the algorithm. Therefore, all changes as well as all performance evaluations were calculated by using this database. Altogether, the database holds more than $15,000$ images and consists of 27 subjects with a big variety of ethnological backgrounds and ages. This sums up to more than $1.1 \cdot 10^8$ possible comparisons. During the recording process it was ensured that there is also a certain variance in the images, regarding sharpness, distance, gaze, illumination, gender and eye color. Due to the blinking, there are fully closed and partially closed eyes, too. The database also contains subjects who had eye surgeries or lesions and even one with a glass eye. Some samples from the database are depicted in the Figures 4.1 and 4.2 and in Appendix C.

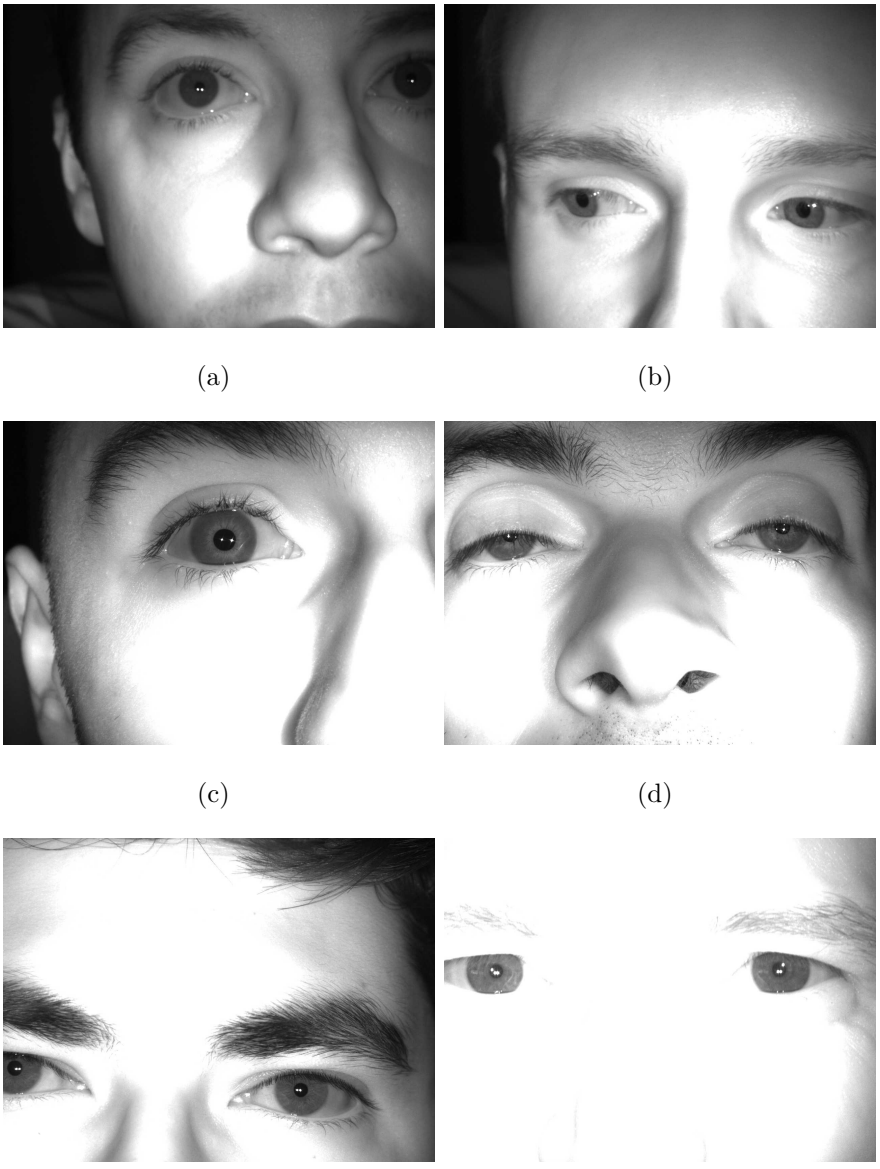


(a) (b)

(c) (d)

(e) (f)

Figure 4.1: Samples from the self-recorded database. The images in this database contain eyes with a big variance in sharpness, distance, gaze (Figure 4.2(b)), illumination, gender, eye opening, eye color, as well as eyes that had some kind of eye surgery (Figure 4.2(f)), were injured or even replaced by a glass eye (Figure 4.1(c)). For further pictures from this database see Appendix C.



(a) (b)

(c) (d)

(e) (f)

Figure 4.2: Samples from the self-recorded database. The images in this database contain eyes with a big variance in sharpness, distance, gaze (Figure 4.2(b)), illumination, gender, eye opening, eye color, as well as eyes that had some kind of eye surgery (Figure 4.2(f)), were injured or even replaced by a glass eye (Figure 4.1(c)). For further pictures from this database see Appendix C.

Chapter 5

Iris Recognition

The human iris is a thin circular diaphragm, which lies between the cornea and the eye lens. In order to perform a biometric identification of individuals the very unique patterns in human irises can be used. These patterns show a very high independence [16], even for genetically identical twins [14]. The iris pattern is one of the most stable features of the human body throughout a persons lifetime [41]. Combined, these points make iris recognition a biometric technology that offers potentially low failure rates at high recognition rates by a non-intrusive scanning. Figure 5.1 shows a sketch of the eye region with the iris inside it. The pupil is located in the center of the iris. The white region surrounding the iris is called sclera. The function of the iris is to control the amount of light that enters the pupil. This is adjusted by the sphincter and the dilator muscles, which are able to adjust the size of the pupil. In most cases the iris, as well as the pupil is circular, but does not form perfect circles [12]. Thus, both are more similar to ellipses than to circles. The iris has an average diameter of 1.2 cm and the pupil can take up between 10 % and 80 % of this space, dependent on the intensity of the illumination [11]. The region of the iris close to the border to the pupil, where the pattern is most

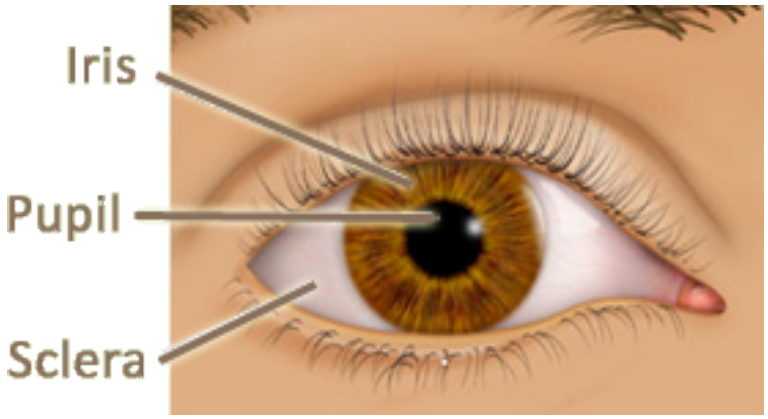


Figure 5.1: Sketch showing the eye region with the iris. The pupil is located near the center of the iris. The white region surrounding the iris is called sclera. The iris, as well as the pupil, are circular but are more similar to ellipses than to circles. The iris has an average diameter of 1.2 cm. The pupil can take up between 10 % and 80 % of this space [35].

dense, is called collarette. As a result, the information density is higher there. Accordingly, the collarette contains more iris information than other parts of the iris [55]. Iris recognition technology typically operates in the near infrared (NIR) spectral band, as most corneal specular reflections can be suppressed there. Moreover, even irises that appear very dark or black in the visible range reveal rich iris textures in the NIR band [11]. Therefore, additional NIR illumination is recommended (see chapter 3), which also enables the system to be operated by night. Further information about the anatomy of the iris or the complete eye region can be found in [60]. The rough steps leading to a successful iris recognition are depicted in Figure 5.2. The process begins with recording the image. John Daugman states that for a proper working iris recognition the minimum amount of pixels that should depict the iris measures up to 70 pixels on its radius [12]. Optimally, the optical camera axis is aligned with the optical eye axis and the camera system is properly focused on the iris.

Figure 5.2: Rough overview of iris recognition steps. It begins with recording and *preprocessing* the image. Afterwards, the *segmentation* takes care of finding the iris in the picture. Subsequently, the *feature extraction* produces a comparable *IrisCode* [13]. Finally, the *matching process* tries to determine the identity of the person by a comparison with a given database [35].

In some cases *preprocessing* has to be done in order to correct the eye gaze and to optimize the contrast or the gamma values in a way that more images become usable (see 5.1). Afterwards, the eye and the iris have to be located. This is managed by the *segmentation* step (see 5.2). In order to ensure that the eyelids, eyelashes and possible reflections will not be considered as part of the iris data, these are marked as noise (see 5.3). Unfortunately, in some cases the *segmentation* fails to properly find the iris. For catching the worst cases, a quick *segmentation quality check* can be performed (see 5.4). Subsequently, after the *normalization* step (see 5.5), the *feature extraction* (sec. 5.6) analyses the unique iris pattern and produces a comparable *IrisCode* [13]. Finally, the *matching process* (see 5.7) compares the created *IrisCode* with a given database and, in case of a match, determines the identity of the person [35].

5.1 Preprocessing

The images from the self-recorded database (see chapter 4) show a certain variance in sharpness, gaze, illumination and eye color. For removing portions of this variance and to allow the further processing of these images, certain preprocessing steps and checks can be applied. In the absence of an *auto focus* lens (see 3.3), a *Sharpness Check* (see 5.1.1) can measure the blurriness of a picture, which can serve as an indicator whether an image is processable at all. In case it is too blurry, *Sharpening Filters* (see 2.4) can be an option. For removing some of the impact of different illuminations and eye colors the *Z-Transform* (see 2.3) and the *CLAHE Filter* (see 2.2) can help. Finally, it is also possible to remove the eye gaze from the images. This technique is adressed in section 5.1.3.

5.1.1 Sharpness Check

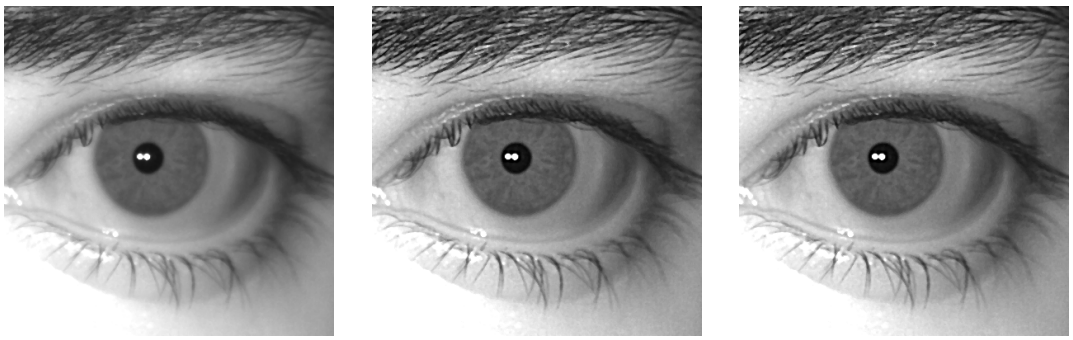
In case the system does not include an auto-focus functionality (see 3.3) or for the purpose of generally verifying whether the eye is well focused, a *Sharpness Check* can be performed. Since the subjects rarely remain completely motionless, the recorded images often appear blurred, which results in less or not usable iris data and therefore increases the error rates. The purpose of the *Sharpness Check* is to eliminate the blurriest pictures at an early stage. Blurriness is most commonly measured by the amount of edges in an image. Hence, the edge detection methods from the *auto focus* (see 3.3) can provide the required measures. Since different subjects have different quantities of natural wrinkles and differently distinct eye sockets, the amount of detected edges varies strongly [57]. These peculiarities are differing in a way that it even becomes possible to utilize them as features for periocular recognition

(see chapter 6). That is also the reason why it is hardly possible to directly determine the sharpness of a single image. A solution for this problem is to measure the impact of a *Median Filter* (see 2.5) and a *Laplace Filter* (see 2.4) on the image's variance. For the two ratios of the variance before the filtering to the variance after it, proper thresholds can be found in order to decide on the sharpness of the picture. Table 5.1 shows how the performance changes if the *Sharpness Check* is applied. Of course, the segmentation rate and as the minimum segmentations per eye drop but stay at an acceptable level. However,

	minimum combined error rate (CER)	min. FRR w/o FA	min. Segmentations per eye	Segmentation rate	Average intra inter distance
Without Sharpness Check	$6.1 \cdot 10^{-2}$	0.348	40	0.762	0.185
With Sharpness Check	$4.2 \cdot 10^{-2}$	0.305	23	0.698	0.194

Table 5.1: Performance values for the *Sharpness Check* using a randomly chosen subset of the self-recorded database (see chapter 4) with 150 images per subject. It is important to note that the results are only comparable within this table. The chosen key figures are described in chapter 8. Of course, the segmentation rate and the minimum segmentations per eye drop, but stay at an acceptable level. However, the minimum combined error rate (CER), the minimum false rejection rate (FRR) without any false acceptances (FA) and the average distance between the intra and inter class distributions improve, which proves the efficaciousness of the approach.

the minimum combined error rate (CER), the minimum false rejection rate (FRR) without any false acceptances (FA) and the average distance between the intra and inter class distributions improve, proving the efficaciousness of the approach. If the *Sharpness Check* recognizes a blurred image, the sharpness can be enhanced by filtering the picture with *Sharpening Filters* (see 2.4). Unfortunately, this method only results in better visibility for the human eye, as it is of course not possible to create information out of nothing. For that reason, images that were artificially sharpened happen to decrease the overall recognition performance. This might be the case, as the sharpening process also introduces some kind of noise pattern, which would be taken as iris data by mistake. The two images on the right side of Figure 5.3 show the effect of the usage of the two *Sharpening Filters* from 2.4 has, by applying them to an image from the self-recorded database (see chapter 4) that is depicted on the left side.

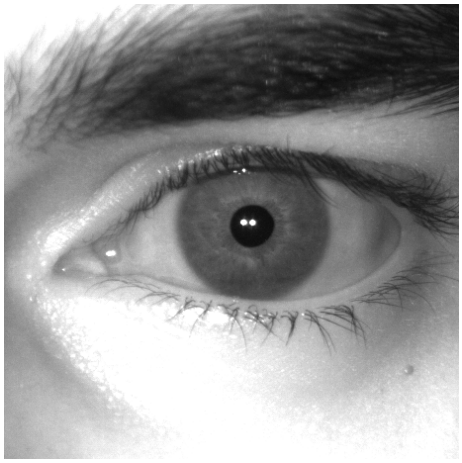


(a) original image (b) *Unsharp masked* image (c) *Laplacian filtered* image

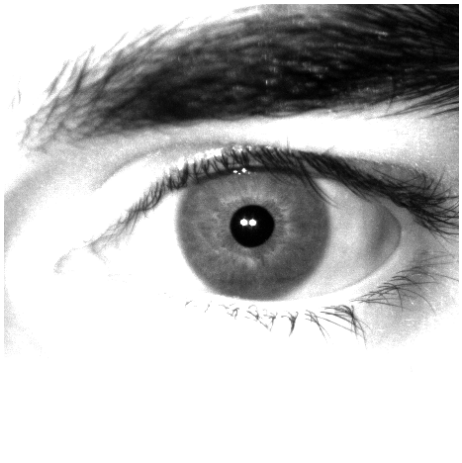
Figure 5.3: The two images on the right show the effect of *Sharpening Filters* from 2.4 on an image from the self-recorded database (see chapter 4).

5.1.2 Brightness Invariance

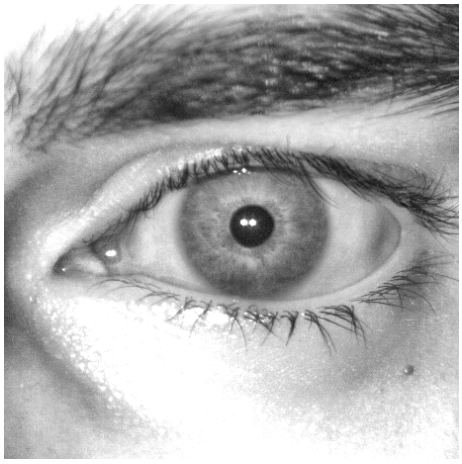
In order to achieve *Brightness Invariance*, it is required to remove portions of the effect from different illuminations and eye colors. Thereby, the goal is to reduce the variance in brightness between the images. In case the irises in the images are constantly too bright or too dark, a gamma correction can be applied to remove that offset. In a more universal but fragile gamma correction approach, it is checked whether the iris itself is over- or underexposed. In most cases, the first distinct peak in the picture's histogram provides information about the wanted intensity range, which gives a hint on how the gamma correction has to be applied [35]. Two alternative approaches are given by the *Z-Score Transform* (see 2.3) and *Contrast Limited Adaptive Histogram Equalization (CLAHE)* (see 2.2). Their impact is depicted in Figure 5.4, which contains the original image 5.4(a), the *Z-Score* transformed image (Figure 5.4(b)), the *CLAHE'd* image (Figure 5.4(c)) and an image that was *Z-Score* transformed as well as *CLAHE'd* (Figure 5.4(d)). The *Z-Score* transformed image shows an increased contrast in the iris region as well as for the eyelashes. In the *CLAHE'd* image the shadow parts from the original image have almost disappeared and the overall contrast is improved. Despite the usage of contrast limiting, the noise was enhanced as well. The final image simply combines all the effects of the two techniques. Similar to the sharpening process shown in Figure 5.3, the mentioned techniques should not be generally applied to any incoming image, as the introduced noise would reduce the recognition rates. Instead, these techniques should be used for cases in which the pictures are actually not usable due to, e.g. really bad illumination conditions, which otherwise cannot easily be processed any further by the algorithm. Applying these methods can enable the successful handling of such images, although with a lower performance than for high quality images.



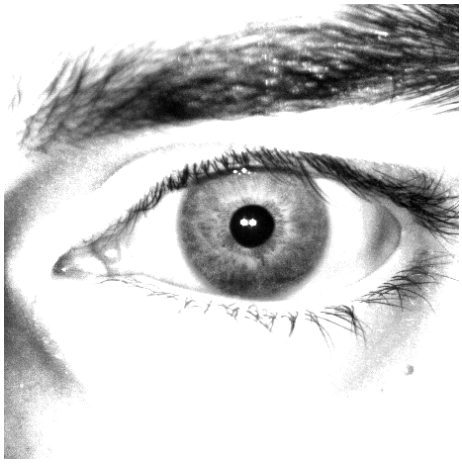
(a) original image



(b) *Z-Score* transformed image (see 2.3)



(c) *CLAHE'd* image (see 2.2)



(d) *Z-Score* transformed (see 2.3) and *CLAHE'd* (see 2.2) image

Figure 5.4: Impact of the *Z-Score Transform* (see 2.3) and *CLAHE* (see 2.2) on an image from the self-recorded database (see chapter 4). The *Z-Score Transform* boosts the contrast in the complete iris region. *CLAHE* removes the shadow parts from the original image and improves the overall contrast. Despite of the contrast limiting, noise is still being enhanced. The fusion of the *Z-Score Transform* and *CLAHE* combines all the mentioned effects.

5.1.3 Eye Gaze Removal

The subjects shown in the self-recorded database (see chapter 4) mostly failed to perfectly align their eye axis with the optical axis of the camera lens. This results in a certain quantity of eye gaze. The more gaze, the worse the recognition rates become and the fewer images will be successfully processed. Furthermore, it is quite unpractical to have a camera placed directly in the center of the field of view in cars, as people of course need to be able to see where they are driving to and what is happening in front of their car. As a consequence, the camera has to be placed somewhere else (e.g. attached to the inside mirror), which involves the introduction of a constant gaze. Certainly, it would be applicable to look into the camera but this would demand an additional action from the driver, compared to the recent key-less-go systems. People demand as much comfort as possible, especially from the vehicles of premium manufacturers, which are most likely the first to establish such a sophisticated biometric authentication system as replacement for the car key. The following technique deals with the removal of eye gaze or more precisely the transformation of the respective images to processable ones. For the development, a database of eight different fixed gaze positions was recorded. It contains $1,169$ images with at least 100 images per direction, including top, bottom, left, right, top left, top

	number of images	segmented images	ratio
without gaze removal	1,169	52	0.04
with gaze removal	1,169	1,138	0.97

Table 5.2: Segmentation rates for the *Gaze Removal* using the *Hough Circle Transform* (see 5.2.1). The process is able to transform most irises into circles.

right, bottom left and bottom right gaze. For each position the parameters for a *Perspective Transformation* [62] can be found. Table 5.2 shows the segmentation rates for the application of this correction method. The segmentation rate is drastically increased from 0.04 to 0.97, which allows the assumption that the correction is working properly. Ideally, the adjustment morphs the irises into circles. This is why the *Hough Circle Transform* (see 5.2.1) was used for the performance evaluation. It is important to mention that this technique does not enable good recognition rates between different positions. Only images from the same position produce a decent *false rejection rate (FRR)*. Nevertheless, this technique is the optimal solution for gaze removal on the inside of vehicles, as the location of the iris relatively to the camera is typical for any fixed camera positioning.

5.2 Segmentation

In order to create an iris template from a given picture, e.g. from the self-recorded database (see chapter 4), the first step is to determine the position in the full picture where the eye is located. Therefore, a multistage eye detection using *Haar cascades* [65] was used. For increased robustness in case of people wearing glasses or having bushy eyebrows, it is possible to use two different *Haar cascades*. Thereby, each cascade detects a defined pattern of geometric objects of a certain fixed size in a given distance and a defined ordering. Consequently, it is required to use them in different scales to detect differently sized eyes. This results in an eye detection approach that is quite reliable. It only fails to detect the presence of an eye in 1.7 % of the images in the self-recorded database (see chapter 4) and there are only very few false positives, like ears or nostrils with eyeish patterns.

After the rough localization of the eye, it is crucial to precisely segment the iris, which is described in the following section. The easiest way to do so is a simple thresholding approach. Thereby, the pixels in a certain range are taken as result. Unfortunately, this approach did not succeed, since the variance in human iris colors, as well as in the environmental illumination conditions is too huge to employ a stable segmentation using this technique. A fast alternative is the *Hough Circle Transform* (see 5.2.1). As it remains quite modest in terms of computational demands, it is suggesting itself for the use with relatively slow car electronic control units (ECUs). One drawback of this technique is that it only detects circles, whereas two ellipses are needed (see chapter 5). Possible workarounds can be additional *ellipse fitting* or the *Snake Algorithm* in 5.2.2. More direct, yet computationally demanding ways, were tried with a *Segmentation in the Polar Representation* (see 5.2.3) as well as with using a *Unet* for the Segmentation (see 5.2.4).

5.2.1 Hough Circle Transform

If it is assumed that the iris and the pupil can be approximated by circles (see 5.2), the *Hough Circle Transform* (see 2.6) (using the *Probabilistic Hough Transform* [40]) offers itself as a fast and computationally undemanding approach. Therefore, a lot of effort was spent to optimize the performance of the previous implementation [35]. The outcome is a multistage detection for iris and pupil, which is able to overcome problems with correctly segmenting some people's irises. These issues occur for people with really strong circular patterns in the eye region besides the iris and pupil or for such that have had some injury to the eye that interrupts the circularity of the iris or pupil. One example image with a subject having an interrupted iris shape is depicted in Figure 5.5. This image was taken from the self-recorded database (see chapter 4). The solution is to use different *preprocessing* steps (see 5.1) like heavy *Median Filters* (see 2.5) that suppress the unnecessary fine granular parts of the

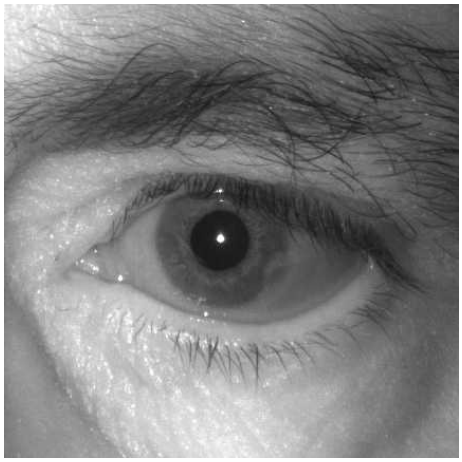


Figure 5.5: Image of an injured iris. The subject has had some injury that caused an interruption of the circular shape of the iris at the bottom right side. This image was taken from the self-recorded database (see chapter 4).

image, leaving only the low level features, such as the iris and pupil. Two more options for optimization are to allow different radii or to vary whether the pupil or the iris is initially searched for. In case a circle is found, the topology of the eye can be used to look for the remaining one in a smaller sub-picture. If such a second circle is detected as well, a check of validity can be done by comparing the parameters of the two circles, allowing only suitable combinations until a good segmentation is determined. In most cases, it is more straightforward to correctly find the pupil rather than the iris, as the pupil iris border is sharper and has higher contrast than the iris sclera border. Two possibilities to re-use the result of the *Hough Circle Transform* and to get better suiting ellipses are the *Snake Algorithm* (see 5.2.2) and the *Ellipse Fitting Extension*.

