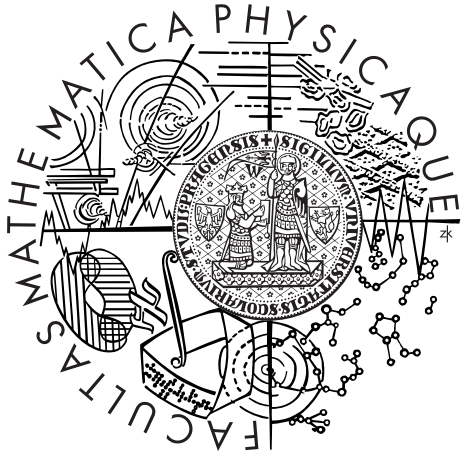
Charles University in Prague
Faculty of Mathematics and Physics

MASTER THESIS



Vojtěch Luhan

Error-Correcting Codes and Iris Recognition

Department of Algebra

Supervisor of the master thesis:	RNDr. Jan Šťovíček, Ph.D.
Study programme:	Mathematics
Specialization:	Mathematical Methods of Information Security

Prague 2014

I thank *Dr. Jan Šťovíček* for leading my thesis, spending time with me on consultations and helping me understand the mathematical background of the Iris Recognition. I thank him once again, together with *Dr. Aleš Drápal*, for their appraisals and constructive comments.

I would also like to thank *Pavel Horal*, who has been interested in this topic in the past and provided me with his results and insights within several amicable consultations in Karlínský Mlýn. I also thank *Dr. Andreas Uhl*, who gave me a few consultations which helped me either to find the best way to continue or to understand appropriate implementations.

Whenever there is a grammatically correct part, it is probably on account of *Carly Guglielmelli*, who read through all the thesis and found enormous number of mistakes.

Last, but definitely not least, my big thanks belong to *my family and friends* who found various ways to motivate me to finish the thesis in times when I needed it.

I declare that I carried out this master thesis independently, and only with the cited sources, literature and other professional sources.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular the fact that the Charles University in Prague has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 paragraph 1 of the Copyright Act.

In Prague on December 3^{rd}, 2013

Název práce: Samoopravné kódy a rozpoznávání podle duhovky

Autor: Vojtěch Luhan

Katedra: Katedra algebry

Vedoucí diplomové práce: RNDr. Jan Šťovíček, Ph.D., Katedra algebry

Abstrakt: Rozpoznávání podle duhovky v dnešní době představuje jednu z nejpřesnějších metod pro identifikaci a autentizaci. Tato práce si klade za cíl popsat matematickým aparátem algoritmy, které se při ní používají. Samotný popis daných algoritmů však není jediným cílem práce. Je v ní také navrženo několik možností, jak by jednotlivé části mohly být vylepšeny, popř. nahrazeny. Práce ale nepřehlíží ani další souvislosti spojené s rozpoznáváním podle duhovky, jako je např. jeho potenciální využití za pomoci samoopravných kódů v duhovkových kryptosystémech.

Druhá verze práce odstraňuje množství chyb a nepřesností objevených ve verzi první, zejména pak v kapitolách *ROI Definition*, *Hough Transform* a *Feature Extraction*. Kromě toho ale také přináší několik nových tvrzení, a hlavně pak v pseudokódech ukazuje, jak by se popisované algoritmy implementovaly v praxi.

Klíčová slova: Duhovka, Rozpoznávání podle duhovky, Gaborovy wavelety, Samoopravné kódy, Kryptosystémy

Title: Error-Correcting Codes and Iris Recognition

Author: Vojtěch Luhan

Department: Department of Algebra

Supervisor of the master thesis: RNDr. Jan Šťovíček, Ph.D., Department of Algebra

Abstract: Iris recognition constitutes one of the most powerful method for the identification and authentication of people today. This thesis aims to describe the algorithms used by a mathematical apparatus. The description of these algorithms is not the only objective of this thesis; the reason they were chosen and potential improvements or substitutions are also discussed. The background of iris recognition, its use in cryptosystems, and the application of error-correcting codes are investigated as well.

The second version of the thesis eliminates errata and a quantum of inaccuracies discovered in the first version, especially in the *ROI Definition*, the *Hough Transform* and the *Feature Extraction* sections. Besides that, it also contains several new propositions. Last, but not least, it shows a potential implementation of the algorithms described by appending pseudocodes to the relevant sections.

Keywords: Iris, Iris Recognition, Gabor Wavelets, Error-Correcting Codes, Cryptosystems

Table of Contents

1 Introduction **3**
- 1.1 Biometric Recognition . . . 3
  - 1.1.1 Physiological Biometrics . . . 4
  - 1.1.2 Behavioral Biometrics . . . 4
- 1.2 Iris Anatomy . . . 5
- 1.3 History of Iris Recognition . . . 6
- 1.4 Iris Recognition Deployment . . . 6
- 1.5 Performance Measurement . . . 7

I Iris Recognition **9**

2 Image Acquisition **10**
- 2.1 Eye Capture . . . 10
- 2.2 Image Storage . . . 11

3 Image Preprocessing **13**
- 3.1 Image Segmentation . . . 13
  - 3.1.1 ROI Definition . . . 13
  - 3.1.2 Boundary Localization . . . 17
    - 3.1.2.1 Integro-Differential Operator . . . 17
    - 3.1.2.2 Hough Transform . . . 21
- 3.2 Iris Normalization . . . 25
- 3.3 Noise Masks . . . 27

4 Feature Extraction **30**
- 4.1 Mathematical Background . . . 30
- 4.2 Deployment . . . 34

5 Iris Code Comparison **40**
- 5.1 Hamming Distance . . . 41
- 5.2 Levenshtein Distance . . . 44

II Iris Cryptosystems **49**

6 Error-Correcting Codes **51**
- 6.1 Hadamard Codes . . . 51
- 6.2 Reed-Solomon Codes . . . 53

7 Fuzzy Commitment Scheme **54**

8 Cancelable Iris Biometrics **57**
8.1 Block Re-Mapping 57
8.2 Mesh Warping 58

9 Potential Attacks **59**
9.1 Attacks on Physical Biometric Data 59
9.1.1 Spoofing 59
9.1.2 False Acceptance Attack 60
9.2 Attacks on Digital Biometric Data 60
9.2.1 Replay Attack 60
9.2.2 Masquerade Attack 61
9.2.3 Injection Attack 61
9.3 Substitution Attack 61
9.4 Tampering 61