Ellipse Fitting Extension

Since the iris and the pupil are both ellipses (see chapter 5) rather than circles, the *Ellipse Fitting Extension* represents a possibility to re-use the outcome of the fast *Hough Circle Transform* (see 5.2.1) as a basis for searching for ellipses. Thereby, only the more accurate pupil location serves as basic positioning for the following fitting approach, which means that it can even be applied in case the algorithm has failed to segment the iris beforehand. A sub-picture containing the estimated pupil is used to perform *Canny Edge Detection* (see 2.6.1). Subsequently, the obtained edge information is taken as point data in order to fit an ellipse to those points. The same procedure is repeated with a partial image showing the left and right borders of the iris to the sclera. It is worth to notice that the top and bottom borders are neglected, as in most cases the eyelids and eyelashes, which are located in this place, would only add distortions. The final results are two ellipses, which are a better suiting representation for pupil and iris than circles.

5.2.2 Snake Algorithm

As mentioned before (see 5.2.1), the *Snake Algorithm* provides a possibility to use circle segmentations as input and calculate a much better suiting elliptical segmentation (see chapter 5). The less the camera axis and the optical eye axis are aligned, the more elliptical the appearances of iris and pupil are and the less accurate the optimum circle segmentation is [35]. *Snake Algorithms* are image processing tools that belong to the group of active contour algorithms. Some rough input is needed as starting point. Therewith the exact contours are adjusted iteratively. The fitting process makes the points look like the movement of a snake, whence the name *Snake Algorithm* is originating. The utilized implementation had initially been developed by Dominik Senninger in order to find the structural parameters of carbon nanotubes [58]. It was adapted and sped up in such a way that it is able to fit eye structures in the given images (see chapter 4) in a decent amount of time. The actual fitting process is done by iteratively minimizing the sum of the so-called inner and outer energies. The initialization is done by passing an ordered set of points p_i to the algorithm, which are somewhere close to the borders. In the following an overview is given for what the particular energies are responsible and how they are associated to one another. In each iteration the energy term

$$E_i = \alpha E_{\mathrm{int}}(p_\mathrm{i}) + \beta E_{\mathrm{ext}}(p_\mathrm{i}) \tag{5.1}$$

is calculated for the neighborhoods of the given points p_i (see Figure 5.6), whereby the inner energy $E_{\mathrm{int}}(p_i)$ only depends on the shape of the contour and the outer energy $E_{\mathrm{ext}}(p_i)$ just depends on the image properties of the respective neighborhood. The factors α and β give a weight to both energies. After this calculation the point p_i is moved to the place in the neighborhood,

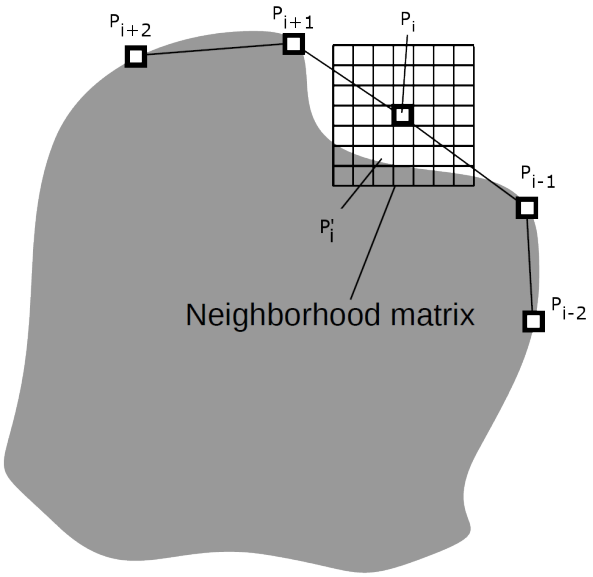


Figure 5.6: An example for the movement of a point p_i of the snake. At the point p'_I an energy minimum is located because of the high contrast [35, 58].

where the energy E_i has its minimum. The better the snake parameters are adjusted to the image conditions, the better the algorithm is able to fit the contours.

The inner energy $E_\mathrm{int}(p_i)$ is responsible to give a shape to the snake and to ensure that the distances between the points always remain similar. It is given by the weighted sum of the continuity energy $E_\mathrm{con}(p_i)$, which forces the contour to embrace an ordered shape and the balloon energy $E_\mathrm{bal}(p_i)$, which expands or shrinks the contour:

$$\alpha E_\mathrm{int}(p_\mathrm{i}) = \omega_\mathrm{c} E_\mathrm{con}(p_\mathrm{i}) + \omega_\mathrm{b} E_\mathrm{bal}(p_\mathrm{i})\,, \tag{5.2}$$

with ω_c and ω_b as the weighting parameters. The outer energy $E_{ext}(p_i)$ pushes the active contour towards the boundaries of an object. It is given by the

weighted sum of the intensity energy $E_{\mathrm{mag}}(p_{\mathrm{i}})$, which moves the contour to regions of higher or lower intensities and the gradient energy $E_{\mathrm{grad}}(p_{\mathrm{i}})$, which shifts the contour towards the edges in the image:

$$\beta E_{\mathrm{ext}}(p_{\mathrm{i}}) = \omega_{\mathrm{m}} E_{\mathrm{mag}}(p_{\mathrm{i}}) + \omega_{\mathrm{g}} E_{\mathrm{grad}}(p_{\mathrm{i}}) \,, \tag{5.3}$$

with ω_{m} and ω_{g} as the weighting parameters [58]. For the given eye image conditions, it turned out that the gradient energy $E_{\mathrm{grad}}(p_{\mathrm{i}})$ has the highest influence on the fitting of the iris. With respect to this impact, its weight ω_{g} was set to a much higher value than the other weighting factors. Figure 5.7 illustrates what the *Snake Algorithm* can theoretically offer. The yellow dots are the points that define the snake. Although the *Hough Circle Transform*

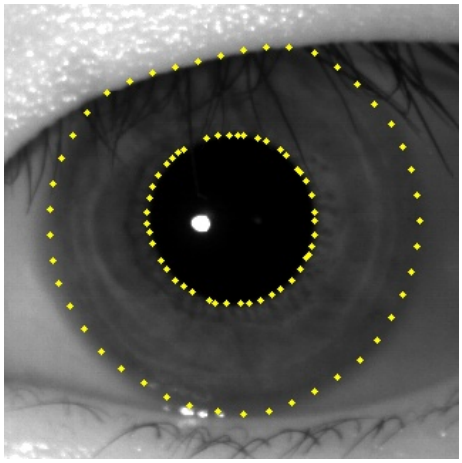


Figure 5.7: Iris segmented with the *Snake Algorithm*. This shows very well what is theoretically possible with the *Snake Algorithm*. The yellow dots are the points that define the snakes. Although the *Hough Circle Transfrom* (see 5.2.1) had also done well in segmenting this picture, the segmentation could even be improved. The elliptical form of the pupil is perfectly adjusted and the top eyelid is also excluded quite nicely [35].

(see 5.2.1) also performed well in segmenting this picture, the segmentation could be improved. The elliptical form of the pupil is perfectly adjusted and the top eyelid is excluded quite nicely, too. A big problem for this approach are the eyelashes, which cause the snake to take strange forms in many cases. A simple solution to this problem would be to increase the amount of points that define the snake. However, this exponentially increases the computation time, which makes it impractical to use in a real car scenario, due to the relatively low computational power of modern car ECUs. As a result, it was only possible to create a robust fitting process for the pupil, which is actually the more important part, since the information at the border between iris and pupil is denser than at the border between iris and sclera (see chapter 5). Figure 5.8 shows cases where the *Hough Circle Transfrom* (see 5.2.1) was not able to segment the pupil properly, whereas the *Snake Algorithm* succeeded in doing so. The three pictures on top (Figures 5.8(a), 5.8(b) and 5.8(c)), with the blue circles, illustrate the result of the *Hough Circle Transfrom*. The *Snake Algorithm* was initialized with these results and succeeded to find the much better fitting yellow ellipses on the three bottom pictures (Figures 5.8(d), 5.8(e) and 5.8(f)). The result is that the *Snake Algorithm* performs well in improving the segmentation quality. This can as well be noted in the performance measures (see 5.2.5) [35, 58].

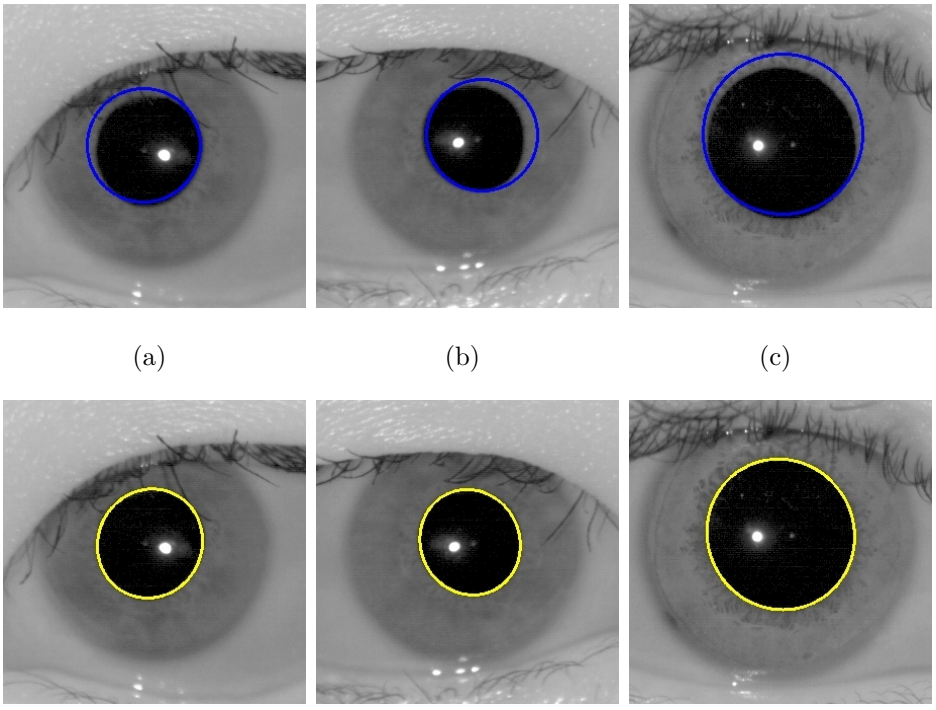
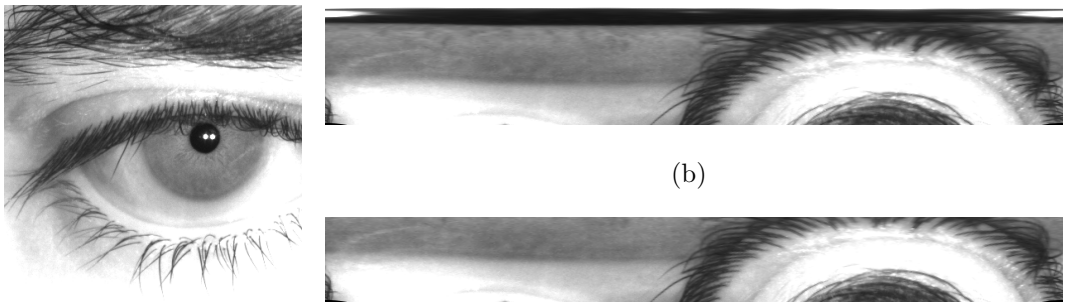


Figure 5.8: Visualisation of the *Snake Algorithm* in comparison to the *Hough Circle Transfrom* (see 5.2.1). The three pictures on top with the blue circles show the results of the *Hough Circle Transfrom*. The *Snake Algorithm* was initialized with these results and found the much better fitting yellow ellipses on the bottom. The result is that the *Snake Algorithm* comes with a good improvement in segmentation quality, which can as well be noted in the performance measures (see 5.2.5) [35].

5.2.3 Segmentation in the Polar Representation

This section introduces a new technique that is inspired by the work of Monteiro [42]. He, as well, presented a new technique that was trying to segment the iris in a polar representation of the eye, centered at the pupil. Thereby, it turned out to be much easier to produce better fittings therewith, than with the normal picture. This finding was used to develop a new, fast and stable technique. The starting point is the output of the multistage eye detection (see 5.2). In order be able to apply a conversion to the polar image respresentation, it is needed to find a roughly approximated center of the iris, respectively the pupil. Therefore, the *Hough Circle Transform* (see 5.2.1) comes with a sufficient accuracy in the pupil case, as it is only required to find a single point that lies inside the pupil. Another possibility would be to search for the largest dark region of the image, which fails more often, especially if people wear darkly framed glasses. A simplified form of the *Rubber Sheet Model* normalization (see 5.5) serves well for the conversion to the polar image if the radius is chosen in a way that it is guaranteed to be bigger than the iris radius. Figure 5.9 shows an image of an eye that is part of the self-recorded database (see chapter 4) and the two polar representations that are used for the segmentation of the pupil, respectively the iris. In Figure 5.9(b) the topmost part is the region of interest. A relatively big black region that belongs to the pupil is visible, as well as a white part originating from the reflections of the used NIR LEDs (see Figure 3.5). With the help of thresholding and a large-scale median filter (see 2.5), the brightest parts of the image are blacked, which allows to reliably get rid of the reflections. Subsequently, in each column the first point from the top along the radial axis with a significant intensity gradient is chosen. These points are exactly located at the border between pupil and iris. After a simple outlier detection, the points are transformed back to the normal image coordinate

system, in order to fit an ellipse to them, which is selected as the final pupil fitting. Similar to the proceedings in the pupil case, the image depicted in Figure 5.9(c) is used to find a proper iris segmentation. It can be obtained by cutting off a bit more than the black pupil part of Figure 5.9(b), after the pupil is successfully located. In contrast to the pupil, the iris is interrupted by the eyelid in most cases. As a consequence, the process has to be adapted to these conditions. Again, a large-scale median filter (see 2.5) is being used. This time to get rid of all the interfering details like eyelashes. The points of each column along the radial axis with the highest intensity gradient are gathered and checked with a more complex outlier detection. The priority thereby is to ensure that the points' positions do not vary too much from the surrounding ones. This procedure kicks out most of the points on the part of the image, where the eyelid is covering the border between iris and sclera. The remaining ones can be removed by neglecting those points that have no



(a) (b) (c)

Figure 5.9: Pictures of the *Segmentation in the Polar Representation*. The image was taken from the self-recorded database (see chapter 4). The Figures 5.9(b) and Figure 5.9(c) depict the polar representations that are used for the segmentation approach of the pupil, respectively the iris. The segmentation of the pupil is way more straightforward as the iris border is interrupted by the eyelid in most cases.

neighboring points, as well as too small groups of connected points. In the case of Figure 5.9(c) only points on the left side, directly at the location of the border between gray and white – corresponding to iris and sclera – would not be neglected. Similar to before, the result is mapped back to the original image, in which the final ellipse can be fit to the detected points.

5.2.4 Unet Segmentation

Unets belong to the group of *Convolutional Neural Networks (CNNs)*, which are inspired by the animal visual cortex [28] and therefore are most commonly used to analyze images. As cortical neurons only respond to input of a certain part of the visual field, known as the receptive field, the layers of *CNNs* can be interpreted as image filters. The first layers are filters that are sensitive to low level features, whereas the last ones respond to high level features. *Unets* were invented by Ronneberger et. al. in 2015 [53] as an improvement of Long et. al.'s work about *Semantic Segmentation* [36]. Figure 5.10 shows an example of an *Unet* architecture. The contracting path follows the typical architecture of *CNNs*. Thereby, layers of two 3×3 convolutions with *Rectified Linear Unit ReLU* activation functions [44] and 2×2 max-pooling operations for down sampling are repeated. Each down sampling step doubles the number of feature channels. The expansive path uses 2×2 up-convolution layers that halve the number of feature channels, as well as a concatenation with the corresponding cropped contracting feature maps, and again, two 3×3 convolutions with *Rectified Linear Unit ReLU* activation functions [44]. Finally, a 1×1 convolution maps the remaining feature channels to the desired amount of output segmentation maps, corresponding to the number of classes that shall be segmented [53]. In order to be able to use an *Unet* for the segmentation of irises, the first step was to manually label some of the data. As Ronneberger et. al. [53]

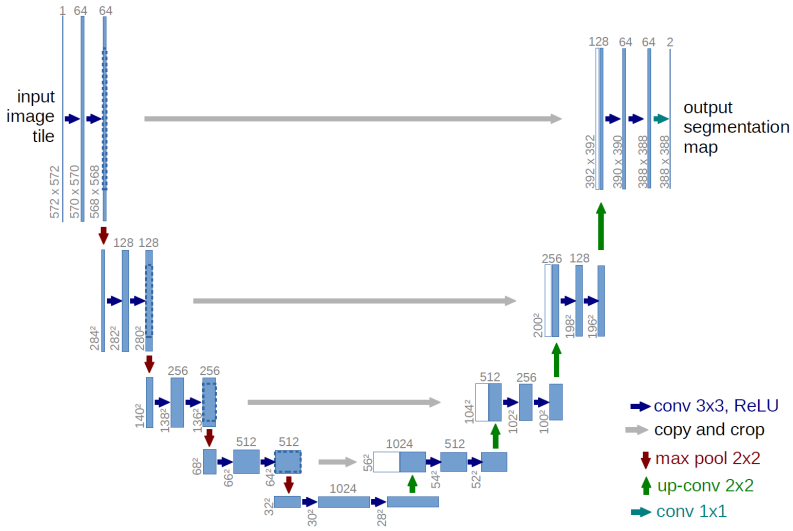


Figure 5.10: Example *Unet* architecture [53]. The contracting path follows the typical architecture of *CNNs*. Thereby, repeated layers of two 3×3 convolutions with *Rectified Linear Unit ReLU* activation functions [44] and 2×2 max-pooling operations are used for down sampling. Each down sampling step doubles the number of feature channels. The expansive path uses 2×2 up-convolution layers resulting in halving the number of feature channels, as well as a concatenation with the corresponding cropped contracting feature maps and again two 3×3 convolutions with *Rectified Linear Unit ReLU* activation functions [44]. In the end a 1×1 convolution maps the remaining feature channels to the desired amount of output segmentation maps, corresponding to the number of classes that shall be segmented [53].

managed to create an architecture, which is able to learn from relatively few samples, this was stopped after 620 labeled images. A thorough process of optimizing the parameters led to an architecture with an input of 128×128 pixels, 16 initial filters and a depth of four contracting respectively expansive steps. The best accuracy on the test set (15 % of the labeled data) that could be achieved was 99.2 %. As the net is unfortunately still delivering some false positives, a smart contour detection checks the validity of the outcome. If the result is not describing a two dimensional torus, it is neglected. Otherwise the points of the inner and outer contours are used to fit ellipses for the pupil and the iris, respectively. The final segmentation is obtained after a last check of validity regarding the suitability of the combination of the ellipses' parameters.

5.2.5 Performance

For the measurement of the performance of the available segmentation methods, a randomly chosen subset of the self-recorded database (see chapter 4) with 150 images per subject was used. For the computations, only the respective techniques were altered, whereas all other parameters were left untouched. This proves to be a good way to compare the techniques, but the specific outcome values are not meaningful enough for comparisons to other results. Table 5.3 contains the outcome of the performance measures. The key figures that were chosen therefor are the segmentation rate, the minimum segmentations per eye, the minimum false rejection rate (FRR) without false acceptances (FA) and the average distance between the intra and inter distributions. These are described in chapter 8. For all the *Hough Transform* (see 5.2.1) measures, the *Ellipse Fitting Extension* was used, as it boosts the segmentation rate and the minimum segmentations per eye with only a little decrease in the recognition performance and no effect on the average distribution distance. All in all, the *Hough Transform* without *Snake Algorithm* is the fastest approach, but as well the one with the worst overall recognition performance. Among all tested algorithms, it is also most prone to errors, due to the fact that it only detects circles. The *Snake Algorithm* (see 5.2.2) helps to noticeably boost the minimum false rejection rate (FRR) without false acceptances (FA) from 0.471 to 0.277 ($-70\,\%$) and the average distance between the intra and inter distributions from 18.3 to 20.2 ($+10\,\%$), without having any impact on the number of segmentations. The *Segmentation in the Polar Representation* (see 5.2.3) features the highest minimum segmentations per eye as well as the best segmentation rates. This means it has the most stable segmentation performance for all subjects. Accordingly, it is the most robust method in terms of reliably segmenting a higher amount of yet unknown subjects. Unfortunately, the mini-

mum false rejection rate (FRR) without false acceptances (FA) and the average distance between the intra and inter distributions are comparably bad. As the technique experiences some interference by the illumination's reflection points, the *Snake Algorithm* is able to improve the recognition performance, without significantly dropping the segmentation rate. Finally, the *Unet Segmentation* (see 5.2.4) has the best overall performance, but is as well the technique that is computationally most costly. It has a decent segmentation rate and a good value for the minimum segmentations per eye. The minimum false rejection rate (FRR) without false acceptances (FA) is quite low and it excels in the average distance between the intra and inter distributions. The results for additionally performing the *Snake Algorithm* do not show distinct improvement. This suggests that the *Unet* has learned really well how to segment the iris. A conspicuous aspect is that all the segmentation rates are within a range of 76.1 % to 80.9 %. This could suggest that there are up to 19 % low-quality images in the database, which cannot be segmented at all. In summary, the results suggest to use either the *Segmentation in the Polar Representation* for a robust segmentation with relatively small computational demands or the *Unet Segmentation* for the best overall performance if the available computing power of the ECU is sufficient therefor. The *Snake Algorithm* can be used to improve the performances of the *Hough Transform*, as well as the *Segmentation in the Polar Representation*, but should not be used in the combination with the *Unet Segmentation*. In the theoretical case of no computational boundaries, a combination of the segmentation techniques could be used. Thereby, only outcomes that are similar for all techniques are further processed, in order to assure that no bad segmentations slip through.

		Segmentation rate	minimum Segmentations per eye	minimum FRR w/o FA	Average intra inter distance
Hough Transform	without snake	0.761	40	0.471	0.183
	with snake	0.761	40	0.277	0.202
Segmentation in the Polar Representation	without snake	0.809	69	0.639	0.183
	with snake	0.807	68	0.507	0.190
Unet Segmentation	without snake	0.795	62	0.368	0.213
	with snake	0.795	62	0.443	0.219

Table 5.3: Performance values for the *segmentation* using a randomly chosen subset of the self-recorded database (see chapter 4) with 150 images per subject. For the computations, only the respective techniques were altered. All other parameters were left untouched. It is important to note that the results are only comparable within this table. The chosen key figures are described in chapter 8. The results suggest to use either the *Segmentation in the Polar Representation* (see 5.2.3), boosted by the *Snake Algorithm*, for a robust segmentation and relatively small computational demands, or the *Unet Segmentation* (see 5.2.4) for the best overall performance if the available computing power of the ECU is sufficient therefor.

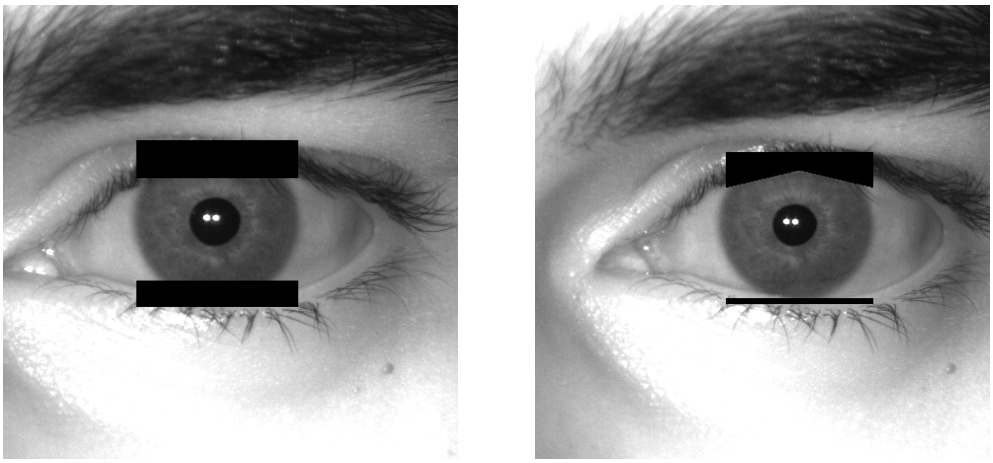
5.3 Noise Removal

It is regularly the case that there is some kind of noise within the region of a successful segmentation. In this context, noise means something that occludes the actual iris pattern, namely eyelids, eyelashes and reflections from the NIR LEDs (see Figure 3.5). As these disturbances do neither act as distinctive features nor are in any way constant, it is crucial for a good overall recognition performance to remove as much noise as possible in a way that as little as possible good iris data is wasted. Similarly to the *segmentation* (see 5.2), a simple thresholding approach alone cannot deliver satisfying results, as there are many cases in which the intensity range of the eyelashes overlaps the range of the iris. Therefore, setting the static threshold in a robust way is quite tricky, as it strongly varies from person to person. Nevertheless, it was used as an addition to all the following approaches, in order to remove some of the strongest disturbances, which are definitly not belonging to the iris, by setting the darkest and lightest regions of the image to zero, which was used as a marker for noise. It makes no significant performance difference which value is used for this purpose. This was evaluated for black and white, as well as the image average intensity and random values for each pixel. In the following, several techniques for noise removal will be addressed. This includes an approach that uses a *Hough Transform* (see 2.6) to search for one or two lines that belong to the eyelids' borders and therewith remove the noise. Another solution is a *Variance Based Removal* (see 5.3.2), which removes noise using a sliding variance window. Typically, the variance is high in the border regions, which includes the eyelashes and eyelids. The *Canny Based Removal* (see 5.3.3) applies an adaptive version of the *Canny Edge detection* (see 2.6.1), in order to use the edge information to determine the location of noise. Finally, an *Adaptive Thresholding* (see 5.3.4) approach is investigated. This technique

provides a way to overcome the limitations and problems of simple thresholding. It applies different thresholds in small areas of the image. Therefore, the problem of defining a static value for the whole picture vanishes.