References **62**

List of Figures **66**

List of Abbreviations **67**

Chapter 1

Introduction

In 1985 the world was fascinated by the eyes of a young girl photographed in Nasir Bagh. Nasir Bagh was a Pakistan camp for Afghan refugees orphaned during the Soviet invasion of Afganistan. Her bright green eyes captivated readers for years and the photo became the National Geographic Society's most recognized photograph in its 114-year history.[1] *Nobody knew her name though. She became known simply as* Afghan girl.[2]

17 years later in 2002 photographer Steve McCurry, author of the original image, returned back to the still standing Nasir Bagh to search for his Afghan girl. Some recognized their own friends in the photo, but after seeing their faces McCurry knew it wasn't them. Finally, in a remote village a week of walk away from the camp he found a woman called Sharbat Gula. She remembered that moment from Nasir Bagh. She had never been photographed before, nor after. McCurry knew it was her. *But nothing was certain until he provided his photos, both old and new, to John Daugman. With his new Gabor-wavelet-based iris recognition method he finally proved that with the probability of error* $1 : 6 \cdot 10^{16}$, *both the pairs of eyes belonged to the same person.*[3] *The Afghan girl was found.*[4]

Iris recognition systems constitute a powerful method for identification and authentication of people. They provide some of the most reliable results including all the benefits and drawbacks that biometrics yield. In my thesis I would like to name them, discuss their potentials and provide a mathematical background of the methods used. I would like to provide arguments for why these algorithms are used and what their alternatives are or how they can be improved.

1.1 Biometric Recognition

Biometrics as a word comes from Greek *bios* (life) and *metron* (measure) and it refers to a process by which a person's unique characteristics are detected and recorded by an electronic device or system for retrieval or confirmation of an identity.

Biometrics generally faces a few cardinal problems. The first one is an inaccuracy of measurement. While digital systems require accurate numbers for delivering correct responses, biometric characteristics registration can hardly be always the same.

Another disadvantage of biometrics is its consistancy. While stability is a

corner-stone of biometric recognition, it is simultaneously one of its major drawbacks. Once a pattern is in some way stolen, it can never be changed like a password.

The next problem is opposite to the previous one. While the patterns remain more-or-less constant, they are not constant absolutely. Their changes over years can cause incorrect system response after a long period of time. Some biometric characteristics are more stable (such as DNA), while some are influenced more (such as face geometry).

There are 2 basic types of biometrics: *physiological* and *behavioral*.

1.1.1 Physiological Biometrics

Physiological biometrics is a group of biometrics which is concerned by the biological and physiological features as captured by a biometric system.[5]

It specifically contains:

- *DNA*: examination of the unique strands found in DNA samples.
- *Fingerprint*: location and determination of the unique characteristics ('minutiae' – ridge endings and bifurcations) of the fingerprint.
- *Hand*: taking a 3-D image of a user's hand and its unique characteristics, such as thickness, length and width of fingers, distance between their joints and the bone structure.
- *Face*: measurement of distances between characteristic face features, such as ears, nose, eyes, mouth and cheeks.
- *Earlobe*: examination of the geometry of the earlobe.
- Last, but not least *Iris*: this technique will be described later in the thesis.

A comparison of the 4 most widely used biometrics from the point of view of effort, intrusiveness, accuracy and price is displayed in Table 1.1.

Accuracy	Iris	>	Finger	>	Face	>	Hand
Non-intrusiveness	Face	>	Iris	>	Hand	>	Finger
Effortlessness	Iris	>	Face	>	Finger	>	Hand
Cheapness	Finger	>	Face	>	Iris	>	Hand

Table 1.1: A comparison of the most widely used biometric methods from the view of effort, intrusiveness, accuracy and price.

1.1.2 Behavioral Biometrics

Behavioral biometrics is a group of biometrics which is concerned by the non-biological or non-physiological identifiers as captured by a biometric system.[5]

It consists of these 4 categories:

- *Signature*: analysis of the way and manner in which the user writes, characteristic in pressure and speed during the writing process.

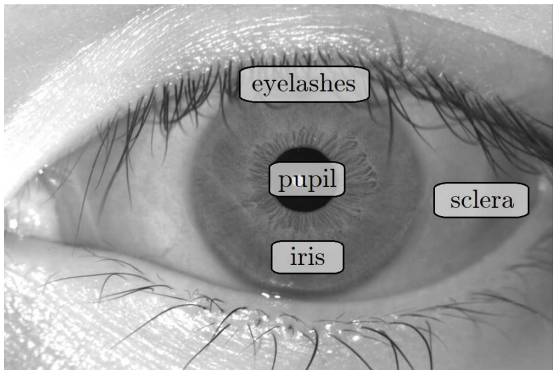


Figure 1.1: An eye from the frontal view

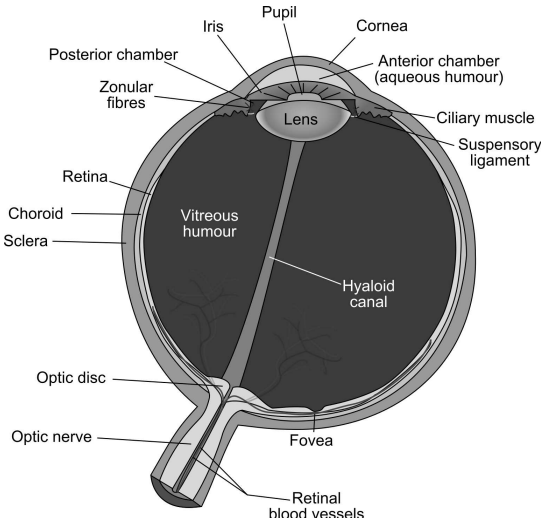


Figure 1.2: Human eye

- *Voice*: examination of patterns of an individual's voice as produced by the vocal tract.
- *Keystroke*: analysis of the way in which a user types on a computer keyboard; especially typing speed, length of holding down the keys and intervals between keystrokes are analysed.
- *Gait*: examination of the way and manner in which somebody walks.

1.2 Iris Anatomy

Iris (from Greek word for rainbow – *iris*) is an annulus-shaped part of eye, from the frontal view bounded by sclera on the outer and pupil on the inner periphery (see Figure 1.1). It is connected to the rest of eye by a ciliary muscle. Its principal function is to operate as a shade and control the amount of light entering retina via pupil. It is covered by a transparent, ca 0.6 mm thick cornea, which fluently merges into white sclera with visible red blood vessels. While eye is a ball with radius of 12 mm, cornea is the only protrusion (see Figure 1.2). It is another ball-shaped part of radius around 8 mm. Cornea is protecting only iris and partially also ciliary muscle. That is a reason why iris is always fully visible, either in a full contraction (which is arranged by dilator muscles) or full stretch (provided by sphincter).[6]

Iris is formed from the third one, up to the eigth month of gestation. At the time of delivery, the eye is complete and only color can finely change due to pigment accretion.[7] Iris is the only colored part of an eye and when one speaks about a color of eye, it is generally meant a color of iris. Despite a variety of different colors, all are generated by a single pigment – dark brown melanin. Its amount and distribution between stroma and epithelial cells determine the final color.[8] Although the color of both eyes of one person appears the same, it does not have to be the case. The quantity of melanin is only one factor and it sets the mean hue, but the real one can be whichever, according to a normal

distribution[9] (the phenomenon of two apparently different eyes of a single person is called *heterochromia*).

However, a color is not the most important attribute for iris recognition. There are also ligaments, furrows, ridges, crypts and freckles visible in any iris and their distribution is as random as genetics can be.[10] Nevertheless, their position within an eye is stable lifelong. That is an inevitable condition for a successful iris recognition.

1.3 History of Iris Recognition

Iris color as a biometric identifier has been used ever; however, Alphonse Bertillon was the first to start recording it for the French police record cards in 1886.[11]

More than 100 years later in 1987 Leonard Flom and Aran Safir patented an idea of the first iris recognition system; however, without any algorithm yet.[12]

John Daugman was the first to introduce modern automated iris recognition system no later than in 1994.[13] Although nearly 20 years have passed since Daugman's patent, almost all contemporary iris recognition systems still use the same algorithm. This algorithm, together with its potential improvements, will be described in this thesis.

1.4 Iris Recognition Deployment

The breakthrough in deployment of iris recognition methods happened in 2001, when the United Arab Emirates Ministry of Interior ordered to use iris recognition for all foreigners entering the country at all 32 air, land and sea ports. Since that time about 7,500,000 enrollments have been made and checked against a watch list resulting in more than 7 trillion iris comparisons altogether. So far 73,180 matches have been found between a person entering the country and one of 'persona non grata' on the watch list.[14]

During the same year iris recognition was also implemented in Netherlands, Amsterdam airport Schiphol. In contrast with the above-mentioned system, only those passengers who want to avoid waiting for the passport inspection are scanned (once registered, average time spent on iris-based control is 15 – 30 s).

The same approach was chosen in 2002 for Iris Recognition Immigration System in United Kingdom. In the past 10 years, over 1 million frequent travellers were enrolled at British airports.[15;16]

Nowadays, iris recognition is deployed at more than 100 airports all around the world.

However, immigration is only one of the high-visibility deployments of iris recognition. It is also an identification system in many prisons, first one being the York County Prison in Pennsylvania, USA.

The most massive project deploying iris recognition was for a long time the one organized by United Nations High Commissioner for Refugees. Within this project, all the refugees older than 6 years returning back home to Afghanistan after the war were enrolled in order to receive a repatriation assistance package.

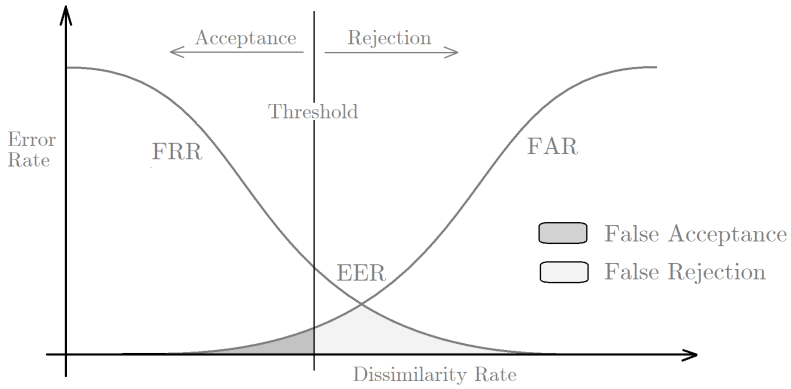


Figure 1.3: Interrelation between FAR and FRR

Between September 2002 and the end of the project in March 2006, 2.26 million irides – without any additional information – were captured.[17]

However, the most massive project in history of iris recognition, and biometrics at all, is without a doubt the project of Aadhaar. Aadhaar is a unique 12-digit identification number planned to be issued to every Indian citizen. This number is connected with personal and biometric information, including a photograph, ten fingerprints and iris code. The project was started in 2009 by the Unique Identification Authority of India *to provide a universal identification for every individual including the underprivileged*. By February 2012, more than 200 million Indian citizens have already undergone the iris scan.[16]

Within last years, iris recognition technics is becoming more and more feasible. This yields to its deployment also in smaller and less specific projects.[18]

1.5 Performance Measurement

Depending on priorities, there are several ways to compare the performance between iris recognition systems. It can be the speed of algorithm, variability of deployment between different systems etc. However, the numbers most significant for the performance of the system are called FAR, FRR and EER.

Every system can be, with respect to sample data, characterized by information about False Rejection Rate (FRR) at fixed False Acceptance Rate (FAR). *False Acceptance Rate* refers to the proportion of verification transactions with zero-effort wrongful claims of identity that are incorrectly confirmed, while *False Rejection Rate* is the proportion of verification transactions with truthful claims of identity that are incorrectly denied.[19]

Both the functions intersect at some point (i.e. FAR = FRR), defining *Equal Error Rate* (EER) of the system. Generally, decrease of FAR leads to increase of FRR and vice versa – this interrelation can be seen on Figure 1.3. *Genuine Rejection Rate* (GRR), defined as GRR := 1 – FAR, represents a complementary area to FAR.