5.3.1 Hough Transform

This approach is using the *Hough Line Transform* (see 2.6.2) to search for lines in the top and the bottom parts of an iris sub-picture and draws black bars that will, if working correctly, cover the overlapping region. The detected lines are most likely part of the eyelids' borders. The lowest coordinates of the lines for the upper eyelid and, respectively, the highest coordinates for the lower eyelid are used as boundary for a black bar that is drawn from that point



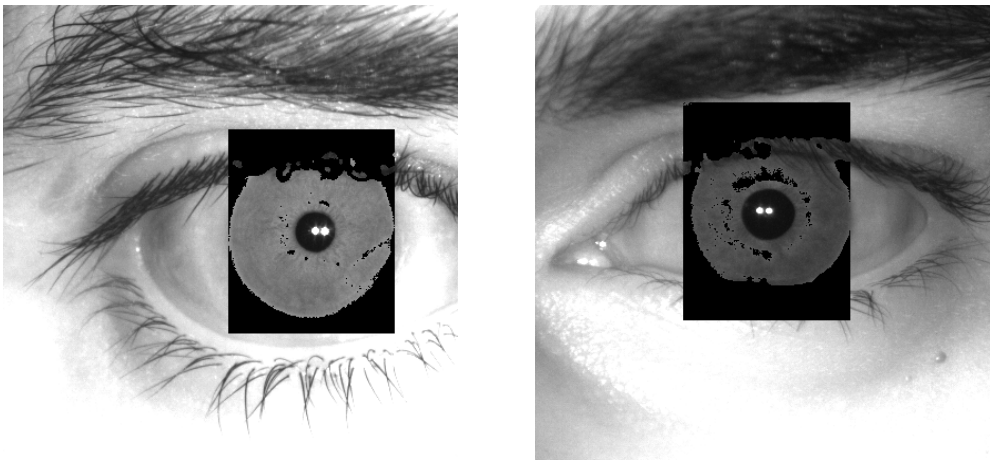
(a) (b)

Figure 5.11: Noise removal using the *Hough Line Transform* (see 2.6.2). In the image on the left side, the eyelids were successfully removed, but huge portions of the iris's data are lost as well. Different from the usage of only one line, two lines have a more efficient area of removal, like it is depicted in the image on the right side. Both images were taken from the self-recorded database (see chapter 4).

outwards to the iris borders and away from the pupil, ensuring proper removal of the eyelids. Figure 5.11(a) shows the outcome of the procedure. In this case, the occlusions were successfully removed, but huge portions of iris data are lost, too [35]. For a more efficient removal, it is possible to search for two instead of one line per eyelid – one on the left side and one on the right side. These lines are then connected and used as boundaries for the removal via two black triangles. Figure 5.11(b) depicts the outcome of this trick. Still, some useful parts of the iris are neglected, but fewer than before. As a final step, thresholding removes portions of the impact of eventually remaining specular reflections.

5.3.2 Variance Based Removal

In contrast to the noise removal with the *Hough Transform* (see 5.3.1), the *Variance Based Removal* does not only remove eyelids, but is also capable of directly handling the eyelashes. This technique uses a sliding window over an iris sub-image to determine the variance in small neighborhoods of the image. Regions with a significantly lower variance than the whole sub-image are marked as noise. In order to be more robust against varying conditions, the maximum allowed difference in variances is set inversely proportional to the output value of a sharpness measure. Figure 5.12 shows two images from the self-recorded database (see chapter 4), to which the *Variance Based Removal* was applied. For most subjects the process works very well (see Figure 5.12(a)). The eyelids, as well as the eyelashes, were properly removed with a good overall efficiency. Unfortunately, for some subjects like the one on the right, huge amounts of usable iris data are lost and still not all of the relevant noise is removed (see Figure 5.12(b)).



(a) (b)

Figure 5.12: Noise removal using the *Variance Based Removal* on two images that were taken from the self-recorded database (see chapter 4). The process works with a good overall efficiency for the subject on the left side because the eyelids, as well as the eyelashes, were properly removed. Unfortunately, for some subjects like the one on the right, huge amounts of usable iris data are lost and still not all of the relevant noise is removed.

5.3.3 Canny Based Removal

This noise removal method is based on an adaptive version of the *Canny Edge Detection* (see 2.6.1). Thereby, its parameters P_1 and P_2 are determined with the help of the image's median as

$$\begin{aligned} P_1 &= max(0, (1.0 - \sigma) \cdot median(I)) \\ P_2 &= min(255, (1.0 + \sigma) \cdot median(I)) \end{aligned} \tag{5.4}$$

for the image I and a parameter σ to determine the overall sensitivity to edges. In order to map the gained edge information to the parts of the image that are finally marked as noise, a combination of dilation and erosion [62] is applied. Figure 5.13 shows an image from the self-recorded database (see chapter 4), in

Figure 5.13: Successful noise removal using the *Canny Based Removal* on an image from the self-recorded database (see chapter 4).

which the *Canny Based Removal* was successfully applied. After the technique was implemented, it turned out that there were two problematic situations, in which it tended to fail. Firstly, if the image is so blurred that there are no sharp edges left, it does not filter anything and therefore keeps all of the noise as iris data. Secondly, in the rare case in which there are a lot of very sharp and strong iris patterns, the technique of course reacts to those sharp edges, which causes the amount of detected noise to be raised drastically, leaving very little iris data.

5.3.4 Adaptive Thresholding

Adaptive Thresholding is a technique that is able to overcome the limitations and problems of simple thresholding. It applies different thresholds in small areas of the image, depending on the intensity of that region. This makes the approach insensitive against varying brightness, originating from the light conditions as well as from the natural diversity of human bodies. Therefore,

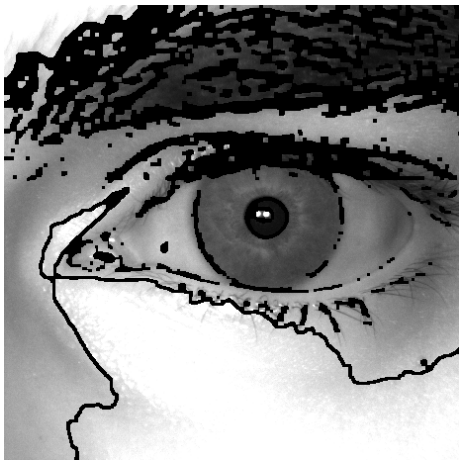


Figure 5.14: Successful noise removal using *Adaptive Thresholding* on an image from the self-recorded database (see chapter 4).

the problem of defining a static value for the whole picture vanishes. There are two different possibilities for determining the threshold of a neighborhood. At first, there is the *Adaptive Mean Thresholding*, which uses the mean of the neighboring area as a threshold. Secondly, the *Adaptive Gaussian Thresholding*, which uses a weighted sum of the neighborhood intensities as a threshold. Thereby, the weights are given by a gaussian window. After the *Adaptive Thresholding* itself was applied, a combination of dilation and erosion [62] helps to optimize the efficaciousness of the approach by removing small amounts of falsely detected noise in the iris pattern and solidifying the true noise area. Figure 5.14 depicts a successful noise removal using *Adaptive Thresholding* on an image from the self-recorded database (see chapter 4).

5.3.5 Performance

In order to measure the performance of the different noise removal methods, a randomly chosen subset of the self-recorded database (see chapter 4) with 150 images per subject were used. Between the measurements all parameters were fixed, only the method was changed. This allows a comparison between the tested techniques, whereas it is not meaningful to compare the results with values of other measures. Table 5.4 shows the outcome of the performance evaluations. For proper comparability the following key figures were chosen: the minimum combined error rate (CER), the lowest false rejection rate (FRR) without false acceptances (FA), the minimum segmentations per eye, the average distance between the intra and inter distributions and the signal to noise ratio. For a detailed description of the key figures see chapter 8. The results suggest to use the *Adaptive Mean Thresholding* (see 5.3.4), as this approach yields the best overall performance. It excels in the minimum combined error rate (CER) and the false rejection rate (FRR) without false acceptances (FA). It is worth noting that both methods using the *Hough Transform* (see 5.3.1), as well as the *Variance Based Removal* (see 5.3.2) have worse key figures than if no noise removal was used. All three are actually helping to prevent false acceptances (FA), but they do not manage to do this at a similar false rejection rate (FRR). It is also surprising that the *Hough Line Transform* with two lines performs slightly worse than the single line approach. This is induced by the general block-wise noise removal of these methods, which does not perfectly meet the demands. The single line option simply removes the complete region, whereas the double line approach tries to fit better to the eyelids. Thereby, it obviously leaves some noise in the middle of the eyelid as data and instead marks good iris data on the left and right side as noise. As a result, this interchange is not improving the quality of the noise removal and therefore has

bad influence on the performance. Furthermore, the goal of the double line approach to be more sparse with noise could also not be met since the signal to noise ratio is lower than the one of the single line approach. Maybe further optimization could lead to some improvement, but it seems unlikely to reach the performance of the pixel-wise removal methods. The values of the *Canny Based Removal* (see 5.3.3) are quite good, especially for the average distance between the intra and inter distributions. It also shows by far the highest sparsity among all of the techniques, as it has the highest signal to noise ratio. Among the two *Adaptive Mean Thresholding* approaches, the one using the mean to determine the local threshold performs better than the Gaussian one in all key figures, except the minimum segmentations per eye, which are slightly lower. Finally, it is worth mentioning that the results without noise removal are not as bad as one would expect. Therefore, this might be an option to, for example, reduce the computational load, without completely dropping the recognition rates, although the *Adaptive Mean Thresholding* is not too costly regarding its computational needs, either.

		min. CER	min. FRR w/o FA	min. Segmen-tations per eye	Average intra inter distance	Signal to noise ratio
Hough Transform	single	$8.0 \cdot 10^{-2}$	0.587	50	0.197	7.6
	double	$7.9 \cdot 10^{-2}$	0.649	49	0.191	5.0
Variance Based Removal		0.104	0.458	17	0.168	6.32
Canny Based Removal		$6.0 \cdot 10^{-2}$	0,456	32	0.235	29.3
Adaptive Thresholding	mean	$5.7 \cdot 10^{-2}$	0.262	36	0.213	8.2
	Gaussian	$6.0 \cdot 10^{-2}$	0.363	40	0.201	6.7
No Noise Removal		$6.5 \cdot 10^{-2}$	0.557	40	0.216	∞

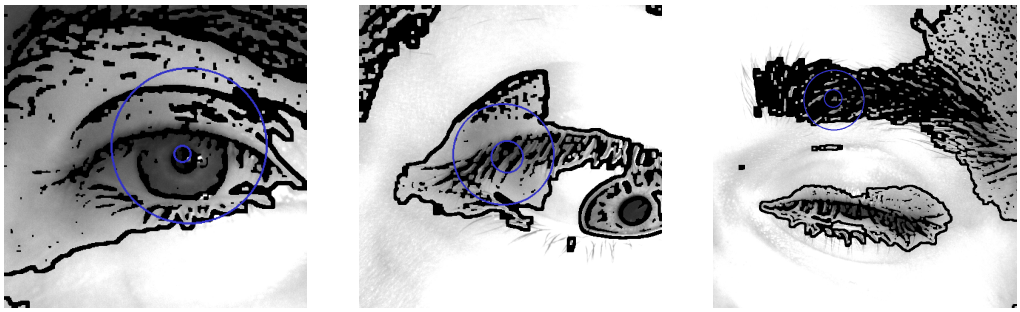
Table 5.4: Performance values for the *noise removal* using a randomly chosen subset of the self-recorded database (see chapter 4) with 150 images per subject. For the evaluations, only the respective techniques were altered, all other parameters were left untouched. It is important to note that the results are only comparable within this table. The selected key figures are described in chapter 8. The results suggest to use the *Adaptive Mean Thresholding*, as this approach shows the best overall performance. It excels in the minimum combined error rate (CER) and the false rejection rate (FRR) without any false acceptances (FA).

5.4 Segmentation Quality Check

In order to catch the worst segmentation faults, which would drastically decrease the algorithm's performance, *Segmentation Quality Checks* can be a solution. Thereby, the result of the segmentation is considered and rated in terms of its quality. It turned out that this task is tougher than expected and lots of approaches had to be aborted because they were either not robust enough or they were discarding too many good segmentations. Many good ideas had to be ruled out: A measure of the amount of strong edges inside the segmented area failed, since there was too much interference caused by eyelids, eyelashes and even by strong iris patterns. Therefore, it was not possible to establish a robust implementation. Similar problems were faced with approaches on checking the neighborhoods of border pixels for intensity differences, *Template Matching* [6, 57] as well as the use of a *Convolutional Neural Network* (see 5.2.4). Consequently, none of these attempts were able to provide adequate solutions. The difficulty of this task might also be associated to the fact that the rating of the segmentation quality strongly varies from one person to another. Of course, there are some clearly fatal segmentations, which can be detected by a simple count of shapes in the segmented region (see 5.4.1). But there are more fuzzy failures, too. These include under- and over-segmentation of the iris or the pupil. This means that there are either parts of the sclera or the pupil in the segmented area or that there are fractions of the iris that were not located correctly. The challenging part is to decide how big the falsely segmented area is allowed to become, before the segmentation should be neglected. However, a *Histogram Based Check* (see 5.4.2), which offers a decent robustness, compares the histograms of the segmented area to the inside and outside regions.

5.4.1 Shape Count Check

The *Shape Count Check* was implemented to ensure that none of the most fatal segmentations are slipping through. Fatal segmentations mostly occur if people have their eyes closed, did not align their eye axis with the camera axis properly or simply do not look into the camera and therefore have a huge eye gaze (see 5.1.3). In those cases either the *segmentation* step or the *noise removal* need to seriously fail in giving a proper result or yield no result at all, to produce a fatal segmentation. If they do, it causes huge portions of unwanted regions to be treated as an iris if no countermeasure is implemented to prevent this. The check is about counting the number of used gray scale values (shapes) in the detected region. Therefore, a histogram of the segmented area can be used. The results showed that if the amount of used shapes is bigger than usual, the probability for a fatal segmentation is high. For the self-recorded database (see chapter 4) a maximum of 185 out of 256 possible used shapes was a good threshold. The *Shape Count Check* is meant to filter exactly the failures that are depicted in Figure 5.15. In the first image (Figure 5.15(a)) the pupil is undersegmented, whereas the iris is oversegmented. In addition, the noise removal failed to appropriately remove parts of the eyelids as well as the sclera. For the second image (Figure 5.15(b)) one can argue what exactly caused the problems, as obviously more than one thing went wrong: the subject did not look into the camera, the eye detection did not capture the full iris, the segmentation reacted on circular patterns of the eyelashes or skin and by chance the relative positioning of the supposed iris and pupil were properly matched. For the last image (Figure 5.15(c)), similar to the middle one, an unfortunate series of events led to a fatal segmentation.



(a) (b) (c)

Figure 5.15: Three fatal segmentations detected by the *Shape Count Check* on images of the self-recorded database (see chapter 4). In the first image (Figure 5.15(a)) the pupil is undersegmented, whereas the iris is oversegmented. In addition, the *noise removal* failed to appropriately remove parts of the eyelids as well as the sclera. For the second and third images (Figures 5.15(b) and 5.15(c)), somehow unfortunate series of events led to these fatal segmentations. The *Shape Count Check* is meant to filter exactly the depicted failures.

5.4.2 Histogram Based Check

This strategy for assessing the quality of iris segmentations makes use of three image histograms. These originate from the area surrounding the segmented iris, the iris itself and the segmented pupil. It is important to mention that the images have to be reshaped to a fixed size and preprocessed with a brightness correction mechanism consisting of a *CLAHE Filter* (see 2.2) and a *Z-Score Transform* (see 2.3), in order to minimize the dependency on illumination and to ensure comparable relations between the histograms. The respective histograms do not need to really have 256 entries, corresponding to all possible intensity values. In fact, it improves the results if neighboring bins are combined by averaging down to 32 values, as the effect of small differences (e.g. noise induced) is removed thereby. Finally, the histograms need to be normalized to remove the impact of the vastly differently sized regions they build

upon. In order to gain knowledge of the particular segmentation quality, the distances $\mathcal{D}$ between the histograms of the iris $\mathcal{H}_\mathrm{i}$, the pupil $\mathcal{H}_\mathrm{p}$ and the outer region $\mathcal{H}_\mathrm{o}$ have to be calculated pairwise. Namely the distances between the outer part and the iris

$$\mathcal{D}_\mathrm{oi} = |\mathcal{H}_\mathrm{o} - \mathcal{H}_\mathrm{i}|\,, \tag{5.5}$$

the pupil and the outer part

$$\mathcal{D}_\mathrm{po} = |\mathcal{H}_\mathrm{p} - \mathcal{H}_\mathrm{o}|\,, \tag{5.6}$$

and the iris and the pupil

$$\mathcal{D}_\mathrm{ip} = |\mathcal{H}_\mathrm{i} - \mathcal{H}_\mathrm{p}|\,. \tag{5.7}$$

All three distances have to lie in certain predefined ranges, otherwise the segmentation is assumed to be of low quality. Other ways of comparing the histograms, like using a *Multi-Layer Perceptron Encoder* or a combination of three *Multi-Layer Perceptron Autoencoders* came with a bigger computational load and could not improve the quality score as much to be worth the longer runtime [57].

5.4.3 Performance

For the performance measures of the *Segmentation Quality Checks*, again a randomly chosen subset of the self-recorded database (see chapter 4) with 150 images per subject was used, as only the comparability between the methods is of interest here. For the evaluations, only the respective techniques were altered, all other parameters were left untouched. It is important to note that the results are only comparable within these measures. The chosen key figures, which were used for comparing the performances of the methods in Table 5.5, are the minimum combined error rate (CER), the lowest false rejection rate (FRR) without false acceptances (FA), the minimum segmentations per eye, and the average distance between the intra and inter distributions. For a detailed description of the key figures see chapter 8. With the two available methods, four combinations were possible. The results show that both checks together clearly achieve the best performance values. But the low value of 8 for the minimum segmentations per eye indicates that the system might be prone to failures for some people. In that specific case this means, there was a subject whose images were only segmented in 5.3 % of the cases. This is mainly the fault of the *Histogram Based Check* (see 5.4.2), as the *Shape Count Check* (see 5.4.1) has a minimum of 50 segmentations per eye (33.3 %). It is notable that the subjects with the least segmentations are not the same in all cases, as the minimum of 11 for the *Histogram Based Check* is reduced to 8 with the additional *Shape Count Check*. Therefore, it is suggestive to use both checks in scenarios that permit the possible drawback of slower identifications to ensure optimum security. For a convenient system the *Shape Count Check* is the best option, as it comes with a significant performance improvement, without reducing the minimum segmentations per eye.

	minimum combined error rate CER	min. FRR w/o FA	min. Segmen-tations per eye	Average intra inter distance
Both Checks	$6.7 \cdot 10^{-2}$	0.247	8	0.207
Histogram Based Check	$7.6 \cdot 10^{-2}$	0.302	11	0.202
Shape Count Check	$8.0 \cdot 10^{-2}$	0.587	50	0.197
No Check	$9.1 \cdot 10^{-2}$	0.610	50	0.190

Table 5.5: Performance values for the *Segmentation Quality Checks* using a randomly chosen subset of the self-recorded database (see chapter 4) with 150 images per subject. For the evaluations, only the respective techniques were altered, all other parameters were left untouched. It is important to note that the results are only comparable within this table. The chosen key figures are described in chapter 8. It is suggestive to use both checks in scenarios that permit the possible drawback of slower identifications, due to the low minimum segmentations per eye, to ensure optimum security, as this option comes clearly with the best performance values. For a convenient system the *Shape Count Check* is the best option, as it comes with a significant performance improvement, without reducing the minimum segmentations per eye.

5.5 Normalization

A properly segmented iris usually has a doughnut shaped form and has to be normalized in order to allow comparisons to other irises. Segmented irises differ in multiple parameters depending on the unique form of each iris, the head tilt, the imaging distance, angle and rotation, the illumination and the size of the pupil. Another problem is that the pupil as well as the iris region cannot always be described by simple circles (see chapter 5) and that the pupil is not always located exactly at the center of the iris. As a result, the same irises will have their characteristic features at different locations, without the normalization step. A proper method to map each segmented iris to the same constant parameters is needed and can be found in Daugman's *Rubber Sheet Model* [11]. By applying the normalization, the segmented iris is projected onto a dimensionless non-concentric polar coordinate system (r, θ), with the radius r and the angle θ, which ranges from 0 to 2π. The remapping from Cartesian coordinates (x, y) to the polar coordinate system (r, θ) is depicted in Figure 5.16 and done in the following way [11]:

$$I(x(r,\theta), y(r,\theta)) \rightarrow I(r,\theta) \tag{5.8}$$

with the image $I(x, y)$ of the segmented iris and the linear combinations $x(r, \theta)$ and $y(r, \theta)$ of the pupil boundary points $(x_{\mathrm{p}}(\theta), y_{\mathrm{p}}(\theta))$ and the iris boundary points $(x_{\mathrm{i}}(\theta), y_{\mathrm{i}}(\theta))$ along the angle θ:

$$x(r,\theta) = (1-r)x_{\mathrm{p}}(\theta) - rx_{\mathrm{i}}(\theta) \tag{5.9}$$

$$y(r,\theta) = (1-r)y_{\mathrm{p}}(\theta) - ry_{\mathrm{i}}(\theta)\,. \tag{5.10}$$

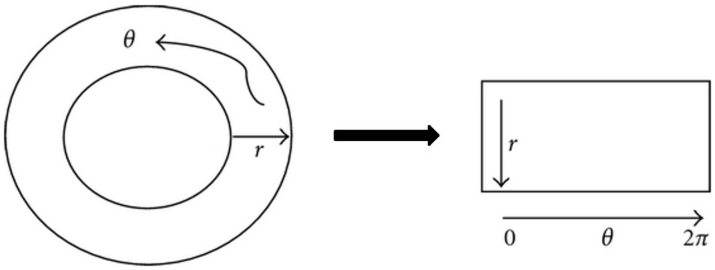


Figure 5.16: Daugman's *Rubber Sheet Model* [11]. The iris is projected from a Cartesian coordinate system (x, y) onto a dimensionless non-concentric polar coordinate system (r, θ) with the width r and the angle θ from 0 to 2π [35].