False acceptance is usually worse than false rejection (letting an invader to enter a system is usually a more serious problem than incorrect refusal of entry as a genuine user can re-enroll). Therefore, the maximal acceptable FAR is often

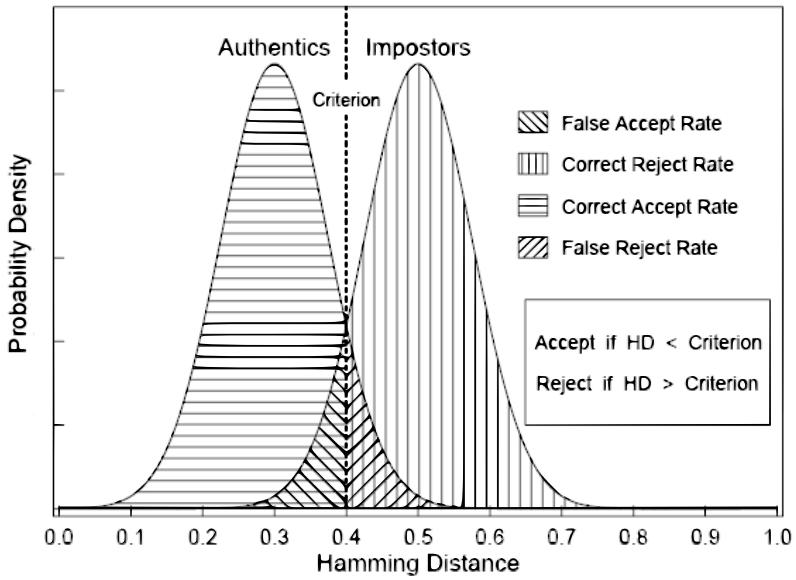


Figure 1.4: The problem of setting a suitable level of FAR and FRR

stated and then the FRR is measured according to it. For a fixed FAR, the performance of the system can be then characterized by a single number.

The problem of setting a suitable threshold value is also displayed in Figure 1.4, where distribution of distances between authentic and impostor irides is visualised. According to a chosen threshold, FAR either increases or decreases conversely to FRR.

Part I

Iris Recognition

Chapter 2

Image Acquisition

2.1 Eye Capture

Eye capture and image acquisition generally play a crucial role in iris recognition. Poor quality of the image results in a radical increase of FRR since it randomizes the image, while an acceptance is based on fail of the test of statistical independence between two iris captures. On the other hand, GRR remains more or less the same as it is largely independent of image quality.

Image segmentation and processing phases are especially sensitive to the following factors:

- occlusions (blink, eyelashes, hair etc.),
- incorrect illumination (reflections, light transitions etc.),
- blur (either motion or out-of-focus),
- off-gaze,
- insufficient resolution.

While other phases can be done repeatedly without further interaction of an investigated person, even the finest algorithm fails once the image source is poor.

The best results are claimed for the iris images taken in near-infrared illumination (which will be denoted as NIR hereafter). There are several reasons why it is preffered to visible wavelength light sources (VW).

The first one is the relative unintrusiveness and elimination of human eye light avoiding mechanisms, such as blinking and pupil motion (contraction). Another reason is better resistivity against the light pollution and reflections. Last, but not least, NIR rays are able to penetrate the surface of the iris and reveal structural patterns barely visible for heavily pigmented (dark-colored) irides.[16] On the other hand, less pigmented irides (such as blue) exhibit a more irregular pattern under VW light. VW images are also easier to implement in the multibiometric systems such as a combination with face recognition, which also operates under VW illumination.

In any case, NIR illumination sources emitting light with wavelengths between 700 and 900 nm are predominantly used (see Figure 2.1).[16;7] However, there are also several open VW iris databases.

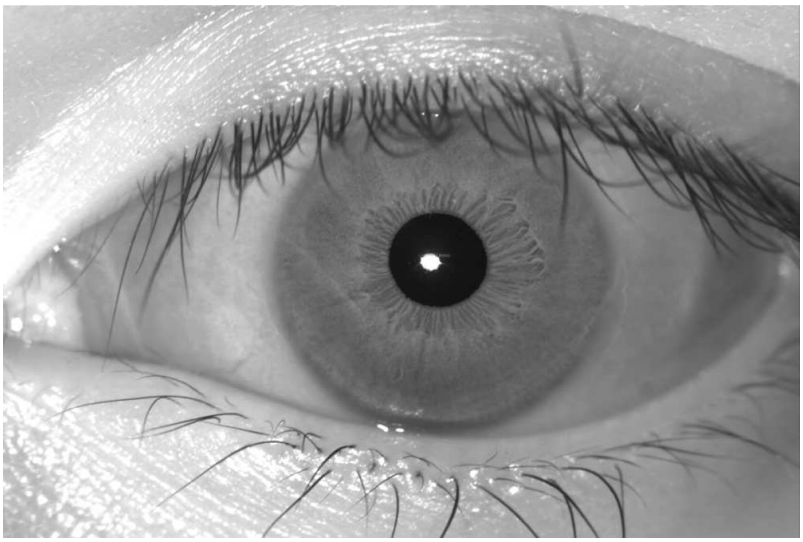


Figure 2.1: An example of eye captured under NIR illumination

To capture sufficient richness of iris pattern John Daugman suggests to expand the iris radius over a minimum of 70 pixels.[7] However, VGA images with an iris radius of 100 – 150 px are prevalent today.

2.2 Image Storage

There are several approaches to store an iris image for later use.

One way is to store just the extracted iris code. This approach has an advantage in its need for limited space only and fast iris code comparison. However, a significant drawback is the impossibility of re-processing the code extraction in the future as there could be a result of different algorithm requested.

Another approach is to store a full image of the iris. This option is also preferred by International Organization for Standardization.[20] Standard ISO/IEC 19794-6:2011 suggests to use the images in raw format. However, Rathgeb et al.[16] claim better results with the use of compressed format. Compressing algorithms smooth the peak intensity values for small areas which are in nature largely caused by noise. The best results are reported with the use of Joint Photographic Experts Group standards, especially JPEG 2000.[21]

As different algorithm authors claim different performance on different image sets there are several requirements for good databases:

- relevancy (large number of samples of the same person under different conditions)
- robustness (sufficient number of samples for supporting the claimed accuracy)
- representativeness (samples differing in gender, age etc.)
- sensor-variance

- time-variance (samples captured over large time spans, especially of the same person)
- pureness (images should not be edited)

A full list of open iris databases together with a brief description can be found in Iris Biometrics book by Rathgeb et al.[16]

Chapter 3

Image Preprocessing

3.1 Image Segmentation

The primary goal of image segmentation is to determine iris boundaries as accurately as possible. This process proceeds in two phases – in the first one region of interest with a probable iris center of gravity is selected. In the second iris boundaries are computed and iteratively precised.