This assures a properly working projection that achieves invariance for position, size and dilation of pupil and iris. The invariance to rotation (see 5.7.2) can be achieved by rolling the resulting picture along the angle axis θ during the matching process (see 5.7). Applying the *Rubber Sheet Model* also has the benefit that the inner circles, where the information is denser (see chapter 5), are weighted more than the outer circles, where less information is located. In order to avoid an introduction of noise in case of an imperfect segmentation, the border regions are neglected [35]. Given the formula for the perimeter p

$$p = 2 \cdot \pi \cdot r \tag{5.11}$$

of a circle with radius r, the optimum ratio between angular and radial resolutions should be close to $2 \cdot \pi \approx 6.28$. Surprisingly, lots of trials, using a subset of the self-recorded database (see chapter 4) with 150 images per subject, showed that the best resolution is 16×128. Regarding the additional canceled inner and outer circles, this corresponds to a ratio of 7.1 rather than the expected 6.28. In order to apply the normalization, a number of evenly distributed data points, equal to the angular resolution, is selected on the detected iris and pupil

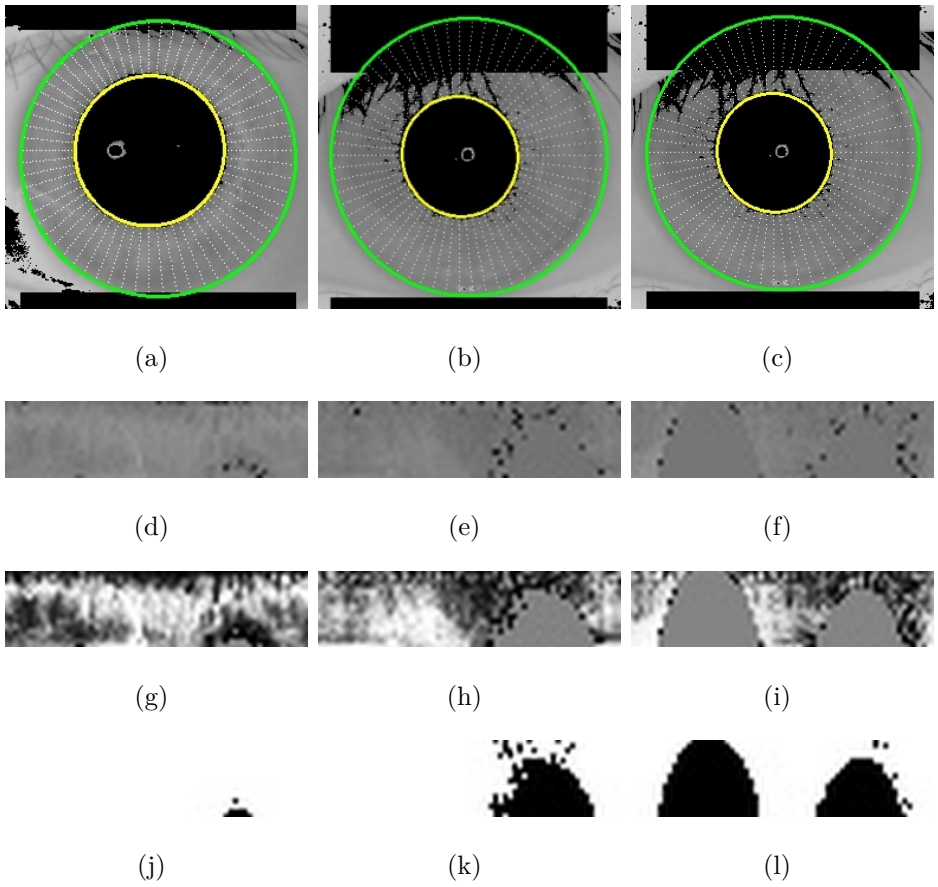


Figure 5.17: Visualization of the normalization process. On the top (Figures 5.17(a), 5.17(b) and 5.17(c)) the radial lines between the iris borders are depicted by white points. Subsequently, the areas defined by those points are used to conduct the remapping. The results with averaged black regions are shown in Figures 5.17(d), 5.17(e) and 5.17(f). Afterwards, a *Histogram Equalization* (see 2.1) is performed which drastically improves the contrast (see Figures 5.17(g), 5.17(h) and 5.17(i)). The Figures 5.17(j), 5.17(k) and 5.17(l) depict the respective noise arrays [35].

contours. The radial lines connecting respective points at the same angles are split in as many parts as the radial resolution defines, neglecting the borders. This creates boxes from which the information for the remapping process will be gathered. The locations of the noisy points are marked with `False` in a noise array and all other points `True`. Finally a *Histogram Equalization* (see 2.1) of the normalized array is performed, to optimize the usage of the available color space and thereby improve the contrast. For optimal impact, the black noise points are averaged beforehand. Figure 5.17 illustrates the normalization process. On the top (see Figures 5.17(a), 5.17(b) and 5.17(c)) the radial lines between the iris borders are marked with white points. The areas enclosed by four of those points are used to conduct the remapping. The resulting normalized arrays with averaged noise regions are shown in Figures 5.17(d), 5.17(e) and 5.17(f). In the end a *Histogram Equalization* (see 2.1) is performed. This drastically improves the contrast and renders the iris pattern nicely visible (see Figures 5.17(g), 5.17(h) and 5.17(i)). Without this step, the visibility is very limited. The parts where the noise is located in the normalized array before and after the *Histogram Equalization* are nicely visible as uniformly gray areas. For the sake of completeness the bottom Figures 5.17(j), 5.17(k) and 5.17(l) depict the respective noise arrays, which coincide with the concolorous areas in the pictures above [35].

5.6 Feature Extraction

Several possibilities exist for the iris *feature extraction* [2, 30, 68]. Most of them make use of a modified *Gabor Filter*, which was invented by Gabor [23]. The reason why the *Gabor Filter* is so widely used is that studies show that the human visual system processes visual information similarly [17]. In this section two methods shall be explained. These are the *Log-Gabor Filter* (see 5.6.1) employed by Masek [39] and Kahlil et al. [30] and the *2D-Gabor Filter* (see 5.6.2) proposed by Daugman [15, 16, 18]. Generally *Gabor Filters* can provide an optimal representation of a signal in space and spatial frequency. In order to construct a *Gabor Filter*, a sine wave or respectively a cosine wave is modulated by a Gaussian. The Gaussian provides localization in space, whereas the sine, or cosine, is perfectly localized in frequency. If the Gaussian is combined with the cosine it allows the construction of the real or symmetric filter. On the other hand, the combination with the sine yields the imaginary or odd filter. In order to specify the center frequency of the filter, the frequency of the sine, or cosine, can be adjusted, whereas the bandwidth of the filter is determined by the width of the Gaussian [35]. In common with the later on described *periocular recognition* (see chapter 6) the *Resnet50* (see 6.1.3.2) was as well established as feature extractor. But in case of the *Rubber Sheet Model* (see Figure 5.16) normalized iris images, its capability to separate and to unambiguously identify different subjects is not given at all. Therefore, it was decided to not address it further in this chapter.

5.6.1 1D Log-Gabor Filter

Standard *Gabor Filters* have a DC component [5] for bandwidths larger than one octave. That means that they depend on the average value of the signal. In order to avoid introducing a DC component it is possible to use a version of the *Gabor Filter* which is Gaussian on a logarithmic scale. This is called the *Log-Gabor Filter* and was introduced by Field [22]. Its frequency response $G(f)$ for the one-dimensional case is given by

$$G(f) = exp\left(-\frac{\left(log\left(\frac{\omega}{\omega_0}\right)\right)^2}{2\left(log\left(\frac{\Delta\omega}{\omega_0}\right)\right)^2}\right) \tag{5.12}$$

with the center frequency ω_0 and the bandwidth of the filter $\Delta\omega$. *Log-Gabor Filters* have to be designed in the frequency domain, as the singularity in the log function at the origin does not allow the construction of an analytic expression that describes the shape in the spatial domain. Just like the *2D-Gabor Filter* (see 5.6.2), the *Log-Gabor Filter* features a complex response. For the goal of splitting the real and imaginary parts, the response has to be transformed back to spatial domain, using a *Fourier Transform* [35].

5.6.2 2D-Gabor Filter

The complex-valued *2D-Gabor Filter* that Daugman [15, 16, 18] suggests can be described by

$$\begin{aligned} G(x,y) =& exp\left(-\pi\left(\frac{(x-x_0)^2}{\alpha^2}+\frac{(y-y_0)^2}{\beta^2}\right)\right)\cdot \\ & \cdot exp\left(-2\pi i\left(u_0(x-x_0)+v_0(y-y_0)\right)\right) \end{aligned} \tag{5.13}$$

in the Cartesian image domain (x, y), where (x_0, y_0) regulates the position in

the image, α and β determine the effective width and length, whereas u_0 and v_0 specify the modulation, which possesses a spatial frequency of $\omega_0 = \sqrt{u_0^2 + v_0^2}$ and the direction $\theta_0 = arctan\left(\frac{v_0}{u_0}\right)$. As the *normalization* (see 5.5) introduces a dimensionless non-concentric polar coordinate system (r, θ) with the *Rubber Sheet Model* (see Figure 5.16), it is needed to re-write the definition $G(x, y)$ for the filter as

$$G(r,\theta) = exp\left(-i\omega(\theta - \theta_0)\right) \cdot exp\left(-\frac{(r-r_0)^2}{\alpha^2}\right) \cdot exp\left(-\frac{(\theta-\theta_0)^2}{\beta^2}\right) \quad (5.14)$$

in the polar image domain (r, θ), where (r_0, θ_0) specifies the center frequency and location of the filter and ω is the filter frequency that is spanning three octaves. α and β co-vary in inverse proportion to ω. They are the multiscale 2D filter size parameters, which set the effective width and length and span an eight-fold range from 0.15 to 1.2 mm, which corresponds to the zones of analysis on the iris. It is suggested to slightly adjust the real parts of the filters in such a way that it results in zero volume and a removal of the DC component [35].

5.6.3 Phase Quantization

The output after applying a *Gabor Filter* is a complex valued amplitude information. Daugman [11] suggests to use only the phase information in order to produce an *IrisCode* [13]. Amplitude information is not very discriminating, since it strongly depends on image contrast, illumination, focus and camera gain [19], whereas the phase information provides the most significant information within an image [47]. Hence, a phase demodulation is employed. Figure 5.18 shows the phase quadrant coding sequence for such a demodulation. Each complex value determines two bits of phase information. The result is a binary

array containing the most discriminating information in order to distinguish between different irises. Another benefit is that comparisons of binary values are much less computationally and storage demanding than comparisons of complex values, which allows a much faster matching process (see 5.7) [35].

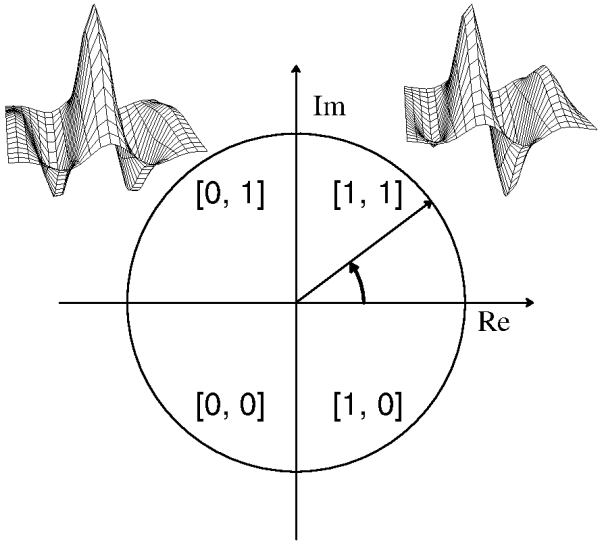


Figure 5.18: In the phase demodulation process the *IrisCode* [13] is generated. It contains the most significant information [47] in order to distinguish between the different irises of different subjects, the phase information. It is generated with the complex valued output of a *Gabor filtered* image. The depicted phase quadrant coding sequence is used to set two bits of phase information from each complex value [35].

5.6.4 Performance

In order to measure the performance of the two described methods, the *1D-Log Gabor Filter* and the *2D-Gabor Filter* were applied on a subset of the self-recorded database (see chapter 4) with 150 images per subject. Thereby, only the respective technique was changed, whereas all other parameters were left untouched. Table 5.6 shows the results of the measures. The chosen key figures are described in chapter 8. Both values clearly suggest that the *2D-Gabor Filter* outperforms the *1D-Log Gabor Filter*. If the performances are compared, the minimum false rejection rate (FRR) without any false acceptances (FA) is lowered by 33.0 % and similarly, the average distance between the intra and inter distributions is improved by 42.2 % for the *2D-Gabor Filter* over the *1D-Log Gabor Filter*. Taking into account that the computational demands do almost not differ between the two methods, it is fully conclusive to use the *2D-Gabor Filter* for all conceivable scenarios.

	minimum FRR without FA	Average intra inter distance
1D Log-Gabor Filter	0.348	0.185
2D-Gabor Filter	0.233	0.263

Table 5.6: Performance values for the *feature extraction* using a randomly chosen subset of the self-recorded database (see chapter 4) with 150 images per subject. For the evaluations only the respective techniques were altered, all other parameters were left untouched. It is important to note that the results are only comparable within this table. The chosen key figures are described in chapter 8. Both key figures suggest to always use the *2D-Gabor Filter*, as it clearly outperforms the *1D-Log Gabor Filter*.

5.7 Matching

The *matching process* handles the comparison of the output of the *feature extraction* to the beforehand created, labeled and saved data. Since the *feature extraction* produces bit-codes as output, the *Hamming Distance* [24] is the optimum metric to compare two templates and their corresponding noise masks. In order to achieve *Rotational Invariance* (see 5.7.2), one of the two questionable templates can be rolled before calculating the *Hamming Distance* again, always keeping the lowest result. The more frequently this is done, the higher the rotational invariance [35]. Due to the noise removal, every created template has a various amount of noise. As a result, the bit-wise comparisons between two of them build upon a different number of Bernoulli trials. In order to become independent from these differences and to be still able to set a static decision threshold for the statistical independence, the matching score has to be adjusted and normalized. This is managed by a *Score Normalization* (see 5.7.3). Finally, it is possible to adjust the calculation of the *Hamming Distance* in a way that the more distinguishing parts of the iris gain a bigger weight. Therefore, *Template Weighting* (see 5.7.4) was exploited.

5.7.1 Hamming Distance

The *Hamming Distance* [24] gives a measure of how many bits coincide between two bit patterns. In order to determine whether two iris templates represent the same or different irises, a decision based on a test of statistical independence can be conducted. If the two bit patterns are generated from the same iris, then the test fails. Daugman [13] uses the fast bit-wise XOR $\otimes$ and bit-wise AND $\cap$ operators for a very fast computation and gives the following equation in order to compute the *Hamming Distance* HD:

$$HD = \frac{\| (codeA \otimes codeB) \cap maskA \cap maskB \|}{\| maskA \cap maskB \|}. \tag{5.15}$$

This equation also disregards comparisons of noisy data points, which are marked with `True` in the noise masks. That means that only those bits of both iris patterns for which both noise masks correspond to `False` are used in the calculation. Binomial statistics state that 50 % of the bits should coincide if two irises are completely independent from each other, as independent patterns can be looked at as if randomly set. So there is a 50 % chance for `True` and the same for `False`. Therefore, the chance that a single bit is equal in both patterns is 50 %, which leads to a Hamming distance of 0.5. Comparing two patterns originating from the same iris results in low *Hamming Distances* values. Daugman and Downing [14] state that the probability of two different irises agreeing by chance in more than 70 % of their phase sequence is about one in 7 billion, which corresponds to the current world population. This means that a decision threshold of 0.3 for the *Hamming Distance* assures that theoretically there will be no chance for two people on earth to have irises that are so similar that they will pass this test of statistical independence [35].

5.7.2 Rotational Invariance

In order to ensure that the *matching process* is invariant to rotations of the iris, a certain amount of *Hamming Distances* is calculated, shifting one of the templates bit-wise to the left and right. The smallest occurring *Hamming Distance* is used as a result, as this value corresponds to the best matching between two iris patterns. Figure 5.19 shows an example for such a bit shifting process. The resulting *Hamming Distance* would be 0 in this case. The more shifts are done, the more rotation can be compensated. A shift of half of

the angular resolution in both directions allows the iris to be twisted by any amount of angular degrees and thus generates complete rotational invariance. In reality a shift by 30 degrees in both directions is more than sufficient. This corresponds to shifts up to one sixth of the angular resolution. As Daugman [11] states, this correction causes the peak of the binomial distribution for the random case to be altered from 0.5 to a value of about 0.45 [35].

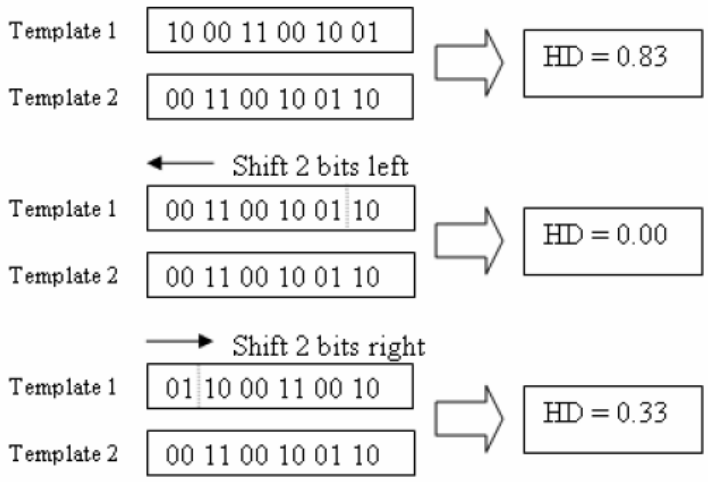


Figure 5.19: This picture shows an example for the shifting process that assures the *Rotational Invariance* of the *matching process*. The more shifts are done, the more rotation can be compensated. The smallest occurring *Hamming Distance* is used as a result, since this value corresponds to the best matching between two iris patterns. In this case the resulting *Hamming Distance* is zero [35].

5.7.3 Score Normalization

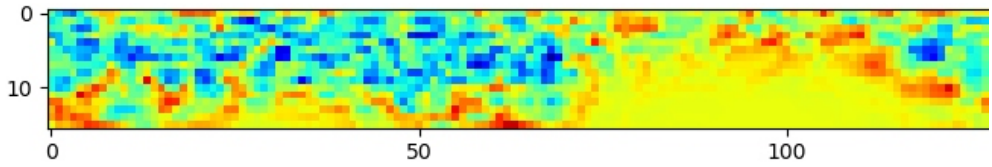
Every created template incorporates various amounts of noise that are determined by the *noise removal* step (see 5.3). As noisy bits are neglected, the bit-wise comparisons between two templates build upon different numbers of Bernoulli trials. In order to remove the impact and to become independent from this incompleteness, the matching score has to be adjusted and normalized. Subsequently, it is still sufficient to set a static decision threshold to decide upon the statistical independence and hence to identify people. Daugman [12] suggests to manage that by normalizing the raw *Hamming Distance* HD_{raw} into HD_{norm} by applying

$$HD_{\text{norm}} = 0.5 - (0.5 - HD_{\text{raw}})\sqrt{\frac{n}{m}} \tag{5.16}$$

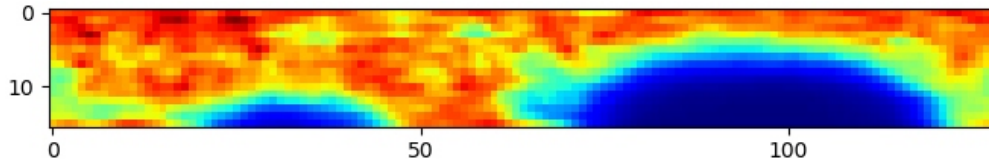
with the amount of bits n used to compute the raw *Hamming Distance* HD_{raw} and a scaling parameter m that should be set to a typical value of n. Therefore, it depends on the parameters that were used for the application of the *Rubber Sheet Model* (see 5.5). For the available method of *Adaptive Mean Thresholding* as *noise removal* method (see 5.3.4) and the optimum resolution parameters of 16×128, a value of $m = 3000$ was determined to be most convenient. Daugman suggests to use 911 or 960 in case the maximum value of n is 1024 [12]. One side effect of the *Score Normalization* is that the change of the average value to 0.45 due to the rolling of the templates to gain rotational invariance (see 5.7.2) is almost shifted back to the anticipated expectation value of 0.5 for Bernoulli trials.

5.7.4 Template Weighting

Template Weighting allows to give bigger weight to more distinguishing parts of the iris pattern and hence improve the recognition rates. Therefore, it was analyzed at which positions of the iris templates the bits differ most frequently. It has to be distinguished between *intra* class comparisons and *inter* class comparisons. In case of comparisons within the same class the reached distances should be as low as possible. Therefore, it would be best to boost the bits that are less discriminative and lower the impact of regions that increase the distances. On the other hand, for *inter* class comparisons, high distances are desirable. As a result, bits with a high discriminativeness should be weighted higher and the other way round. Figure 5.20 shows a heat map of how often bits differed in a full cross comparison of the self-recorded database (see chapter 4). Thereby, blue color represents for low differing frequency and red means that the respective bit differs quite often. For both heat maps, the impact of noise masking was considered and removed. Nevertheless, the blue parts of the inter comparison heat map coincide with the usual placement of the eyelids. It is remarkable that exactly these parts are the most discriminative ones for the intra case. As an admirable consequence, lowering the impact of these regions boosts both types of comparisons. Due to the exact distributions being specific to the database, it is not advisable to directly use them as weights, in order to avoid over-adaptation. Instead, it is much better to weight the lower quarter of the template less and the upper one higher in return. In terms of location, this results in a higher weight for the regions closer to the pupil and in lower weights for the most outside pixels, closer to the sclera (see chapter 5).



(a) *Intra* class comparison heat map



(b) *Inter* class comparison heat map

Figure 5.20: Heat maps of the discriminativeness of template bits for a full cross comparison of the self-recorded database (see chapter 4). Blue color means low frequency of differing and red means that the respective bit has differed quite often. The impact of the noise masking was considered and removed. Due to the nearly inverse distribution of discriminativeness, it becomes possible to boost the overall recognition rates for both cases, by weighting the lower quarter of the template less and the upper one higher in return.

5.7.5 Performance

To gain an overview of how the presented approaches achieve *Rotational Invariance* (see 5.7.2), the *Score Normalization* (see 5.7.3) as well as the *Template Weighting* (see 5.7.4) have an effect on the performance of the matching process. Once more a subset of the self-recorded database (see chapter 4) with 150 images per subject was used. Just like before, only the respective techniques were altered, while all other parameters were left untouched. Therefore, it is important to note that the results are only comparable within the respective tables. For all performance comparisons, the chosen key figures are the minimum false rejection rate (FRR) without false acceptances (FA) and the average distance between the intra and inter distributions, as they mirror best how the performance is changing if one of the techniques is applied. Both are described in chapter 8. First of all, Table 5.6 shows the results for the *Rotational Invariance*, for angles from zero up to 47.8°. The results show that the higher the desired amount of corrected rotation becomes, the lower the average distance between the intra and inter distributions will be. One more thing that the values suggest is that the minimum false rejection rate (FRR) without false acceptances (FA) is not becoming lower if more than 25.3° are corrected. This allows to conclude that the database does not contain images that are rotated by larger angles. As a general bottom line, it is important to set the value of how much rotation shall be corrected as high as needed but as low as possible for optimum performance. Table 5.8 shows the results for the *Score Normalization* and the *Template Weighting*. Both techniques independently increase the average distance between the intra and inter distributions. Therefore, it is not surprising that the combination of both methods results in an even increased distance, more precisely, from 22.5 to 26.3. For the minimum false rejection rate (FRR) without false acceptances (FA), the *Score Normalization* improves

amount of corrected *Rotational Invariance*	minimum FRR without FA	Average intra inter distance
47.8°	0.228	0.260
25.3°	0.228	0.273
8.4°	0.230	0.295
0.0°	0.248	0.302

Table 5.7: Performance values for the *matching process* with *Rotational Invariance* using a randomly chosen subset of the self-recorded database (see chapter 4) with 150 images per subject. For the evaluations, only the respective techniques were altered, all other parameters were left untouched. It is important to note that the results are only comparable within this table. The chosen key figures are described in chapter 8. The results show a decreasing average distance between the intra and inter distributions the more the rotation is corrected. Because the minimum false rejection rate (FRR) without false acceptances (FA) is not falling any more above a correction of 25.3°, it is possible to conclude that the database does not contain images that are rotated more.

the result, whereas the *Template Weighting* worsens it. For the combined case the result remains at a similar value as if using none of the techniques. As a result, it is advisable to either use only the *Score Normalization* for increasing the overall performance or both techniques combined for an optimized separation of the intra and inter distributions, with an almost stable minimum false rejection rate (FRR) without any false acceptances (FA).

	minimum FRR without FA	Average intra inter distance
None	0.231	0.225
Score Normalization only	0.219	0.233
Template Weighting only	0.276	0.253
both	0.233	0.263

Table 5.8: Performance values for the *matching process*, with *Score Normalization* and *Template Weighting*, using a randomly chosen subset of the self-recorded database (see chapter 4) with 150 images per subject. For the evaluations, only the respective techniques were altered, all other parameters were left untouched. It is important to note that the results are only comparable within this table. The chosen key figures are described in chapter 8. Both techniques independently increase the average distance between the intra and inter distributions. Therefore, it is not surprising that the combination of both methods results in an even increased distance, more precisely, from 22.5 to 26.3. For the minimum false rejection rate (FRR) without false acceptances (FA), the *Score Normalization* improves the result, whereas the *Template Weighting* worsens it. For the combined case the result stays at a similar value as without utilizing any of the techniques.

Chapter 6

Periocular Recognition

Iris Recognition is clearly one of the best non-invasive biometric technologies available, but there are cases in which it fails. Image capturing from a large distance and certain environmental factors may cause failures [3]. Therefore, periocular recognition – the recognition based on the area around the eye – offers a technique to overcome these obstacles, as it is insensitive to changes in distance, facial expressions or illumination conditions [52]. For the purpose of supporting iris recognition, periocular recognition is the most obvious solution, since the periocular area is being captured anyway by recording the iris. Therefore, it does not add a big computational load to the system. The first to use periocular recognition as a biometric were Park et al. [48] in 2011, but over the recent years a few more researches were conducted in this field, which suggest that the periocular region is unexpectedly discriminative [1, 31, 56]. Some of them have as well tried to fuse iris recognition and periocular recognition [45]. Prominent biometric features included in the periocular region (see Figure 6.1) are often of geometric nature like eyebrows, eyelids, eye corners, tear duct, sclera and also ratios, curvatures and angles between those elements as well as skin textures [31, 34, 67].

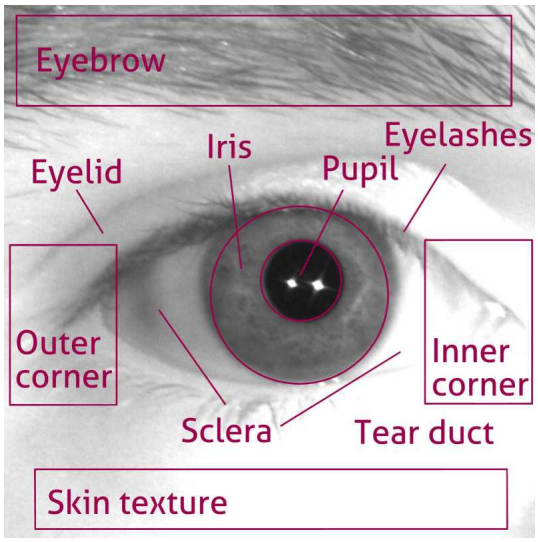


Figure 6.1: Periocular regions of interest [52]: features in the periocular region are often of geometric nature like eyebrows, eyelids, eye corners, tear duct, sclera and also ratios, curvatures and angles between those elements as well as skin textures, fine wrinkles or skin pores around the eyes [31, 34, 67]

6.1 Feature Extraction

In most cases the conducted research on periocular recognition uses databases that were recorded for face or iris recognition. This work makes use of the same approach, too, as the performance was evaluated using the available iris recognition database (see chapter 4). Similar to the iris segmentation step (see 5.2), a multistage eye detection approach utilizing a *Haar cascade* [65] is used to find subimages containing an eye in the self-recorded database (see chapter 4). The feature extraction methods that this thesis covers include *Local Binary Patterns (LBPs)*, *Z-Images*, a *Deep Belief Network* with three layers of *Restricted Boltzmann Machines (RBMs)*, and a *Residual Neural Network (ResNet)*.

6.1.1 Local Binary Pattern Histogram (LBPH)

Local Binary Patterns (LBP) are a popular gray-scale invariant texture measure and image descriptor for classification in the field of computer vision. They were introduced by Ojala et. al. [46] and have been improved by several researchers [59]. They are calculated by labeling the pixels of an image by thresholding the neighborhood of each pixel and interpreting the resulting binary code as a decimal. In the simplest approach the neighborhood consists of the 8 surrounding pixels, which results in decimals in the range of $[0, 255]$. More complex versions feature bigger radii and therefore larger neighborhoods. There exist also approaches to make the technique invariant to rotation, as well as to reduce the size of the feature space [38]. A mathematical description for the *Local Binary Patterns (LBP)* is given by

$$LBP(x_\mathrm{c}, y_\mathrm{c}) = \sum_{\mathrm{n}=0}^{\mathrm{N}-1} s(i_\mathrm{n} - i_\mathrm{c}) 2^n \,, \tag{6.1}$$

with $(x_\mathrm{c}, y_\mathrm{c})$ being the center point, N the neighborhood size, i_c the center pixel intensity, i_n the intensity of the n^{th} surrounding pixel and s defined as

$$s(x) = \begin{cases} 1, & \text{if } x \geq 1 \\ 0, & \text{otherwise}\,. \end{cases} \tag{6.2}$$

Figure 6.2 shows an example of how LBP-code is generated by applying the process in equation (6.1). The left side shows the pixel intensities of an example neighborhood. In the thresholding process, the center pixel's intensity is compared to the intensities of the neighboring pixels. The result, on the right side, is then read as an 8-bit binary number. Finally, the decimal value is calculated: $10110011_2 = 179_{10}$. This procedure is repeated for the whole

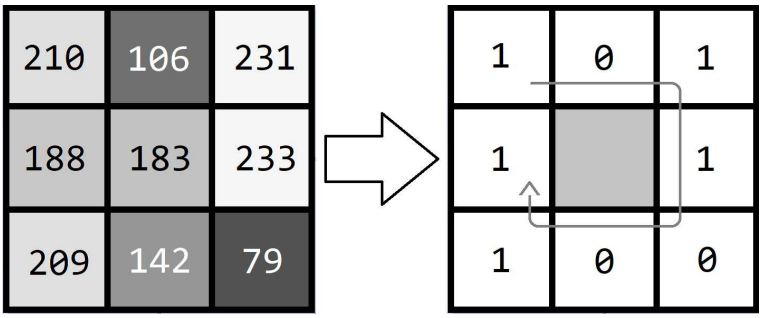


Figure 6.2: Example of generating LBP-code [52]. The left side shows the pixel intensities of an example neighborhood. In the thresholding process, the center pixel's intensity is compared to the intensities of the neighboring pixels. The result, on the right side, is then read as an 8-bit binary number. Finally, the decimal value is calculated: $10110011_2 = 179_{10}$. The full procedure can be described by equation (6.1).

image, in order to create the LBP-image. In order to create the *Local Binary Pattern Histogram (LBPH)*, the LBP-image is sliced into M equally sized tiles. For each of the tiles the respective histograms are used as the feature vectors. Thereafter, the full image feature vector is created by chaining the tile's feature vectors together. The length l of the resulting vector is given by

$$l = 2^N \cdot M \,, \tag{6.3}$$

with the neighborhood size N and the amount of tiles M. It can then be used to compare the image to other images by applying a similarity measure or using a classifier.

6.1.2 Z-Images

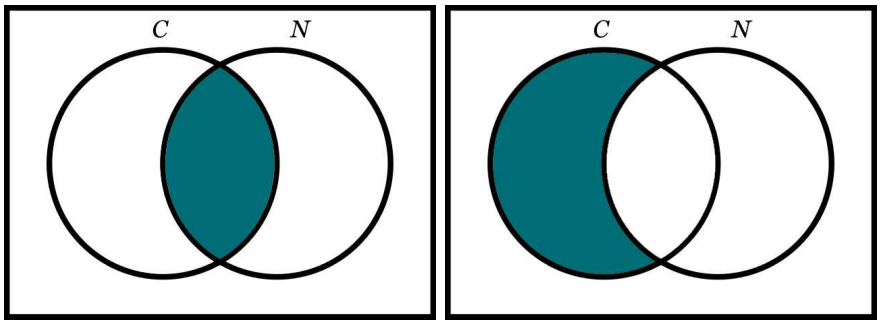
Vigneron et al. [64] try to address a weakness of *Local Binary Patterns (LBP)* (see 6.1.1) regarding the fact that the observed center pixel and its neighborhood are not necessarily independent of each other. Additionally, *LBPs* do not pay regard to the center pixels, which can lead to a failure in summarizing the local region. These problems make *LBPs* sensitive to noise in homogeneous regions of the image. Therefore, a new encoding mechanism that is independent of the neighborhood's size, as well as of the reading order, is introduced. It is based on Zeckendorf's Theorem [70], which states that every positive integer N can be uniquely decomposed into a sum of distinct Fibonacci numbers in such a way that the representation does not include two consecutive ones. Fibonacci numbers form the sequence

$$x(n) = x(n+1) + x(n-2) \tag{6.4}$$

for $n \geq 0$. In order to get the Zeckendorf representation, all pixels of a monochrome image (range of $[0, 255]$) are decomposed using the 12 relevant Fibonacci numbers $\mathcal{F}_{rel} = (1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233)$ [52]. A few examples of the utilized representations are shown in Table 6.1. In order to cre-

Value	Decomposition	Representation
210	$3 + 8 + 55 + 144$	$(3, 8, 55, 144)$
143	$2 + 5 + 13 + 34 + 89$	$(2, 513, 34, 89)$
79	$3 + 21 + 55$	$(3, 21, 55)$

Table 6.1: Examples for the Zeckendorf representation.



(a) Intersection (b) Difference

Figure 6.3: Visualization of the intersection and difference operators. The operators are used to compare binary Z-Representations of a center pixel C with a neighboring pixel N [52].

ate the resulting Z-Image, Zeckendorf represented neighborhoods of size 3×3 are monitored. For all neighborhoods the sets of the center pixels C are compared to the ones of the neighboring pixels N, using either the intersection or the difference operator. For a visualization of both see Figure 6.3. Among the eight comparison results of a neighborhood either the highest occurring Fibonacci number (quantization) or the maximum sum of Fibonacci numbers (contouring) is taken as the final Z-code for the center pixel. If the value equals zero in the quantization case, the original center pixel value is taken as the resulting Z-code. Following this procedure for all pixels respectively neighborhoods, the complete Z-image can be obtained. Similar to the *Local Binary Pattern Histograms* (see 6.1.1), the Z-image can be sliced into M equally sized tiles, in order to use the chained histograms of the tiles as feature vectors [52].

6.1.3 Deep Neural Networks

Deep Neural Networks are one of several tools in the field of Machine Learning that are able to learn data representations. These methods can be divided into different learning procedures: *supervised* and *unsupervised*. Additionally, a mixture of both – *semi-supervised learning* – exists as well. The difference of *Deep Neural Networks* to *Neural Networks* is that the first have a lot more hidden layers and hence incorporate an enormous inner structure. Nevertheless, it is not clearly defined how many layers are needed to call a *Neural Network* a *Deep Neural Network*. The technique itself had been known for a long time, but its huge success began with the wide-spread availability of graphical processing units (GPUs) with a high processing power. These GPUs consist of many independent processing cores, which allow to perform many parallel computation tasks. This results in a huge execution time boost for *Deep Neural Networks*, since the nodes within the layers resemble independent basic mathematical computations that can be fully parallelized. Nowadays, the large amount of application fields for *Deep Neural Networks* includes computer vision, natural language processing, medical image analysis and many more.

6.1.3.1 Deep Belief Network

Deep Belief Networks are a class of *Deep Neural Networks* and consist of multiple fully connected hidden layers with no connections within the layers. Thus, this means they can be seen as networks of stacked *Restricted Boltzmann Machines*, which can be found in many applications such as classification, feature learning or dimensionality reduction and can be trained supervisedly as well as unsupervisedly. *Restricted Boltzmann Machines* are composed of a visible and a hidden layer of binary-valued nodes, which have weighted connections to all nodes of the other layer but no connections within their own layers. In

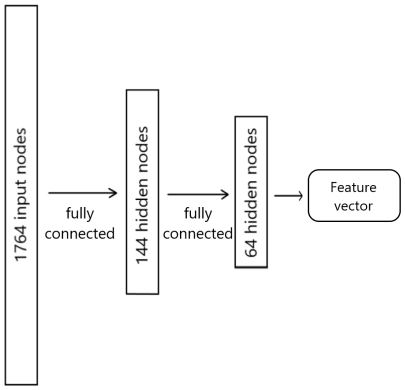


Figure 6.4: Deep Belief Network architecture [34]: stacked layers of *Restricted Boltzmann Machines* form the network. The input layer has $1,764$ nodes, which means that input images have to be resized to 42×42 pixels and flattened in order to be processed. The hidden layers consist of 144 and 64 nodes, respectively. The output of the last hidden layer is used as the feature vector.

this work a *Deep Belief Networks* built with two layers of *Restricted Boltzmann Machines* was used to learn the feature space of the periocular region, using the available database (see chapter 4). The hyperparameters were optimized by a thorough testing process [34], which led to the network architecture shown in Figure 6.4. The input layer has $1,764$ nodes, which means that input images have to be resized to 42×42 pixels and flattened in order to be processed. The hidden layers consist of 144 and 64 nodes, respectively. The output of the last hidden layer is used as the feature vector that can be saved in a database or directly fed into a classifier. It would be possible, but is not preferable, to add another layer that acts as a classifier, for example using the softmax or sigmoid functions [4]. Since the number of possible classes is not fixed, it is better to use a classifier that calculates the distance between two feature vectors [34].

6.1.3.2 Residual Neural Network

Residual Neural Networks (ResNets) are *Deep Neural Networks* inspired by the cells in the cerebral cortex. Similar to the structure of the human brain, a *ResNet* is implemented with layer skips, which are also called shortcut connections [25]. Figure 6.5 shows a building block for residual learning. Instead of learning the desired mapping $\mathcal{H}(x)$ directly, the layers learn a residual mapping $\mathcal{F}(x) := \mathcal{H}(x) - x$ instead [25]. A benefit that comes with such a skipping is that the problem of vanishing gradients is avoided. During the training of deep networks it can happen that updating the network's weights completely stops, as the gradient vanishes the higher the depth of the network becomes [26]. Due to the network design of *ResNets*, layers are reusing the activations from previous ones. It can be noted that this makes them actually an ensemble of relatively shallow networks. With less layers to propagate through, the gradient vanishing problem does not appear. The result is as well a speed up of the training phase. All the skipped layers are slowly restored in the course of the network learning the feature space. The combination of the already described *Convolutional Neural Networks (CNNs)* (see 5.2.4) with a *Residual Neural Network* can be called *Convolutional Residual Neural Network* and has

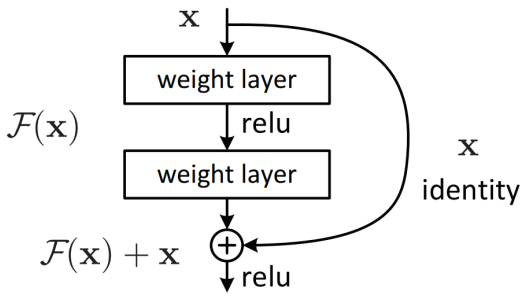


Figure 6.5: Residual learning: a building block [25].

won the *Large Scale Visual Recognition Challenge (ILSVRC'15)* [54]. As it is able to represent a huge portion of the respective feature space it has somehow learned well how to "see". This ability was exploited by many researchers in many different applications in the field of computer vision since the learned weights are publicly available [25]. The *ResNet* used in this work has 50 layers and is called *ResNet50* [25]. It needs an input of $224 \times 224 \times 3$, which corresponds to a three-channel 224×224 image. Therefore, the monochrome images have to be rescaled and converted to three channels to fit. In order to be able to work with the available weights as fast and easily as possible, the output layer is simply cut off. This leaves the output of the last filter as feature vector. It would also be possible to re-learn the last layer, but the first option suits the needs of biometrics better, in which it is crucial to have a fast adding and deleting of persons to and from the database.

6.2 Classifiers

Like in all biometrics, the output of the *feature extraction* (see 6.1) has to be compared to the previously created, labeled and saved feature vectors from the database. This operation is handled by the classifier, which decides how similar the incoming data is to the known persons. Features show a high similarity to data of the same class and a low one to other classes. In most cases a threshold is utilized to classify to which person the new data belongs.

6.2.1 Cosine Distance

One of the best known methods to compute the distance of two vectors is the *Cosine Distance (CD)*. It is used in information retrieval, text mining and data mining, for example. The cosine of the angle ϕ between two vectors $\mathbf{A}$ and $\mathbf{B}$ is given by

$$\cos(\phi) = \frac{\mathbf{A} \cdot \mathbf{B}}{\| \mathbf{A} \| \, \| \mathbf{B} \|} = \frac{\sum_{\mathrm{i}=1}^{\mathrm{n}} A_{\mathrm{i}} B_{\mathrm{i}}}{\sqrt{\sum_{\mathrm{i}=1}^{\mathrm{n}} A_{\mathrm{i}}^2} \sqrt{\sum_{\mathrm{i}=1}^{\mathrm{n}} B_{\mathrm{i}}^2}} \,, \tag{6.5}$$

with the components A_{i} and B_{i} of $\mathbf{A}$ and $\mathbf{B}$. The results range from -1 to 1, with 1 indicating parallelism, 0 orthogonality and -1 the highest possible dissimilarity. In order to ensure a proper distance metric, negative values have to be prevented. Therefore, the equation for the *Cosine Distance* that was used in this work is defined as

$$CD = 1 - \cos\left(\phi\right). \tag{6.6}$$

6.2.2 Jensen-Shannon Divergence

For the use of measuring the similarity between two probability distributions, the *Jensen-Shannon divergence (JSD)* offers a tool that is utilized in the fields of probability theory, machine learning and statistics. It is also called the total divergence to the average [10] and is a smoothed and symmetrized *Kullback-Leibler Divergence (KLD)*. For two discrete probability distributions P and Q in the same probability space the *KLD* and *JSD* are defined as follows:

$$KLD(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x) \log \left(\frac{P(x)}{Q(x)} \right) , \tag{6.7}$$

$$JSD(P \parallel Q) = \frac{1}{2} KLD(P \parallel M) + \frac{1}{2} KLD(Q \parallel M) , \tag{6.8}$$

with M being the mixture distribution of P and Q given by

$$M = \frac{1}{2}(P + Q) . \tag{6.9}$$

The square root of the *JSD* is a metric and is also called the *Jensen-Shannon Distance* [21]. In the case that the two distributions P and Q are exactly the same, the *JSD* equals to 0. Otherwise, for the binary logarithm it holds that

$$0 \leq JSD(P \parallel Q) \leq 1 . \tag{6.10}$$

6.3 Performance

To compute the performance of the periocular recognition, once more a randomly selected subset of the self-recorded database (see chapter 4) with 150 images per subject was used, as only the comparability between the methods is of interest in this context. For the computations, only the respective techniques were altered, all other parameters were left untouched. It is important to note that the results are only comparable within these measures. The key figures that were chosen for comparing the performances are the minimum combined error rate (CER), the false acceptance rate (FAR) and the false rejection rate (FRR) at the threshold with the lowest combined error rate (CER) and the average distance between the intra and inter distributions. For a detailed description of the key figures see chapter 8. All possible combinations of feature extractors (see 6.1) and classifiers (see 6.2) were checked. All resulting performances are shown in Table 6.2. Unsurprisingly, the *ResNet50* (see 6.1.3.2) shows the best overall performance and has the lowest combined error rates (CER). The errors are distributed in a way that it is amongst the lowest values for the false acceptance rates (FAR), as well as for the false rejection rates (FRR). Only regarding the average distance between the intra and inter distributions and in case of the cosine classifier it is beaten by the *contouring Z-images approaches* (see 6.1.2). Nevertheless, taking into account all key figures, the gap to the *Local Binary Patterns* (see 6.2) and the *contouring Z-images approaches* (see 6.1.2) is much smaller than expected. Whereas the *Deep Belief Network* (see 6.1.3.1) and the *quantization Z-images approaches* (see 6.1.2) are distanced a bit more. Compared to the other techniques, the *Local Binary Patterns* have combined error rates CER that are among the smallest. But the errors are distributed in a way that the false rejection rates FRR are small, whereas the false acceptance rates FAR are not. Additionally, it has

the lowest capability to separate the intra and inter distributions of all used feature extractors. Like the *Local Binary Patterns*, the *contouring Z-images approaches* have comparably small combined error rates (CER). But this time the false rejection rates (FRR) are higher, whereas the false acceptance rates (FAR) are among the smallest, except for the difference contouring method using the *Jensen-Shannon-Divergence*. For the *Cosine Distance*, the *contouring Z-images approaches* have the largest average intra and inter distances among the tested combinations. The *Deep Belief Network* does not produce any prominent results. All of its values are mediocre if they are compared to the other techniques. The *quantization Z-images approaches* have the highest combined error rates (CER) and have only one bright spot. Namely a capability to separate the intra and inter distributions that is among the best values for the *difference quantization approach* using the *Jensen-Shannon-Divergence*. Among the two classifiers, the *Jensen-Shannon-Divergence* (see 6.2.2) outperforms the *Cosine Distance* (see 6.2.1) in the majority of cases. As a result, it is advisable to use the *Resnet50* with the *Jensen-Shannon-Divergence* classifier for scenarios in which computation times do not matter. Of course, the amount of calculations that are needed for the *Resnet50* is relatively big. In case the runtime has to be optimized, the best options are switching to the still quite well performing but much faster *contouring Z-images approaches* combined with the *Jensen-Shannon-Divergence* classifier or to the *local binary patterns* in combination with the *Jensen-Shannon-Divergence* classifier, which is also performing well but has even lower computational needs.

		min. CER	FAR	FRR	Average intra inter distance
Local Binary Pattern	cosine	0.227	0.114	0.340	0.042
	JSD	0.235	0.150	0.321	0.068
Deep Belief Network	cosine	0.289	0.195	0.383	0.063
	JSD	0.287	0.213	0.362	0.077
Z-Images diff. Contouring	cosine	0.272	0.109	0.436	0.243
	JSD	0.246	0.121	0.372	0.154
Z-Images diff. Quantization	cosine	0.294	0.156	0.431	0.201
	JSD	0.298	0.148	0.448	0.231
Z-Images inter Contouring	cosine	0.282	0.084	0.481	0.257
	JSD	0.249	0.096	0.402	0.212
Z-Images inter Quantization	cosine	0.370	0.211	0.529	0.105
	JSD	0.372	0.276	0.468	0.115
ResNet50	cosine	0.214	0.104	0.325	0.161
	JSD	0.215	0.092	0.337	0.230

Table 6.2: Performance values for the periocular recognition using a randomly chosen subset of the self-recorded database (see chapter 4) with 150 images per subject. For the computations, only the respective techniques were altered, all other parameters were left untouched. It is important to note that the results are only comparable within this table. The chosen key figures are described in chapter 8. Unsurprisingly, the *ResNet50* (see 6.1.3.2) shows the best performance, but the gap to the *Local Binary Patterns* (see 6.2) and the *contouring Z-images* (see 6.1.2) is much smaller than expected. Among the two classifiers, the *Jensen-Shannon-Divergence* (see 6.2.2) clearly outperforms the *Cosine Distance* (see 6.2.1) in all cases.

Chapter 7

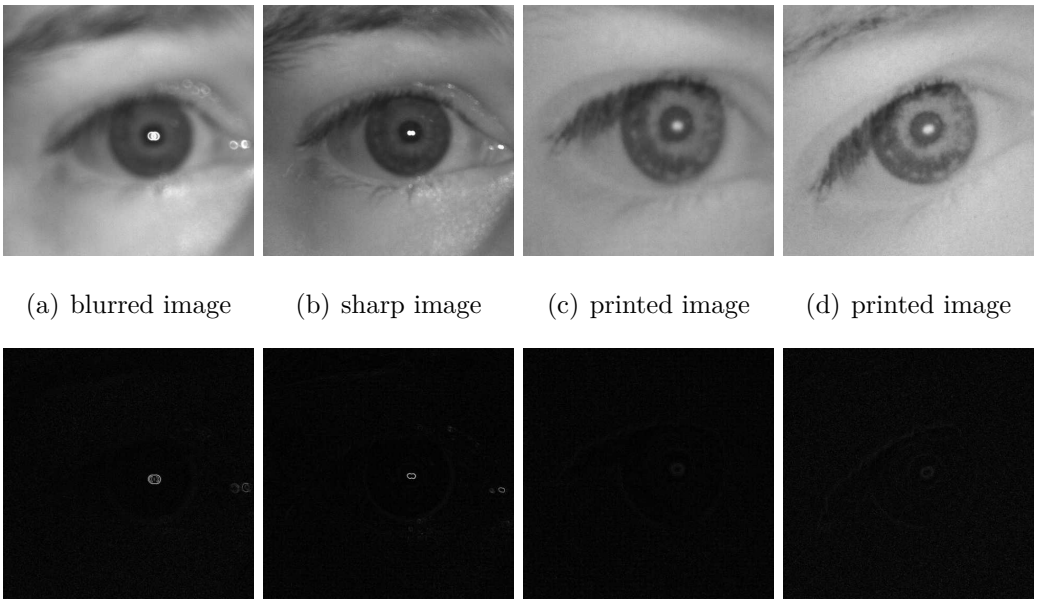
Liveness Detection and Anti-Spoofing

In the field of security there has always been a race between those who design more secure systems and those who want to bypass them in order to get to whatever has been secured. Of course, this is the case for biometrics as well. As a consequence, all biometric systems need to employ some countermeasures to prevent spoofing. Czajka gives an overview of the possibilities for such countermeasures [9]:

- Passive measurement of a static object:
  This approach is passively working with the data that is already there in order to detect static fake patterns in printouts.