3.1.1 ROI Definition

Precise definition of both iris center and the boundaries is computationally demanding and time complex. This is the main reason for incorporation of the first phase – thresholding – to roughly localize one of the iris boundaries and its center.

The thresholding process uses the difference in brightness between particular parts of an eye. Let b_p denote mean brightness of the pupil, b_i mean iris brightness, b_{sc} mean brightness of sclera, b_{sk} mean skin brightness and b_h mean brightness of hair and eyelashes. Then it holds[16]

- $b_p < b_i < b_{sc}$,
- $b_p < b_{sk} < b_{sc}$,
- $b_i < b_h$.

The approach to the localization process of an iris center can vary in dependence to the properties of the image. If the eye was captured under VW, then there is a significant difference between b_i and b_{sc}, while there is no sharp edge at the limbic boundary. Conversely, images acquired under NIR light conditions do not provide a sharp edge between iris and sclera, while there is a well distinguishable pupillary boundary.[16]

Regardless of the interdatabase variability, we will only consider the eye and its close neighborhood on the images now. Then investigating a horizontal line in the middle of the image, the brigtness function has the lowest values in the middle (around b_p), rises in both the directions to the right and to the left through the

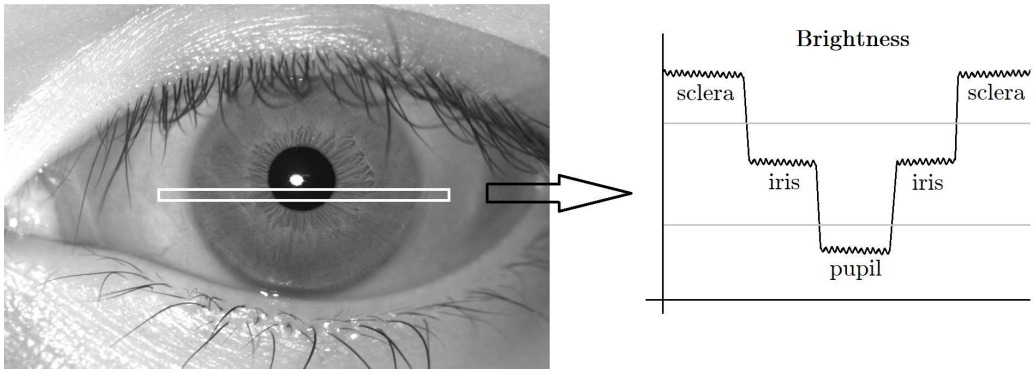


Figure 3.1: Investigation of a horizontal line in the middle of the image

values of b_i up to b_{sc} (see Figure 3.1). Depending on the image, there can be a brightness drop down to b_{sk} around the image sides.

Let us set the threshold value

$$t_{sc} := \frac{1}{2}\left(b_{sc} + \frac{1}{2}\left(b_i + b_{sk}\right)\right)$$

according to the past experience and define a new image J in a following way:

$$J\left(x,y\right) := \begin{cases} 0 & \text{if } I\left(x,y\right) \leq t_{sc}, \\ 255 & \text{if } I\left(x,y\right) > t_{sc}, \end{cases}$$

where $I\left(x,y\right)$ denotes the brightness (i.e. value) of image I at point $[x,y]$, 0 denotes black color and 255 the white one. Then the image J is white only on those places, where there is sclera on I, except for noise such as reflection.

There are two big white areas one on each of the left and the right halves of the image J. Let c_1 and c_2 denote their centers of gravity. In the case of the right capture of the eye, the joining of c_1 and c_2 should be a horizontal line. In the opposite case (let $\varphi \in (-\frac{\pi}{2}, \frac{\pi}{2}]$ denote an angle between the joining and the horizontal line), the face was probably tilted during the acquisition. Then we rotate the image over $-\varphi$ and continue the segmentation process with the adjusted image.

If the eye was captured correctly and the image covers the entire eye, then the center of the joining lies in close neighborhood of the center of iris gravity and represents the mid-point of the region of interest (ROI) R for precise iris center localization in the next phase. Meanwhile, the length of the black part of the joining represents an approximate value of the iris diameter. However, the image does not always contain a full sclera so there are also alternative ways to find the center of gravity area, such as the following one:

By setting a threshold value

$$t_p := \frac{1}{2}\left(\frac{1}{2}\left(b_i + b_{sk}\right) + b_p\right)$$

and creating an image

$$J\left(x,y\right) := \begin{cases} 0 & \text{if } I\left(x,y\right) \leq t_p, \\ 255 & \text{if } I\left(x,y\right) > t_p, \end{cases}$$

in the same way as with the previous method, we obtain a white image with a black circle in the middle representing the pupil. Depending on the illumination system there is probably a white figure inside the pupil caused by a reflection and black lines representing the eyelashes. However, this does not confuse the results significantly.

The center of gravity of the black area defines ROI R and the longest black line in J going through the center of gravity describes an approximate value of the pupil diameter.

In the case of VW images a threshold value for finding both iris and pupil can be implemented instead. However, the possibility of confusing the iris with the skin arises.

By both methods in the previous phase we have specified ROI $R, \mathrm{dom}(R) \subset \mathrm{dom}(I)$, which is a reduced area of I domain containing an iris center. We have also roughly estimated either the pupil (let it be denoted $2r_p$) or the iris ($2r_i$) diameter. According to the accuracy of estimation we set the toleration constants $\delta_1, \delta_2 \in \mathbb{R}_0^+$.

Depending on the light conditions during the image acquisition, pupil radius extensibility ranges from 11.5% – 14,7% (in miosa – full stretch of iris) to 69,2% – 91,8% (in mydriasa – full contraction of iris) of the outer iridian border radius.[6;22] To simplify the formulas, we will use values 10% and 80% as John Daugman does.[7]

Then an interval of radius interest for both the pupillary and limbic boundaries is

$$R_i := [0.1 \cdot r_i - \delta_1, r_i + \delta_2]$$

in the case of estimated iris radius r_i,

$$R_p := [r_p - \delta_1, 10 \cdot r_p + \delta_2]$$

respectively, if r_p has been estimated. If the full knowledge of radii is known, we can focus only on the smallest interval

$$R_{ip} := [r_p - \delta_1, r_i + \delta_2]\,,$$

$R_i \supset R_{ip} \subset R_p$.

While an image is from the mathematical point of view a continuous function, in the computer implementation pseudocodes (and in the Hough transform and noise mask generation phases), we always consider image to be a finite 2-D matrix consisting of 8-bit brightness values $\{0, \ldots, 255\}$.

Despite the fact that matrices have rows as a first coordinate and columns as a second one, we will use an inverse notation $[column, row]$ *– which is more similar to the cartesian system used for functions – not to confuse the reader.*

In this phase, there is no demand for precise results yet so we can work with a downscaled image to increase the processing speed.[23]

Algorithm: ROI Definition (1 of 2)

Input: Discrete image $image$ of resolution $width \times height$.
Output: Potential centres of gravity region corners tl, tr, bl, br and limits of boundary radius $minRadius$ and $maxRadius$.

for $i \leftarrow \text{Round}\left(\frac{1}{4}width\right)$ **to** $\text{Round}\left(\frac{3}{4}width\right)$
  do $line\left[i - \text{Round}\left(\frac{1}{4}width\right)\right] \leftarrow image\left[i, \text{Round}\left(\frac{1}{2}height\right)\right]$
if $\min\left(line\right) \leq 16$
  then $threshold \leftarrow \min\left(\text{mean}\left(line\right), 16\right)$
  else $threshold \leftarrow \text{mean}\left(line\right)$
for $i \leftarrow 0$ **to** $width$
  do { **for** $j \leftarrow 0$ **to** $height$
    do { **if** $image\left[i,j\right] \leq threshold$
      then $image\left[i,j\right] \leftarrow 0$
      else $image\left[i,j\right] \leftarrow 255$
$w \leftarrow 0$
$h \leftarrow 0$
$count \leftarrow 0$
for $i \leftarrow \text{Round}\left(\frac{1}{4}width\right)$ **to** $\text{Round}\left(\frac{3}{4}width\right)$
  do { **for** $j \leftarrow \text{Round}\left(\frac{1}{4}height\right)$ **to** $\text{Round}\left(\frac{3}{4}height\right)$
    do { **if** $image\left[i,j\right] = 0$
      then { $w \leftarrow w + i$
        $h \leftarrow h + j$
        $count \leftarrow count + 1$
$center \leftarrow \left[\text{Round}\left(\frac{w}{count}\right), \text{Round}\left(\frac{h}{count}\right)\right]$
$tl \leftarrow \left[center\left[0\right] - \delta_1, center\left[1\right] - \delta_2\right]$
$tr \leftarrow \left[center\left[0\right] - \delta_1, center\left[1\right] + \delta_2\right]$
$bl \leftarrow \left[center\left[0\right] + \delta_1, center\left[1\right] - \delta_2\right]$
$br \leftarrow \left[center\left[0\right] + \delta_1, center\left[1\right] + \delta_2\right]$
for $i \leftarrow \text{Round}\left(\frac{1}{4}width\right) + 1$ **to** $\text{Round}\left(\frac{3}{4}width\right) - 1$
  do { **for** $j \leftarrow \text{Round}\left(\frac{1}{4}height\right) + 1$ **to** $\text{Round}\left(\frac{3}{4}height\right) - 1$
    do { $count \leftarrow 0$
      for $m \leftarrow -1$ **to** 1
        do { **for** $n \leftarrow -1$ **to** 1
          do { **if** $image\left[i+m, j+n\right] = 0$
            then $count \leftarrow count + 1$
      $neighbors\left[i,j\right] \leftarrow count$
$index \leftarrow 0$
for $i \leftarrow \text{Round}\left(\frac{1}{4}width\right) + 1$ **to** $\text{Round}\left(\frac{3}{4}width\right) - 1$
  do { **for** $j \leftarrow \text{Round}\left(\frac{1}{4}height\right) + 1$ **to** $\text{Round}\left(\frac{3}{4}height\right) - 1$
    do { **if** $neighbors\left[i,j\right] \geq 8$
      then { $black\left[index\right] \leftarrow \left[i,j\right]$
        $index \leftarrow index + 1$

Algorithm: ROI Definition (2 of 2)

$dist \leftarrow 0$
for $i \leftarrow 0$ **to** $index - 1$
do $\begin{cases} \textbf{for } j \leftarrow i+1 \textbf{ to } index-1 \\ \textbf{do} \begin{cases} distances\,[dist] \leftarrow \text{DistanceBetween}\,(black\,[i]\,, black\,[j]) \\ dist \leftarrow dist + 1 \end{cases} \end{cases}$
$minRadius \leftarrow \frac{1}{2}\max\,(distances) - \delta_3$
$maxRadius \leftarrow 5\max\,(distances) - \delta_4$
return $(ul, ur, ll, lr, minRadius, maxRadius)$

3.1.2 Boundary Localization

Boundary localization is a process of finding both the pupillary and the limbic boundary, as accurate as possible, together with their centers. There are 2 major approaches to the boundary localization: *Integro-differential operator* and *Hough transform*.

3.1.2.1 Integro-Differential Operator

Integro-differential operator is a method proposed by John Daugman.[7;13] It is an iterative process of exhaustive searching for the greatest gradient over the circles with an origin from a specified region and radius within a given interval.

For each iteration a *Gaussian smoothing* is applied to an image. Gaussian smoothing G_σ is a filter which assigns each point a combination of its brightness and the brigtness of surrounding points according to a 2-D Gaussian distribution:

Definition 1. *Let* $\sigma \in \mathbb{R}$. *Then symmetric* 2-D Gaussian function *is a function* $G_\sigma : \mathbb{R}^2 \to \mathbb{R}$ *defined by*

$$G_\sigma\,(x, y) := \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}.$$

We call σ *a* standard deviation.

A Gaussian filter smooths the image, especially high brightness differences between close pixels or regions within the image. It therefore eliminates small edges such as those produced by eyelashes, veins etc., while it emphasizes the brigtness differences between large areas such as pupil, iris or sclera.