- Active measurement of a static object:
  This means to perform an action in order to induce a static answer that allows to distinguish between real and fake.

- Passive measurement of dynamic objects:
  It is possible to passively search for a dynamic object that shall or shall



(a) blurred image (b) sharp image (c) printed image (d) printed image

(e) Edges from 7.1(a) (f) Edges from 7.1(b) (g) Edges from 7.1(c) (h) Edges from 7.1(d)

Figure 7.1: Images from the liveness detection database and their edge maps. The non-fake images show sharp reflections. The printouts were held in front of the same camera system, therefore they were exposed to NIR light again. The result is that they appear blurred and milky, with less sharp original reflections. Moreover, there are no additional isolated reflections from the repeated exposure to NIR light. This results in a feature that is able to separate the data in fake and non-fake. The edge maps were extracted using a combination of *Median Filtering* (see 2.5) and a *Canny Filter* (see 2.6.1). Clearly, the non-fake images have a much stronger filter answer than the printouts, on which hardly any can be spotted.

not be there in the case of an attack.

- Active measurement of dynamic objects:
  Performing an action to induce some dynamics that allows to detect spoofing.

The scenario that will be tackled here is a printout attack. Therefore, some images of the database (see chapter 4) were printed out and held in front of the camera, building up a new database containing almost $2,700$ fake and $15,000$ non-fake images. Using this fake image database, several approaches were evaluated, including methods based on *Gabor Filtering*, *Laplace Filtering*, *FFT*, *Local Binary Patterns* and even a *dynamic heartrate check* was tried without satisfying success. Like it is visible in the images in Figures 7.1(a) and 7.1(b), the non-fake images show sharp reflections. Due to the fact that the printouts were held in front of the same camera system, they were exposed to the NIR illumination again. This causes them to appear blurred and milky, which causes the initial reflections to appear less sharp. Moreover, there are no additional isolated reflections from the repeated exposure to NIR light (Figures 7.1(c) and 7.1(d)). Therefore, the sharpness of the reflections is a feature that is able to separate the data in fake and non-fake. Finally, the solution this thesis suggests is a *Canny* (see 2.6.1) based approach. It belongs to the countermeasure type of an *active measurement of a static object*. In order to create the edge maps, which are depicted in Figures 7.1(e) - 7.1(h), the images were first preprocessed using a *Median Filter* (see 2.5). This makes sure to defuse all the random widespread reflections that may occur in the printouts and could possibly add up to a strong edge. Afterwards, the *Canny Filter* (see 2.6.1) is applied. Clearly, the non-fake images have a much stronger filter answer than the printouts, at which one can hardly spot any. For the performance measures the ISO/IEC 30107-3 standard contains the relevant

error rates as follows:

- Attack presentation classification error rate (APCER):
  Proportion of attack presentations incorrectly classified as bona fide presentations

- Bona fide presentation classification error rate (BPCER):
  Proportion of bona fide presentations incorrectly classified as attack presentations

Figure 7.2 shows a subgraph of the results' histograms for the suggested technique. As expected, the sum of the edges in the images deliver a feature that is able to almost fully separate the results into fake (blue graph) and non-fake (orange graph) images, where the orange graph is of course cut off. Much higher values for the amount of edges occur in the full graph as well, but the depicted part of the distribution has a better visibility of the separability. The biggest value for the printouts is 0.0027. Therefore, a threshold of 0.0074 results in a *APCER* of 0 and a *BPCER* of 0.026. The algorithm is able to detect all the fake images, at a very low *BPCER* with a threshold that has quite some distance to the highest value of the fake images. Moreover, most of the *bona fide presentation classification errors* are simply closed eyes that really do not have any sharp edges and of course do not contain a processable iris. Finally, it is possible to conclude that the suggested approach is an almost perfect countermeasure for the given scenario.

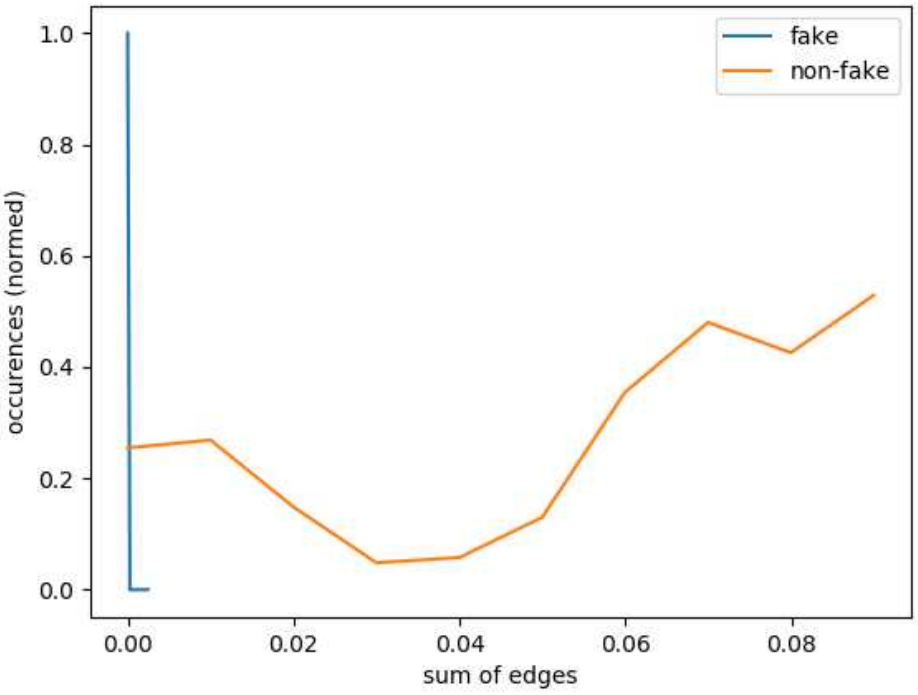


Figure 7.2: Normalized subgraph of the performance histogram for the liveness detection. The results are almost fully separated. The blue graph belongs to the fake images and the orange graph to the non-fake ones, where the orange graph is of course cut off. Much higher values for the amount of edges occur in the full graph as well, but the shown part of the distribution has a better visibility of the separability. The biggest value for the printouts is 0.0027. With a threshold of 0.0074 the *APCER* is 0.0 and the *BPCER* is 0.026. Most of the *bona fide presentation classification errors* are simply closed eyes that really do not have any sharp edges and of course do not contain a processable iris.

Chapter 8

Performance

This chapter is meant to give an overview over what is finally possible with the described approaches. Therefore, no longer only a randomly chosen subset of the self-recorded database (see chapter 4) is used but the full amount of available images. It is important to mention that the results are still not to be interpreted as absolute performance values but rather only give an indication, since the amount of comparisons is limited by the database's size. Throughout this thesis the following key figures were used for comparing the performances of the different techniques:

- The minimum combined error rate (CER) that is given by

$$CER = \frac{(FAR + FRR)}{2}\,, \tag{8.1}$$

  with FAR being the false acceptance rate and FRR being the false rejection rate. Similar to the equal error rate (EER), the CER is a measure of the quality of a method, which comes with better comparability between different methods in some cases. The smaller the CER is, the better the performance can become.

- The false acceptance rate FAR is given by the amount of falsely accepted

inter class comparisons divided by the total amount of inter class comparisons

- The false rejection rate (FRR) is given by the amount of falsely rejected intra class comparisons divided by the total amount of intra class comparisons

- The lowest false rejection rate (FRR) without false acceptances (FA): Since the operating threshold of iris recognition systems is set in a way that zero FA are ensured, this value is a good indicator for the system's actual performance. The lower the value is, the better the recognition rates are.

- The minimum segmentations per eye: This key figure is a value that allows to compare the time a subject needs to interact with the system. High values mean fastness and convenience. Whereas low values indicate that it might take a long time or even be impossible for some people to successfully enroll or identify themselves with the system.

- The average distance between the intra and inter distributions: Gives a measure of how good the method is able to separate the different subjects. Higher values mean higher distances and therefore a better separability.

- The signal to noise ratio: Gives a measure of how sparse the used noise removal method (see 5.3) is. The higher the signal to noise ratio, the more iris data is available for comparisons. It does not apply to the periocular recognition, because no noise removal is used there.

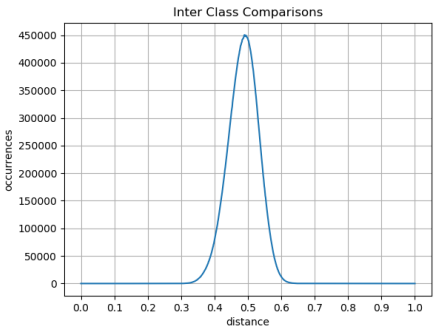
As a final result regarding the iris recognition, it is most advisable for the evaluated scenario and the available self-recorded database (see chapter 4) to use the *Sharpness Check* (see 5.1.1), in order to ensure that only sharp images

are processed, the *Segmentation in the Polar Representation* (see 5.2.3) for a robust and fast segmentation, *Adaptive Thresholding* (see 5.3.4) for optimum noise removal, the *Shape Count Check* (see 5.4.1) for security against fatal segmentations, the *2D-Gabor Filter* (see 5.6.2) for optimum feature extraction, and the *Score Normalization* (see 5.7.3) as well as the *Template Weighting* (see 5.7.4) for improving the matching process. It is important to emphasize that this recommendation is only valid for the considered case and that it is composed with regard to the trade-off between security and convenience, which developers always have to face in the domain of biometrics. For any changes, the best combination will have to be re-evaluated. Under these conditions, it is possible to set the recognition threshold in a way that no false acceptances occur, whereas the false rejection rate (FRR) is still at a decent value of 0.422 (see Table 8.1). This result was calculated by performing roughly $5.6 \cdot 10^7$ distinct comparisons. The remaining ones, up to the maximum of $1.1 \cdot 10^8$ comparisons, were not computed, due to images that were rejected due to failed segmentations, too much noise or fatal segmentations. The results for the periocular recognition do not really show applicability for the technique as a sole approach (see Table 8.1). Nevertheless, the main goal why periocular recognition was considered in this work, was to fuse iris recognition and periocular recognition (see chapter 6). Therefore, a simple weighted fusing technique was used. In order to calculate the final distance result, the periocular recognition result has a weight of 5 %, whereas the iris recognition is responsible for the remaining 95 %. As an outcome the recognition performance is further enhanced. The key figures in Table 8.1 show how the results improve for the used fusing technique. The minimum false rejection rate (FRR) without false acceptances (FA) is lowered by 24.6 %, down to a value of 0.318 and the average distance between the intra and inter distributions is increased by 15.7 %, up to a value of

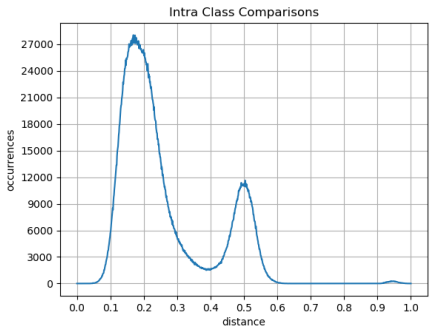
	minimum FRR without FA	Average intra inter distance
Iris Recognition	0.422	0.229
Periocular Recognition	0.967	0.049
Fused Recognition	0.318	0.265

Table 8.1: Performance values for the fusion of periocular and iris recognition, using the whole self-recorded database (see chapter 4). For the evaluations only the respective techniques were altered, all other parameters were left untouched. It is important to note that the results are only comparable within this table. The final result for the iris recognition shows a decent value of 0.422 for the false rejection rate (FRR), if the recognition threshold is set in a way that the false acceptance rate (FAR) is zero. The results for the periocular recognition do not really show applicability for the technique as a sole approach, but periocular recognition was primarily considered in this work, in order to be fused with iris recognition. Therefore, a simple weighted fusing technique was used, which weights the periocular recognition with 5 % and the iris recognition with 95 %. Thereby, the key figures are significantly improved. The minimum false rejection rate (FRR) without false acceptances (FA) improved to a value of 0.318, which corresponds to a decrease by 24.6 %. On the other hand, the average distance between the intra and inter distributions is raised up to 0.265, which is equivalent to an increase by 15.7 %. For the corresponding graphs showing the distributions of the intra class and inter class distances, see Figure 8.1.

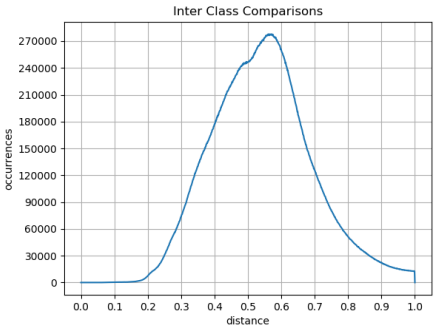
0.265. Figure 8.1 shows typical distributions of the distances for the inter class as well as the inter class comparison cases. What directly catches the eye, is the small peak in the iris recognition intra class comparison distribution at the same distance, where the inter class comparison distribution has its peak. This originates from the fact that it is almost random which eye will be computed in case there are two eyes in an image. This depends on the eye detection algorithm (see 5.2), which does not always deliver the same result because the used *OpenCV* [29] implementation uses a probabilistic approach. As a result, comparisons between the left and the right eye of subjects will happen and these are not any different from inter class comparisons as the patterns of the two eyes are not correlated with each other. Therefore, the expectation value for the distance is 0.5, which suits to the peak's location. Similar to its key figures (see Table 8.1), the middle graph does not look very promising. Especially, the intra class distance distribution does not even show distinct peaks, which suggests that there is either a big variance in how good the algorithm works for different subjects or that the approach is very sensitive to changes in the output of the eye detection algorithm (see 5.2), regarding the size and exact location of the clipped eye image. Nevertheless, the mixture of iris recognition and periocular recognition results in an improvement of the performance, as mentioned above. This can be also seen in the distribution graphs. Boosting the outcome that is shown in the graphs on the top (iris recognition) with the outcome that is depicted in the middle graphs (periocular recognition) results in the bottom graphs. Thereby, the performance is quite well improved. In the graphs, the increase in the distance between the two distributions is clearly visible. Compared to the iris recognition, the inter class comparison distribution is moved towards higher distances and the intra class distribution is shifted towards the lower ones. In comparison to [35], the recognition performance



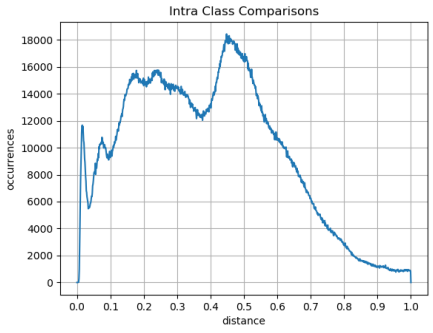
(a) iris inter class distribution



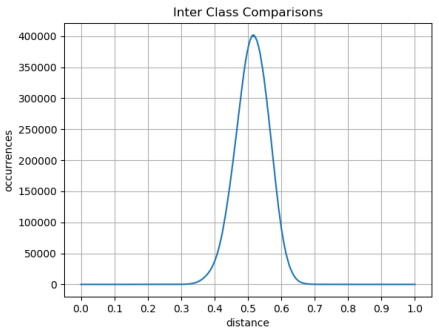
(b) iris intra class distribution



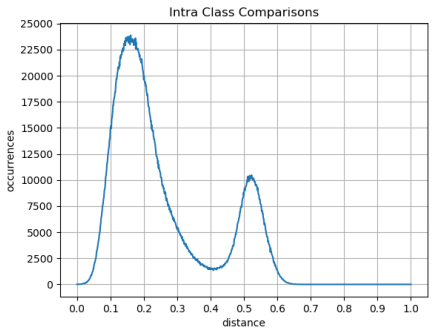
(c) periocular inter class distribution



(d) periocular intra class distribution



(e) fused inter class distribution



(f) fused intra class distribution

Figure 8.1: Graphs showing the distribution of distances for the inter and intra class cases for a full cross comparison using the full self-recorded database (see chapter 4). For the corresponding key figures, see Table 8.1.

has improved. Of course, direct and absolute comparisons between the given results are almost meaningless, because the databases in use are not the same as in [35] and only performance values for a single static threshold were evaluated. Nevertheless, it is possible to get an idea of how big the improvement is, by comparing a standard value for the key figure of minimum false rejections rates (FRR) without false acceptances (FA) for a full database iris recognition cross comparison of this work (0.422, see Table 8.1) with the lowest achieved weighted average value of the false rejection rate (FRR) of [35]: 0.552. Thereby, the false acceptance rate (FAR) is not anywhere close to being 0 but is 0.052. Hence, even though the FRR of 0.552 is significantly worse compared to the 0.422 which were achieved in this work, more than 5 % false acceptances (FA) occur as well. Other given results in [35] state a FAR of zero, too, but only in combination with a FRR of 0.995. This is actually a value that results in a completely useless outcome. Furthermore, the supposedly best values of [35] only reach roughly the range of the performance that could be achieved for the periocular recognition (see Table 6.2) in this work. Taking into account that also the database (see chapter 4) is way more challenging than before [35], all the mentioned results allow to conclude that the recognition performance really has improved by magnitudes. Thereby, portions of the enhancement might as well originate from the change in the normalization resolution from 16×64, respectively 8×128, to the new and higher resolution of 16×128 (see 5.5), which is of course as well a finding of this thesis.

Chapter 9

Conclusion and Prospects

All the presented techniques and the achieved performances half way through this thesis had convinced the responsible marketing department to present the developed iris recognition algorithm as part of a vehicle smart access scenario at the Consumer Electronics Show (CES) in 2018, which took place in Las Vegas. With this success, the initial goal to implement the system in a car demonstrator had been achieved. Now, the decision is at the car manufacturers to order the system as a replacement of the car key. Yet, it is not clear which biometrics will be used for authentication in the automotive world. But it is only a matter of time that any of the available choices will be implemented in series production. The final results of this thesis show that iris recognition is fully capable of providing the needed recognition rates, while the users can comfortably be identified from a distance of up to one meter, without even looking directly into the camera if the presented gaze removal technique (see 5.1.3) is used. In summary, the performance values are quite impressive if it is taken into account that the used self-recorded database (see chapter 4) has an ethnological variance close to reality and contains lots of possible obstacles, like people who have potentially problematic eyes with lesions or even glass

eye replacements. Although there are no values for comparison available, it has of course to be assumed that the results most probably cannot compete with the high constraint and human controlled commercial systems that are for example being used at the airport of the United Arab Emirates. In the automotive world comfort plays a much bigger role and the system cannot be human controlled for any car implementations, the results are well enough, especially if the system is fused with the periocular recognition.

Throughout this work, for each task that is needed for iris recognition, multiple techniques were implemented, thoroughly optimized and compared to each other. For the recording of images an automotive camera (see 3.2) in combination with an auto-focus functionality using a liquid lense (see 3.3) was used. In contrast to [35], both of these can now be controlled directly by the algorithm. This allows a fast adaptation to changing environmental conditions. For *preprocessing* (see 5.1), the *Sharpness Check* (see 5.1.1) is able to detect most of the unsharp pictures, by measuring the impact of sharpness and blurring filters on the image and by comparing the outcome image's variances to the original image's variance. It was pursued to sharpen the detected images with *Unsharp Masking* as well as *Laplacian Filtering*, with good optical results but with limited success for the recognition rates. Similarly, for achieving *Brightness Invariance* (see 5.1.2) the checked techniques were only able to give good optical results, whereas there was no performance improvement. A great success is the technique for *Eye Gaze Removal* (see 5.1.3) that uses a *Perspective Transform* to correct big eye gazes and make the images processable. The improvement in *segmentation* (see 5.2), relative to [35], is substantial, as the presented techniques feature a far more robust segmentation than before. Even the *Hough Transform* (see 5.2.1) was enhanced by using the new multi-stage approach. Especially the newly introduced *Segmentation*

in the Polar Representation (see 5.2.3), boosted with the *Snake Algorithm* (see 5.2.2), results in a good improvement. This technique searches for gradient maximums corresponding to the pupil and the iris along the radial axis of a polar represented eye originating from the center of the pupil and is afterwards boosted with an active contour algorithm that iteratively approximates and optimizes the contours. Finally, the *Unet Segmentation* (see 5.2.4), using a special *Convolutional Neural Network (CNN)* to learn the segmentation, is performing even better, although it comes with much bigger computational load. Similarly, considerable progress can be observed in the *noise removal* (see 5.3), where the superior technique, the *Adaptive Thresholding* (see 5.3.4), outperforms the older *Hough Line Transform* method (see 5.3.1), as well as the two other implemented techniques using a *Variance Analysis* (see 5.3.2), respectively an adaptive version of the *Canny Edge Detector* (see 5.3.3). The few remaining fatal segmentations can be coped with by using a *Segmentation Quality Check* (see 5.4), which offers the reliable *Shape Count Check* (see 5.4.1) and the stricter and more secure *Histogram Based Check* (see 5.4.2). The research on *feature extraction* (see 5.6) suggests to use the superior *2D-Gabor Filter* (see 5.6.2), as it performs significantly better than the *1D-Log Gabor Filter* (see 5.6.1). In the *matching process* (see 5.7), the recognition rates can be easily boosted by exploiting *Score Normalization* (see 5.7.3), which creates invariance to the amount of detected noise and *Template Weighting* (see 5.7.4), which gives bigger weights to the parts of the iris with a higher information density. Finally, it was revealed that iris recognition can be fused with periocular recognition. For this purpose, several methods for periocular *feature extraction* (see 6.1) and *classification* (see 6.2) were evaluated, resulting in a nice boosting of the overall recognition performance. Thereby, it is most advisable to chose the *Resnet50* (see 6.1.3.2), with the *Jensen-Shannon-Divergence* classifier (see

6.2.2), for scenarios in which computation times do not matter. For lower computational powers, the *contouring Z-images approaches* (see 6.1.2) or the *Local Binary Patterns* (see 6.2), both with the *Jensen-Shannon-Divergence*, can be used. In the end, the research on *Liveness Detection and Anti-Spoofing* (see chapter 7) led to a reliable detection method for printout attacks.

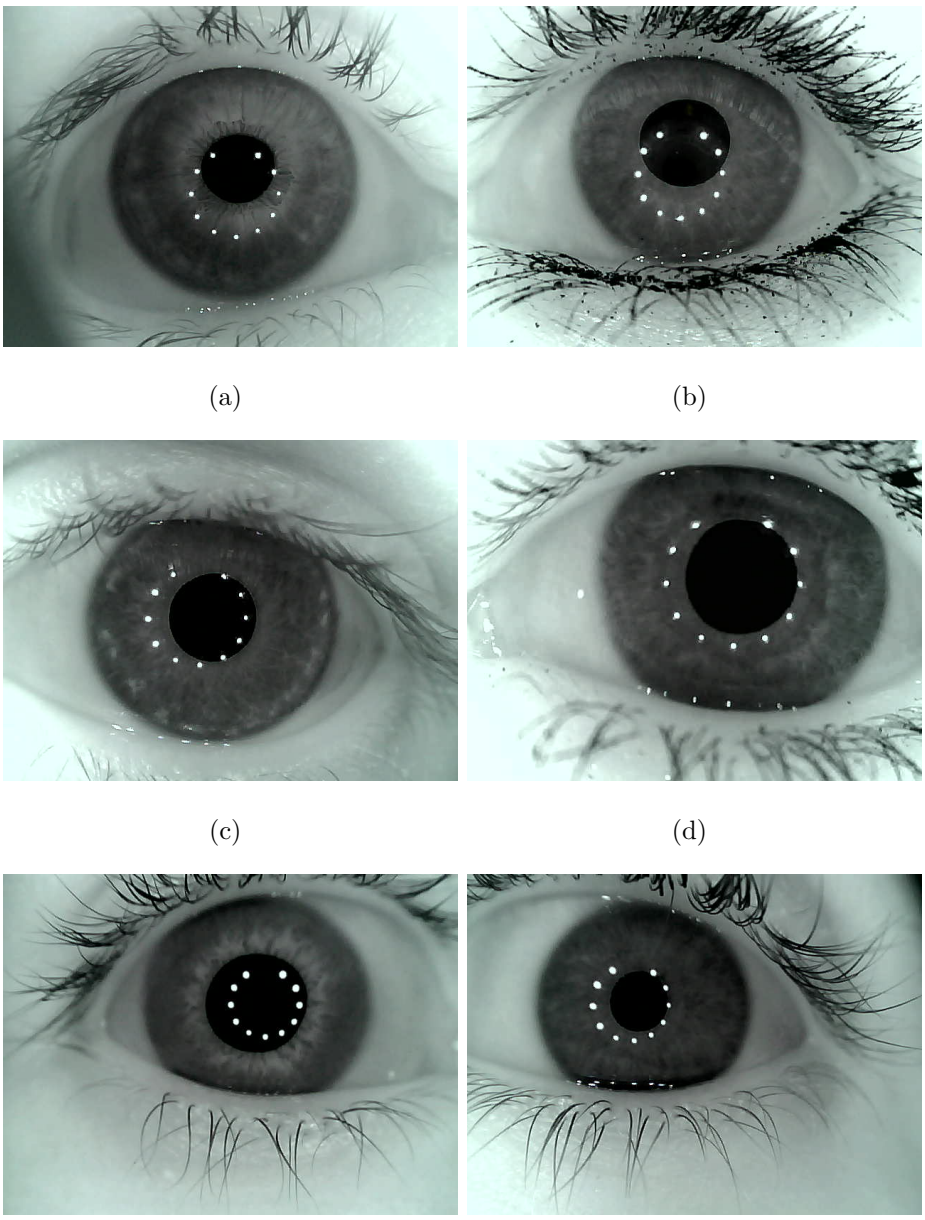
It turned out that the algorithm also works fine with most glasses as well as with contact lenses. Even most sunglasses do not block the NIR light emitted from the OSRAM NIR LEDs (see Figure 3.5) and therefore still allow an identification.