First iterations of the operation proceed on highly smoothed eye images to avoid indication of incorrect edges as boundaries. With each repetition and specification of center regions and radius intervals, the level of Gaussian smoothing decreases (i.e. σ lowers) to refine the edge localization. In each iteration we look for the circle with the maximum gradient using the formula

$$\arg\max_{r \in R_p, (x_0, y_0) \in \mathrm{dom}(R)} \left| \frac{\partial}{\partial r} \oint_{(r, x_0, y_0)} \frac{I\,(x, y)}{2\pi r} ds \right|. \tag{3.1}$$

Let us have a closer look at the equation (3.1) now. We are integrating over the circles in I, which are simple, piecewise smooth curves in $\mathbb{R}^2$:

Definition 2. *A* simple, piecewise smooth curve *in* $\mathbb{R}^2$ *is a set of points* $[x, y]$ *given by*

$$\begin{aligned} x &= \varphi_x(t), \\ y &= \varphi_y(t), \qquad t \in [a, b]\,, \end{aligned} \tag{3.2}$$

where $a, b \in \mathbb{R}$ *and*

- *functions* $\varphi_x(t), \varphi_y(t)$ *are continuous on* $[a, b]$,
- $(\varphi_x(t_1) = \varphi_x(t_2)) \wedge (\varphi_y(t_1) = \varphi_y(t_2))$ *holds for no* $t_1, t_2 \in (a, b)\,, t_1 \neq t_2$,
- *functions* $\varphi_x^{'}(t), \varphi_y^{'}(t)$ *are piecewise continuous on* $[a, b]$,
- $\varphi_x^{'}(t) = \varphi_y^{'}(t) = 0$ *holds for no* $t \in [a, b]$.

Note. A simple, piecewise smooth curve is said to be *closed* iff

$$(\varphi_x(a) = \varphi_x(b)) \wedge (\varphi_y(a) = \varphi_y(b))\,.$$

Example. A circle with center $[a, b] \in \mathbb{R}^2$ and radius $r \in \mathbb{R}^+$, given by

$$\begin{aligned} x &= a + r\cos t, \\ y &= b + r\sin t, \qquad t \in [0, 2\pi]\,, \end{aligned}$$

is a closed, simple, piecewise smooth curve on $\mathbb{R}^2$.

Definition 3. *Let* $\Gamma \subseteq \mathbb{R}^2$ *be a simple, piecewise smooth curve given by* (3.2) *and function* $f(x, y)$ *is continuous on* Γ. *Then we define a* line integral *as follows:*

$$\int_\Gamma f(x, y)ds := \int_a^b \sqrt{\left(\varphi_x'(t)\right)^2 + \left(\varphi_y'(t)\right)^2} f\left(\varphi_x(t), \varphi_y(t)\right) dt.$$

Note. If Γ is closed, then we write a line integral with a small circle in the middle:

$$\oint_\Gamma f(x, y)ds.$$

Example. Let $\Gamma \subseteq \mathbb{R}^2$ be a circle with center $[a, b] \in \mathbb{R}^2$ and radius $r \in \mathbb{R}^+$, given by

$$\begin{aligned} x &= a + r\cos t, \\ y &= b + r\sin t, \qquad t \in [0, 2\pi]\,, \end{aligned}$$

and $I : \mathbb{R}^2 \to \mathbb{R}$ is an image. Then

$$\begin{aligned} \oint_\Gamma \frac{I(x, y)}{2\pi r} ds &= \frac{1}{2\pi r} \int_0^{2\pi} \sqrt{(-r\sin t)^2 + (r\cos t)^2}\, I(a + r\cos t, b + r\sin t)\, dt \\ &= \frac{1}{2\pi r} \int_0^{2\pi} \sqrt{r^2\left(\sin^2 t + \cos^2 t\right)}\, I(a + r\cos t, b + r\sin t)\, dt \\ &= \frac{1}{2\pi r} \int_0^{2\pi} \sqrt{r^2}\, I(a + r\cos t, b + r\sin t)\, dt \\ &= \frac{1}{2\pi} \int_0^{2\pi} I(a + r\cos t, b + r\sin t)\, dt \end{aligned}$$

Definition 4. *Let $\mathcal{V}$ be a vector space over $\mathbb{R}$. A function $m : \mathcal{V} \times \mathcal{V} \to \mathbb{R}$ is called a* metric *iff it satisfies following three properties for any elements $x, y, z \in \mathcal{V}$:*

- non-negativity*:* $(m(x,y) \geq 0) \ \wedge \ (m(x,y) = 0 \Leftrightarrow x = y)$,
- symmetry*:* $m(x,y) = m(y,x)$,
- triangle inequality*:* $m(x,y) + m(y,z) \geq m(x,z)$.

Definition 5. *Let $\Gamma \subseteq \mathbb{R}^2$ be a set, $x \in \Gamma$ its point and $m(\cdot,\cdot)$ a metric. Then we call x an* interior point *of Γ iff*

$$\exists_{\delta>0} \ \ \forall_{y \in \mathbb{R}^2} \quad m(x,y) < \delta \Longrightarrow y \in \Gamma.$$

Definition 6. *Let $\Gamma \subseteq \mathbb{R}^2$, $f : \Gamma \to \mathbb{R}$, $a \in \mathbb{R}^2$ and x be an interior point of Γ. If the limit*

$$\lim_{h \to 0} \frac{f(x+ha) - f(x)}{h},$$

exists and if it is finite, then we call it a directional derivative of f along the vector *a at point x. We denote it $\frac{\partial}{\partial a} f(x)$ or $f_a'(x)$.*

Then

$$\frac{\partial}{\partial r} \oint_{(r,x_0,y_0)} \frac{I(x,y)}{2\pi r} ds = \lim_{h \to 0} \frac{1}{h} \left(\oint_{(r+h,x_0,y_0)} \frac{I(x,y)}{2\pi r} ds - \oint_{(r,x_0,y_0)} \frac{I(x,y)}{2\pi r} ds \right)$$

and in the equation (3.1), we are looking for the greatest absolut value of circle gradient over all the potential centers and radii.

In the integro-differential operator pseudocode, we consider the use of 'pupil' method in the ROI definiton phase, i.e. ROI defined according to an approximate pupil center and the radius interval set by an approximate pupil radius.