In order to deploy the algorithm on an actual modern car ECU, the main programming language needs to be changed from *Python* to *C++*. As the implementation mainly uses *OpenCV* [29] for the image processing, it would be only relatively little effort, as *OpenCV* offers an interface to *C++* as well. The reason why this switch has not already been done is that prototyping is way more rapid with *Python*.

The most recent findings suggest that it might be helpful for the final applicability of the system to employ a smart way of enrolling new subjects, in a way that the variance between the retained templates is as small as possible. This would come with another mechanism for preventing falsely extracted data from penetrating the system. Moreover, a smart way of combining a number of results that are close to the defined recognition threshold could lead to an adequate identification as well, without rising the danger of false acceptances (FA). This could decrease the required amount of images and hence the time needed for identification. Finally, it is definitively required to think about several other countermeasures to spoofing scenarios, in order to further secure the system against those, before series production could start.

Appendix A

Images recorded with the FOSCAM Night Vision Camera



(a) (b)

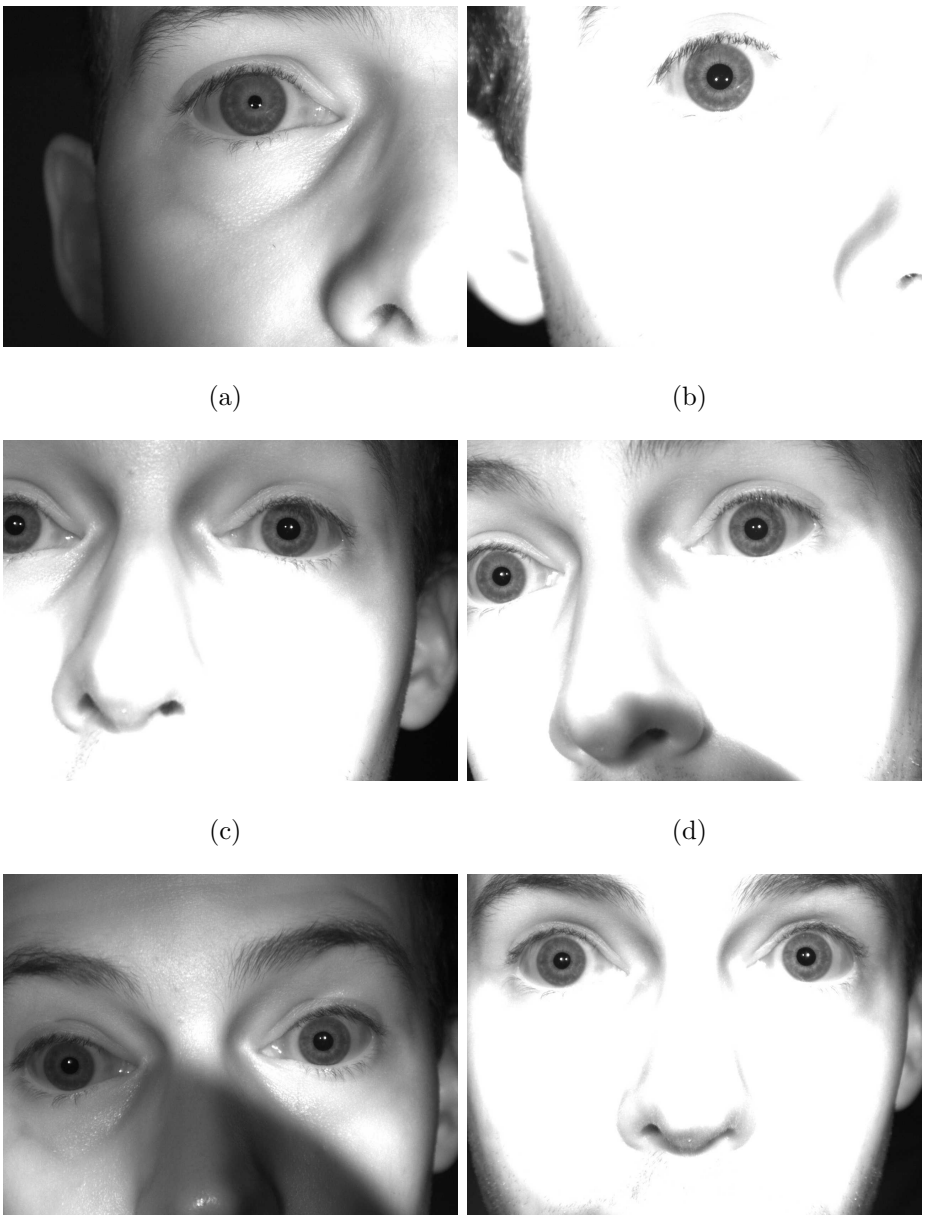
(c) (d)

(e) (f)

Figure A.1: Pictures of different eyes recorded with the *Foscam camera*, with IR illumination. The ring of white points prove that the IR LEDs were switched on. All images show a hint of green caused by the camera's status LED [35].

Appendix B

Images recorded with the Basler Automotive Camera

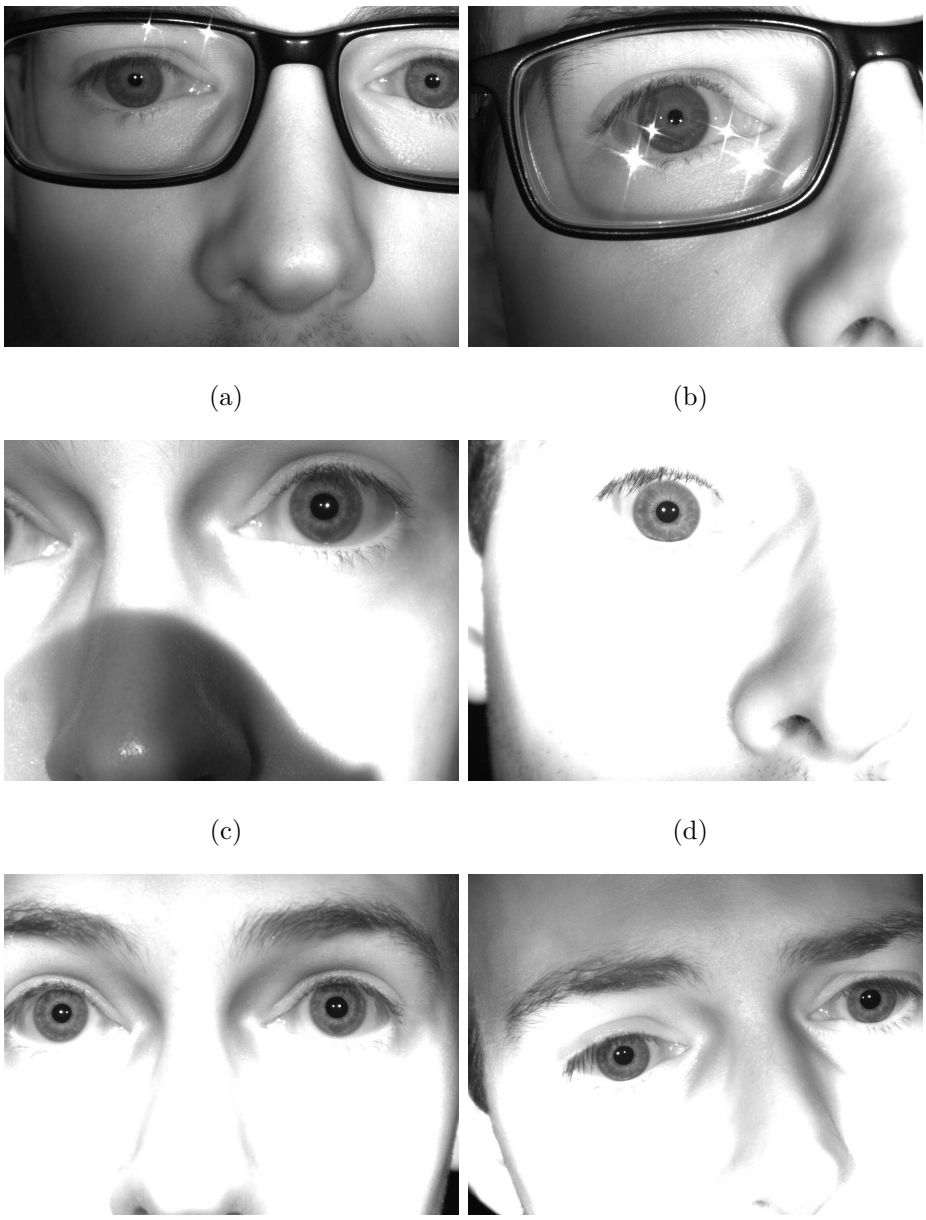


(a) (b)

(c) (d)

(e) (f)

Figure B.1: Pictures recorded with the *Basler camera*, with IR illumination, from different directions and distances and with different camera settings. The reflections prove that the *OSRAM IR LEDs* were switched on [35].



(a) (b)

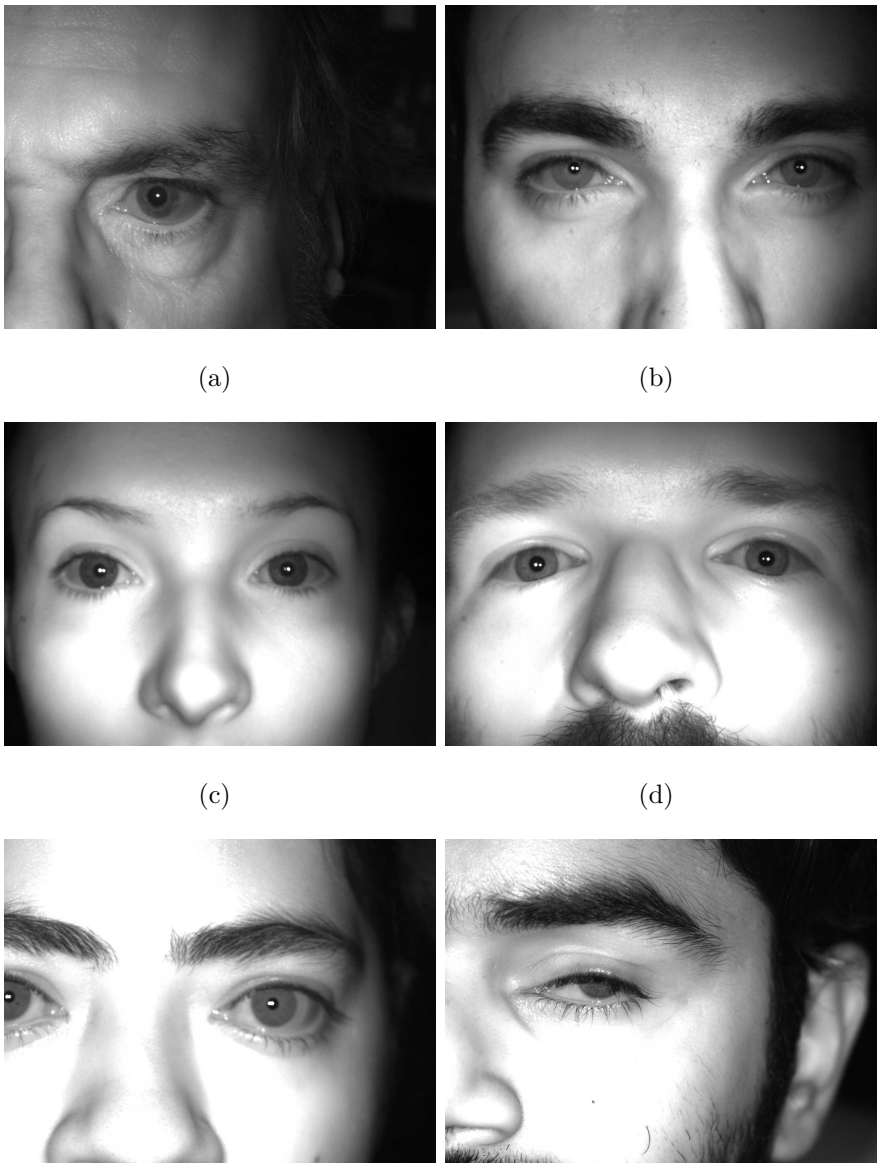
(c) (d)

(e) (f)

Figure B.2: Pictures recorded with the *Basler camera*, with IR illumination, from different directions and distances and with different camera settings. The reflections prove that the *OSRAM IR LEDs* were switched on [35].

Appendix C

Images from the Self-Recorded Database

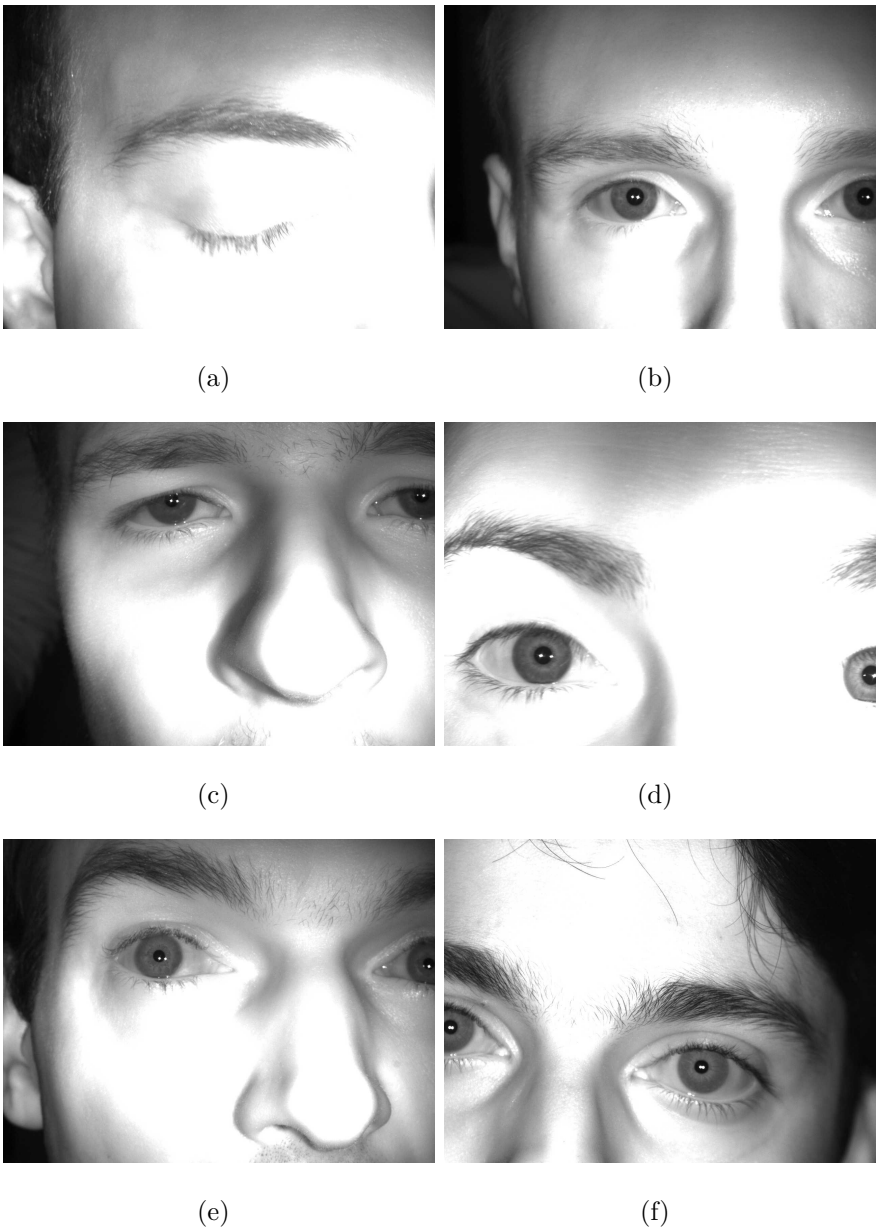


(a) (b)

(c) (d)

(e) (f)

Figure C.1: Pictures from the self-recorded database. The images in this database contain eyes with a big variance in sharpness, distance, gaze, illumination, gender, eye opening, eye color, as well as eyes that had a surgery, were injured or even replaced by a glass eye. This Figure shows some example images.

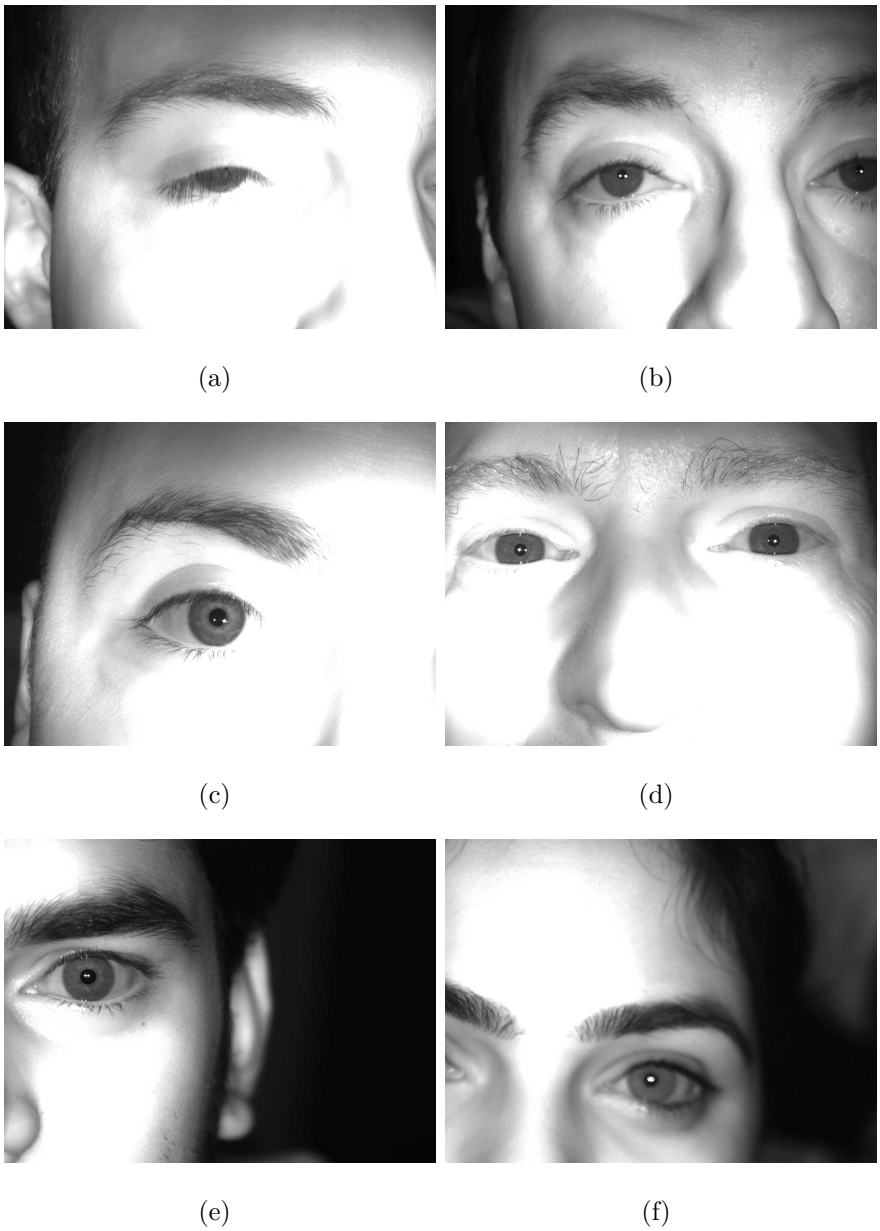


(a) (b)

(c) (d)

(e) (f)

Figure C.2: Pictures from the self-recorded database. The images in this database contain eyes with a big variance in sharpness, distance, gaze, illumination, gender, eye opening, eye color as well as eyes that had surgery, were injured or even replaced by a glass eye. This Figure shows some example images.

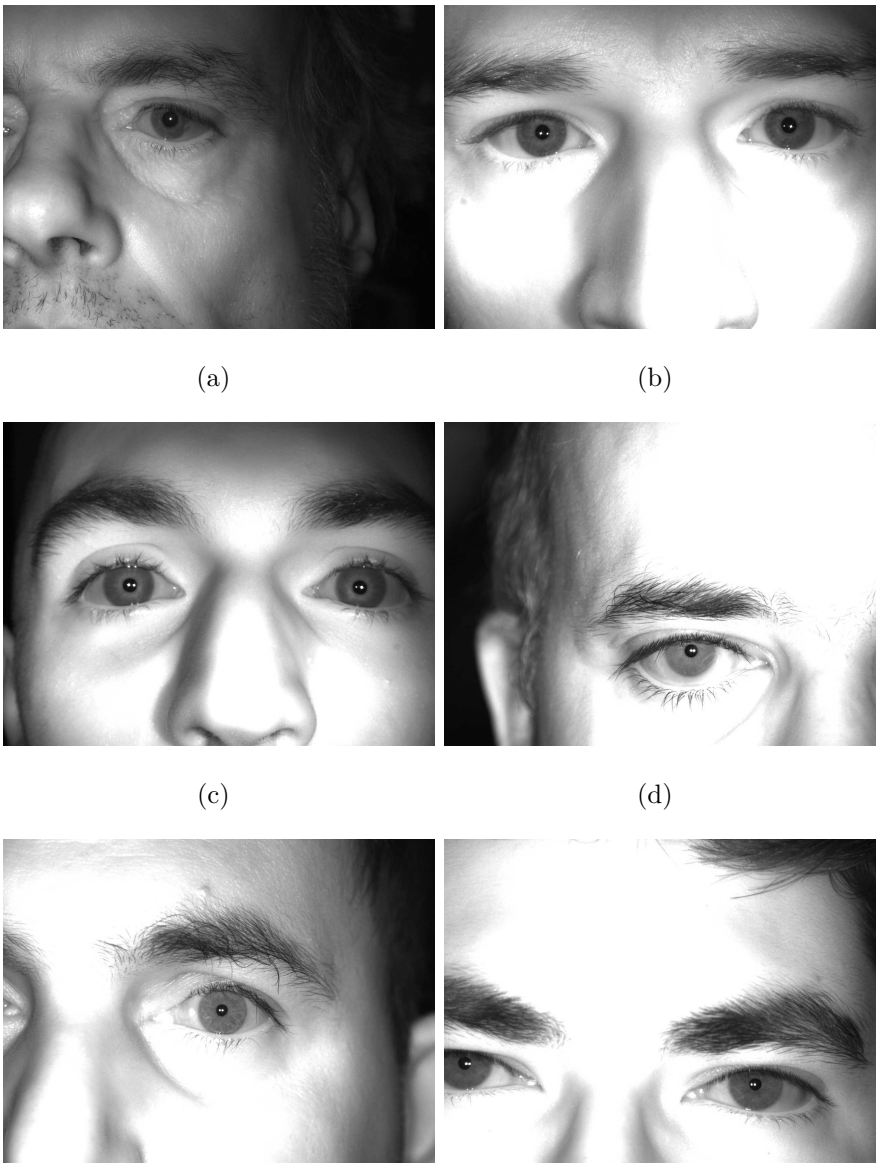


(a) (b)

(c) (d)

(e) (f)

Figure C.3: Pictures from the self-recorded database. The images in this database contain eyes with a big variance in sharpness, distance, gaze, illumination, gender, eye opening, eye color as well as eyes that had surgery, were injured or even replaced by a glass eye. This Figure shows some example images.

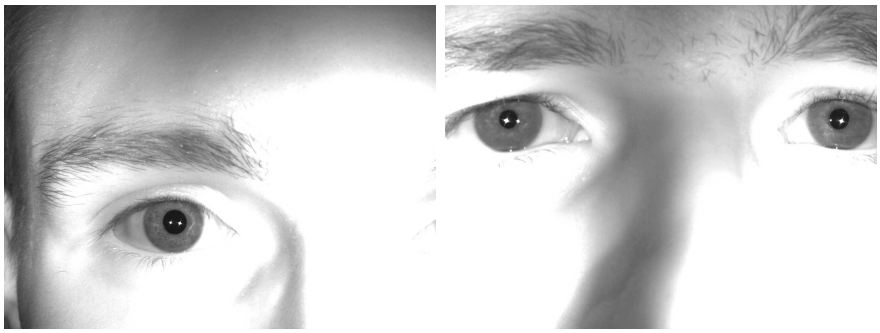


(a) (b)

(c) (d)

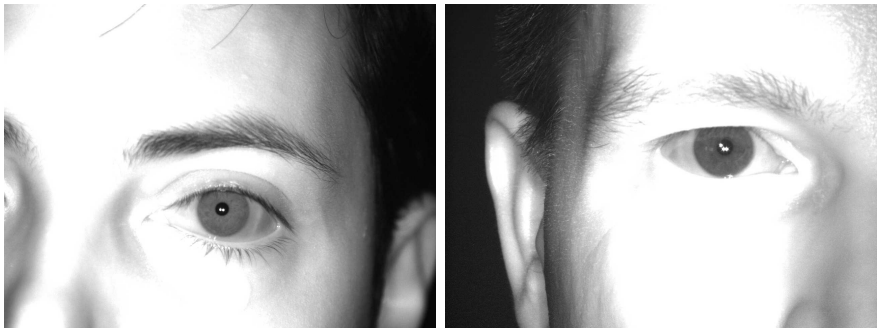
(e) (f)

Figure C.4: Pictures from the self-recorded database. The images in this database contain eyes with a big variance in sharpness, distance, gaze, illumination, gender, eye opening, eye color as well as eyes that had surgery, were injured or even replaced by a glass eye. This Figure shows some example images.



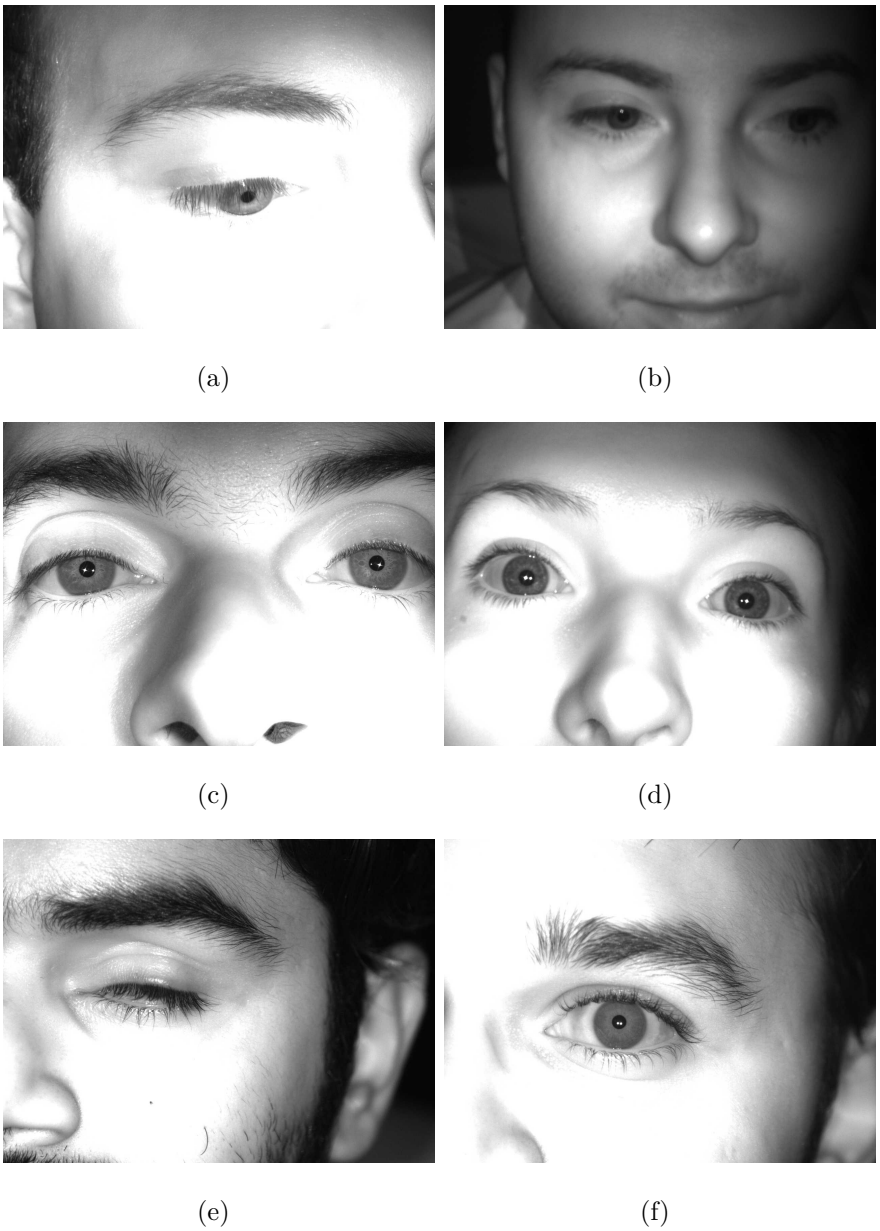
(a) (b)

(c) (d)



(e) (f)

Figure C.5: Pictures from the self-recorded database. The images in this database contain eyes with a big variance in sharpness, distance, gaze, illumination, gender, eye opening, eye color as well as eyes that had surgery, were injured or even replaced by a glass eye. This Figure shows some example images.



(a) (b)

(c) (d)

(e) (f)

Figure C.6: Pictures from the self-recorded database. The images in this database contain eyes with a big variance in sharpness, distance, gaze, illumination, gender, eye opening, eye color as well as eyes that had surgery, were injured or even replaced by a glass eye. This Figure shows some example images.

List of Figures

List of Tables

Bibliography

[1] Alonso-Fernandez, Fernando ; Bigun, Josef: A Survey on Periocular Biometrics Research. In: *Pattern Recogn. Lett.* 82 (2016), Oktober, Nr. P2, 92–105. http://dx.doi.org/10.1016/j.patrec.2015.08.026. – DOI 10.1016/j.patrec.2015.08.026. – ISSN 0167–8655

[2] Bastos, C. A. C. M. ; Ren, T. I. ; Cavalcanti, G. D. C.: Analysis of 2D log-Gabor Filters to Encode Iris Patterns. In: *Proc. 22nd IEEE Int. Conf. Tools with Artificial Intelligence*, 2010. – ISSN 1082–3409, S. 377–378

[3] Bharadwaj, S. ; Bhatt, H. S. ; Vatsa, M. ; Singh, R.: Periocular biometrics: When iris recognition fails. In: *2010 Fourth IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, 2010, S. 1–6

[4] Bishop, Christopher M.: *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg : Springer-Verlag, 2006. – ISBN 0387310738

[5] Boukerroui, Djamal ; Noble, J. A. ; Brady, Michael: On the Choice of Band-Pass Quadrature Filters. In: *J. Math. Imaging Vision* 21 (2004),

Juli, Nr. 1-2, 53. http://dx.doi.org/10.1023/B:JMIV.0000026557.50965.09. – DOI 10.1023/B:JMIV.0000026557.50965.09

[6] Brunelli, Roberto: *Template Matching Techniques in Computer Vision: Theory and Practice*. Wiley Publishing, 2009. – ISBN 0470517069, 9780470517062

[7] Canny, J.: A Computational Approach to Edge Detection. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-8 (1986), November, Nr. 6. http://dx.doi.org/10.1109/TPAMI.1986.4767851. – DOI 10.1109/TPAMI.1986.4767851. – ISSN 0162–8828

[8] Corning: *Corning® Varioptic® Lenses brochure*. v10. Invenios France SAS 24B rue Jean Baldassini 69007 Lyon, France, 2018

[9] Czajka, Adam: Iris Liveness Detection by Modeling Dynamic Pupil Features. In: *Handbook of Iris Recognition*. London : Springer London, 2016. – ISBN 978–1–4471–6784–6

[10] Dagan, Ido ; Lee, Lillian ; Pereira, Fernando: Similarity-based Methods for Word Sense Disambiguation. In: *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*. Stroudsburg, PA, USA : Association for Computational Linguistics, 1997 (ACL '98/EACL '98), 56–63

[11] Daugman, J.: How iris recognition works. In: *IEEE Transactions on Circuits and Systems for Video Technology* 14 (2004), Jan. http://dx.doi.org/10.1109/TCSVT.2003.818350. – DOI 10.1109/TCSVT.2003.818350. – ISSN 1051–8215

[12] Daugman, J.: New Methods in Iris Recognition. In: *Part B (Cybernetics) IEEE Transactions on Systems, Man, and Cybernetics* 37 (2007), Oct. http://dx.doi.org/10.1109/TSMCB.2007.903540. – DOI 10.1109/TSMCB.2007.903540. – ISSN 1083–4419

[13] Daugman, J.: Information Theory and the IrisCode. In: *IEEE Transactions on Information Forensics and Security* 11 (2016). http://dx.doi.org/10.1109/TIFS.2015.2500196. – DOI 10.1109/TIFS.2015.2500196. – ISSN 1556–6013

[14] Daugman, J ; Downing, C: Epigenetic randomness, complexity and singularity of human iris patterns. In: *Proc Biol Sci* 268 (2001), Aug. http://dx.doi.org/10.1098/rspb.2001.1696. – DOI 10.1098/rspb.2001.1696

[15] Daugman, J. G.: Complete discrete 2-D Gabor transforms by neural networks for image analysis and compression. In: *and Signal Processing IEEE Transactions on Acoustics, Speech* 36 (1988), Jul. http://dx.doi.org/10.1109/29.1644. – DOI 10.1109/29.1644. – ISSN 0096–3518

[16] Daugman, J. G.: High confidence visual recognition of persons by a test of statistical independence. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15 (1993), Nov. http://dx.doi.org/10.1109/34.244676. – DOI 10.1109/34.244676. – ISSN 0162–8828

[17] Daugman, JG: Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. In: *J Opt Soc Am A* 2 (1985), Jul

[18] Daugman, JG ; Downing, CJ: Demodulation, predictive coding, and spatial vision. In: *J Opt Soc Am A Opt Image Sci Vis* 12 (1995), Apr

[19] Daugman, John: The importance of being random: statistical principles of iris recognition. In: *Pattern Recognition* 36 (2003), Feb. http://dx.doi.org/10.1016/s0031-3203(02)00030-4. – DOI 10.1016/s0031–3203(02)00030–4

[20] Duda, Richard O. ; Hart, Peter E.: Use of the Hough Transformation to Detect Lines and Curves in Pictures. In: *Commun. ACM* 15 (1972), Jan. http://dx.doi.org/10.1145/361237.361242. – DOI 10.1145/361237.361242. – ISSN 0001–0782

[21] Endres, D. M. ; Schindelin, J. E.: A new metric for probability distributions. In: *IEEE Transactions on Information Theory* 49 (2003), Jul. http://dx.doi.org/10.1109/TIT.2003.813506. – DOI 10.1109/TIT.2003.813506. – ISSN 0018–9448

[22] Field, DJ: Relations between the statistics of natural images and the response properties of cortical cells. In: *J Opt Soc Am A* 4 (1987), Dec

[23] Gabor, D.: Theory of communication. In: *Journal of the Institution of Electrical Engineers-Part I: General* 94 (1947), Jan. http://dx.doi.org/10.1049/ji-1.1947.0015. – DOI 10.1049/ji–1.1947.0015

[24] Hamming, R. W.: Error Detecting and Error Correcting Codes. In: *Bell System Technical Journal* 29 (1950). http://dx.doi.org/10.1002/j.1538-7305.1950.tb00463.x. – DOI 10.1002/j.1538–7305.1950.tb00463.x. – ISSN 1538–7305

[25] He, Kaiming ; Zhang, Xiangyu ; Ren, Shaoqing ; Sun, Jian: Deep Residual Learning for Image Recognition. In: *CoRR* abs/1512.03385 (2015). http://arxiv.org/abs/1512.03385

[26] Hochreiter, Sepp ; Bengio, Yoshua ; Frasconi, Paolo: Gradient Flow in Recurrent Nets: the Difficulty of Learning Long-Term Dependencies. (2001)

[27] Hough, P. V. C.: Machine Analysis Of Bubble Chamber Pictures. In: *Proceedings, 2nd International Conference on High-Energy Accelerators and Instrumentation, HEACC 1959: CERN, Geneva, Switzerland, September 14-19, 1959*, 1959, 554-558

[28] Hubel, D. H. ; Wiesel, T. N.: Receptive fields and functional architecture of monkey striate cortex. In: *The Journal of Physiology* 195 (1968), 215-243. `http://dx.doi.org/10.1113/jphysiol.1968.sp008455`. – DOI 10.1113/jphysiol.1968.sp008455

[29] Itseez: *Open Source Computer Vision Library*. OpenCV 2.4.13 2016. `http://opencv.org/`. Version: May 2016

[30] Kahlil, A. T. ; Abou-Chadi, F. E. M.: Generation of iris codes using 1D Log-Gabor filter. In: *Proc. Int Computer Engineering and Systems (ICCES) Conf*, 2010

[31] Klontz, Joshua C. ; Burge, Mark: Periocular Recognition from Low-Quality Iris Images. In: *Handbook of Iris Recognition*, 2013

[32] Kong, W. K. ; Zhang, D.: Accurate iris segmentation based on novel reflection and eyelash detection model. In: *Proc. Int Intelligent Multimedia, Video and Speech Processing Symp*, 2001

[33] Kreyszig, E.: *Advanced Engineering Mathematics*. Wiley, 1979. – ISBN 9780471021407

[34] Kürner, Paul: *Machine Learning Techniques for Periocular Recognition*, Universität Regensburg, Bachelor Thesis, October 2018

[35] Langgartner, Florian: *Machine Learning Techniques for Iris Recognition from Distant Viewpoints*, Universität Regensburg, Masterarbeit, September 2016

[36] Long, Jonathan ; Shelhamer, Evan ; Darrell, Trevor: Fully Convolutional Networks for Semantic Segmentation. In: *CoRR* abs/1411.4038 (2014). http://arxiv.org/abs/1411.4038

[37] Ma, Li ; Wang, Yunhong ; Tan, Tieniu: Iris recognition using circular symmetric filters. In: *Proc. 16th Int Pattern Recognition Conf*, 2002. – ISSN 1051–4651

[38] Mäenpää, Topi: *The local binary pattern approach to texture analysis - extensions and applications*, Faculty of Technology, University of Oulu, Diss., 2003

[39] Masek, Libor: *Recognition of Human Iris Patterns for Biometric Identification*, School of Computer Science and Software Engineering, The University of Western Australia, Dissertation, 2003. http://www.peterkovesi.com/studentprojects/libor/

[40] Matas, J. ; Galambos, C. ; Kittler, J.: Robust Detection of Lines Using the Progressive Probabilistic Hough Transform. In: *Comput. Vis. Image Underst.* 78 (2000), April. http://dx.doi.org/10.1006/cviu.1999.0831. – DOI 10.1006/cviu.1999.0831. – ISSN 1077–3142

[41] Mehrotra, Hunny ; Vatsa, Mayank ; Singh, Richa ; Majhi, Banshidhar: Does Iris Change Over Time? In: *PLoS ONE* 8 (2013),

Nov. http://dx.doi.org/10.1371/journal.pone.0078333. – DOI 10.1371/journal.pone.0078333

[42] Monteiro, Joao C.: *Robust Iris Recognition under Unconstrained Settings*, Faculdade de Engenharia da Universidade do Porto, Master Thesis, August 2012

[43] Mugele, Frieder ; Baret, Jean-Christophe: Electrowetting: from basics to applications. In: *Journal of Physics: Condensed Matter* 17 (2005), Jul. http://dx.doi.org/10.1088/0953-8984/17/28/r01. – DOI 10.1088/0953–8984/17/28/r01

[44] Nair, Vinod ; Hinton, Geoffrey E.: Rectified Linear Units Improve Restricted Boltzmann Machines. In: *Proceedings of the 27th International Conference on International Conference on Machine Learning*. USA : Omnipress, 2010 (ICML'10). – ISBN 978–1–60558–907–7, 807–814

[45] Nigam, Ishan ; Vatsa, Mayank ; Singh, Richa: Ocular Biometrics. In: *Inf. Fusion* 26 (2015), Nov. http://dx.doi.org/10.1016/j.inffus.2015.03.005. – DOI 10.1016/j.inffus.2015.03.005. – ISSN 1566–2535

[46] Ojala, Timo ; Pietikäinen, Matti ; Harwood, David: A comparative study of texture measures with classification based on featured distributions. In: *Pattern Recognition* 29 (1996), 51 - 59. http://www.sciencedirect.com/science/article/pii/0031320395000674. – ISSN 0031–3203

[47] Oppenheim, A. V. ; Lim, J. S.: The importance of phase in signals. In: *Proceedings of the IEEE* 69 (1981), May. http://dx.doi.org/10.1109/PROC.1981.12022. – DOI 10.1109/PROC.1981.12022. – ISSN 0018–9219

[48] Park, U. ; Jillela, R. R. ; Ross, A. ; Jain, A. K.: Periocular Biometrics in the Visible Spectrum. In: *IEEE Transactions on Information Forensics and Security* 6 (2011), Mar. http://dx.doi.org/10.1109/TIFS.2010.2096810. – DOI 10.1109/TIFS.2010.2096810. – ISSN 1556–6013

[49] Petkovic, Tomislav ; Loncaric, Sven: An Extension to Hough Transform Based on Gradient Orientation. In: *CoRR* abs/1510.04863 (2015). http://arxiv.org/abs/1510.04863

[50] Pizer, Stephen M. ; Amburn, E. P. ; Austin, John D. ; Cromartie, Robert ; Geselowitz, Ari ; Greer, Trey ; Romeny, Bart Ter H. ; Zimmerman, John B.: Adaptive Histogram Equalization and Its Variations. In: *Comput. Vision Graph. Image Process.* 39 (1987), September. http://dx.doi.org/10.1016/S0734-189X(87)80186-X. – DOI 10.1016/S0734–189X(87)80186–X. – ISSN 0734–189X

[51] Rafael C. Gonzalez, Richard E. W.: *Digital Image Processing*. 3rd. PRENTICE HALL, 2007. – ISBN 013168728X

[52] Ramsauer, Dominik: *Implementation of Periocular Biometrics*, Universität Regensburg, Bachelor Thesis, January 2019

[53] Ronneberger, Olaf ; Fischer, Philipp ; Brox, Thomas: U-Net: Convolutional Networks for Biomedical Image Segmentation. In: *CoRR* abs/1505.04597 (2015). http://arxiv.org/abs/1505.04597

[54] Russakovsky, Olga ; Deng, Jia ; Su, Hao ; Krause, Jonathan ; Satheesh, Sanjeev ; Ma, Sean ; Huang, Zhiheng ; Karpathy, Andrej ; Khosla, Aditya ; Bernstein, Michael ; Berg, Alexander C. ; Fei-Fei, Li: ImageNet Large Scale Visual Recognition Challenge. In: *International Journal of Computer Vision (IJCV)* 115 (2015). http:

//dx.doi.org/10.1007/s11263-015-0816-y. – DOI 10.1007/s11263–015–0816–y

[55] Santhi, V. ; Acharjya, D. P. ; Ezhilarasan, M.: *Emerging Technologies in Intelligent Applications for Image and Video Processing*. http://dx.doi.org/10.4018/978-1-4666-9685-3. Version: 2016

[56] Santos, G. ; Proença, H.: Periocular biometrics: An emerging technology for unconstrained scenarios. In: *2013 IEEE Symposium on Computational Intelligence in Biometrics and Identity Management (CIBIM)*, 2013. – ISSN 2325–4300

[57] Schiessl, Patrick: *Automated Evaluation of an Iris Segmentation Quality*, Universität Regensburg, Master Thesis, October 2018

[58] Senninger, Dominik: *Analysis of electron diffraction patterns from carbon nanotubes with image processing to determine structural parameters*. http://epub.uni-regensburg.de/22789/. Version: December 2011

[59] Silva, Caroline ; Bouwmans, Thierry ; Frelicot, C: An eXtended Center-Symmetric Local Binary Pattern for Background Modeling and Subtraction in Videos. In: *VISAPP*, 2015

[60] Smerdon, D.: Anatomy of the eye and orbit. In: *Current Anaesthesia & Critical Care* 11 (2000), Dec

[61] Sund, Torbjørn ; Møystad, Anne: Sliding window adaptive histogram equalization of intraoral radiographs: Effect on image quality. In: *Dento maxillo facial radiology* 35 (2006). http://dx.doi.org/10.1259/dmfr/21936923. – DOI 10.1259/dmfr/21936923

[62] Szeliski, Richard: *Computer Vision: Algorithms and Applications*. 1st. Berlin, Heidelberg : Springer-Verlag, 2010. – ISBN 1848829345, 9781848829343

[63] Tisse, Christel-Loc ; Martin, Lionel ; Torres, Lionel ; Robert, Michel: Person Identification Technique Using Human Iris Recognition. In: *Proc. of Vision Interface*, 2002

[64] Vigneron, Vincent ; Syed, Tahir ; T. Duarte, Leonardo ; Lang, Elmar ; Iqbal, Sadaf ; Tomé, Ana: Z-Images. In: *Pattern Recognition and Image Analysis: 8th Iberian Conference, IbPRIA 2017*, 2017. – ISBN 978–3–319–58837–7

[65] Viola, P. ; Jones, M.: Rapid object detection using a boosted cascade of simple features. In: *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition CVPR 2001*, 2001. – ISSN 1063–6919

[66] Wildes, R. P. ; Asmuth, J. C. ; Green, G. L. ; Hsu, S. C. ; Kolczynski, R. J. ; Matey, J. R. ; McBride, S. E.: A system for automated iris recognition. In: *Proc. Second IEEE Workshop Applications of Computer Vision*, 1994

[67] Woodard, D. L. ; Pundlik, S. J. ; Lyle, J. R. ; Miller, P. E.: Periocular region appearance cues for biometric identification. In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, 2010. – ISSN 2160–7508

[68] Yao, Peng ; Li, Jun ; Ye, Xueyi ; Zhuang, Zhenquan ; Li, Bin: Iris Recognition Algorithm Using Modified Log-Gabor Filters. In: *Proc. 18th Int. Conf. Pattern Recognition (ICPR'06)* Bd. 4, 2006, S. 461–464

[69] Yuen, H. K. ; Princen, J. ; Illingworth, J. ; Kittler, J.: Comparative Study of Hough Transform Methods for Circle Finding. In: *Image Vision Comput.* 8 (1990), Feb. http://dx.doi.org/10.1016/0262-8856(90)90059-E. – DOI 10.1016/0262–8856(90)90059–E

[70] Zeckendorf, E.: Représentation des nombres naturels par une somme de nombres de Fibonacci ou de nombres de Lucas. In: *Bull. Soc. R. Sci. Liège* 41 (1972). – ISSN 0037–9565

Acknowledgments

I would like to thank all the people who supported me during my work on this thesis:

- Prof. Dr. Elmar Lang for giving me the opportunity to work on this interesting topic and his mentoring
- All members of the AG Lang at the University of Regensburg, especially
  - Patrick Schießl for his Master thesis about segmentation quality scores and for proof-reading my thesis
  - Konstantin Igl and Maximilian Melzner for proof-reading my thesis
  - Paul Kürner and Dominik Ramsauer for their Bachelor theses about periocular recognition
  - Dominik Senninger for providing his snake algorithm
- All Members of Artificial Intelligence and Robotics Lab at Continental Automotive GmbH, especially
  - Demetrio Aiello and Herbert Hofmann for giving me the opportunity to work on this interesting topic, their management and leadership
  - Oliver Horeth, Nicola Mularoni and Anuj Shah Arps for enriching discussions about the topic and a good colleagueship

  - Sanjay Bharadwaj for a good colleagueship and proof-reading my thesis

- Michael Moosbühler for proof-reading my thesis
- Dr. Stefan Solbrig for correcting this thesis
- My father Bernhard Langgartner for his support throughout my study and proof reading my thesis
- My mother Maria Grazia Langgartner for her support throughout my study
- Andrea Wanninger for her patience and support

Erklärung

In dieser Dissertation sind Geschäftsgeheimnisse der Continental Automotive GmbH enthalten. Eine Weitergabe oder Vervielfältigung oder Veröffentlichung der Arbeit sowie die Verwertung und Mitteilung ihres Inhalts ist ohne ausdrückliche schriftliche Genehmigung der Continental Automotive GmbH, Siemensstraße 12, 93055 Regensburg, nicht gestattet.

Ich habe die Arbeit selbständig verfasst, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt und bisher keiner anderen Prüfungsbehörde vorgelegt.

Regensburg, den Unterschrift