Algorithm: Integro-Differential Operator (1 of 3)

Input: Discrete image *image* of resolution $width \times height$, region of interest *roi* and boundary radius limits *minRadius* and *maxRadius*.
Output: Center of pupil $[pupilX, pupilY]$, pupil radius *pupilRadius*, center of iris $[irisX, irisY]$ and iris radius *irisRadius*.

procedure LineInt($center, integral, i, j$)
$x \leftarrow center[0] - i \sin\left(\frac{2\pi j}{vertices}\right)$
$y \leftarrow center[1] + i \cos\left(\frac{2\pi j}{vertices}\right)$
$integral \leftarrow integral + image[x, y]$
return ($integral$)

Algorithm: Integro-Differential Operator (2 of 3)

procedure Blur($image$)
for $i \leftarrow 0$ **to** $width - 1$
do { **for** $j \leftarrow 0$ **to** $height - 1$
do $temp[i, j] \leftarrow image[i, j]$
for $i \leftarrow 0$ **to** $width - 1$
do { $temp[i, -1] \leftarrow temp[i, 0]$
$temp[i, height] \leftarrow temp[i, height - 1]$
for $j \leftarrow 0$ **to** $height - 1$
do { $temp[-1, j] \leftarrow temp[0, j]$
$temp[width, j] \leftarrow temp[width - 1, j]$
$temp[-1, -1] \leftarrow temp[0, 0]$
$temp[-1, height] \leftarrow temp[0, height - 1]$
$temp[width, -1] \leftarrow temp[width - 1, 0]$
$temp[width, height] \leftarrow temp[width - 1, height - 1]$
for $i \leftarrow 0$ **to** $width - 1$
do { **for** $j \leftarrow 0$ **to** $height - 1$
do $blur[i, j] \leftarrow 0.64 \cdot temp[i, j] + 0.08 \cdot (temp[i - 1, j] +$
$+temp[i + 1, j] + temp[i, j - 1] + temp[i, j + 1]) +$
$+0.01 \cdot (temp[i - 1, j - 1] + temp[i + 1, j - 1] +$
$+temp[i + 1, j - 1] + temp[i + 1, j + 1])$
return ($blur$)

procedure FindBoundary($type, image, roi, minRadius, maxRadius$)
for each $center$ **in** roi
do { **for** $i \leftarrow minRadius$ **to** $maxRadius$
do { $int[center, i] \leftarrow 0$
if $type =$ 'pupil'
then { **for** $j \leftarrow 1$ **to** $vertices$
do { $int[center, i] \leftarrow$
$\leftarrow$ LineInt($center, int[center, i], i, j$)
else { **for** $j \leftarrow 1$ **to** $\frac{1}{8}vertices$
do { $int[center, i] \leftarrow$
$\leftarrow$ LineInt($center, int[center, i], i, j$)
for $j \leftarrow \frac{3}{8}vertices$ **to** $\frac{5}{8}vertices$
do { $int[center, i] \leftarrow$
$\leftarrow$ LineInt($center, int[center, i], i, j$)
for $j \leftarrow \frac{7}{8}vertices$ **to** $vertices$
do { $int[center, i] \leftarrow$
$\leftarrow$ LineInt($center, int[center, i], i, j$)
$int[center, i] \leftarrow 2 \cdot int[center, i]$
if $i = minRadius$
then $partial[center, i] \leftarrow 0$
else { $partial[center, i] \leftarrow int[center, i] - int[center, i - 1]$
$[center, radius] \leftarrow \arg\max(partial[center, i])$
return ($center[0], center[1], radius$)

Algorithm: Integro-Differential Operator (3 of 3)

main
$vertices \leftarrow 400$
$[pupilX, pupilY, pupilRadius] \leftarrow$ FindBoundary('pupil', Blur($image$), roi, $minRadius$, Round $(minRadius + \delta)$)
$[irisX, irisY, irisRadius] \leftarrow$ FindBoundary('iris', Blur($image$), roi, Round $(1.25 \cdot minRadius)$, $maxRadius$)
$specifiedRoi \leftarrow [(pupilX - \delta_c, pupilX + \delta_c), (pupilY - \delta_c, pupilY + \delta_c)]$
$[pupilX, pupilY, pupilRadius] \leftarrow$ FindBoundary('pupil', $image$, $specifiedRoi$, $pupilRadius - \delta_r, pupilRadius + \delta_r$)
$specifiedRoi \leftarrow [(irisX - \delta_c, irisX + \delta_c), (irisY - \delta_c, irisY + \delta_c)]$
$[irisX, irisY, irisRadius] \leftarrow$ FindBoundary('iris', $image$, $specifiedRoi$, $irisRadius - \delta_r, irisRadius + \delta_r$)
return $(pupilX, pupilY, pupilRadius, irisX, irisY, irisRadius)$

3.1.2.2 Hough Transform

A big drawback of the integro-differential operator method is the speed due to exhaustive searching for both the centres and the radii of pupillary and limbic boundaries. That is the reason for introducing Hough transform.

However, Hough transform requires discrete image consisting of finitely many points. Let the desired dimensions of discrete image be $height \times width$, $height$, $width \in \mathbb{N}$. Let also ROI, linearly transformed into the discrete image, be bounded by $y = h_{min}$ and $y = h_{max}$, $h_{min}, h_{max} \in \mathbb{N}_0$, $h_r := h_{max} - h_{min}$ vertically and $x = w_{min}$ and $x = w_{max}$, $w_{min}, w_{max} \in \mathbb{N}_0$, $w_r := w_{max} - w_{min}$ horizontally. Finally, let $r_{max} := \lceil \min \{r_i + \delta_2, 10 \cdot r_p + \delta_2\} \rceil \in \mathbb{N}$ be the maximal iris radius estimated in the previous phase.

Then the discrete image is a $height$-by-$width$ matrix I_d, where

$$I_d[i,j] := I\left(h_{min} + i\frac{h_{max} - h_{min}}{h-1}, w_{min} + j\frac{w_{max} - w_{min}}{w-1}\right)$$

for $i = 0, \ldots, height - 1$ and $j = 0, \ldots, width - 1$.

Circular Hough transform is then computed via Hough 3-D matrix of discrete parameters $(x_0, y_0, r), x_0, y_0, r \in \mathbb{N}_0$ describing circles in the image. Let this $(h_r + 1) \times (h_r + 1) \times r_{max}$ matrix be denoted by H. Then the value $H[x_0, y_0, r]$ accumulates the votes registering a boundary in I_d with center $[x_0, y_0]$ and radius r.

At the beginning of the process all the values of matrix H are set to 0. Then we scan pixel-wise all the discrete image I_d in a following way. If a sign of boundary is registered (i.e. there is a considerable difference between adjacent pixels), a point is incremented/subtracted for all the coordinates of H corresponding to the triples (x_0, y_0, r) having a boundary at this point (see Figure 3.2). If the boundary is merging from darker to lighter in the direction from point $[x_0, y_0]$, 1

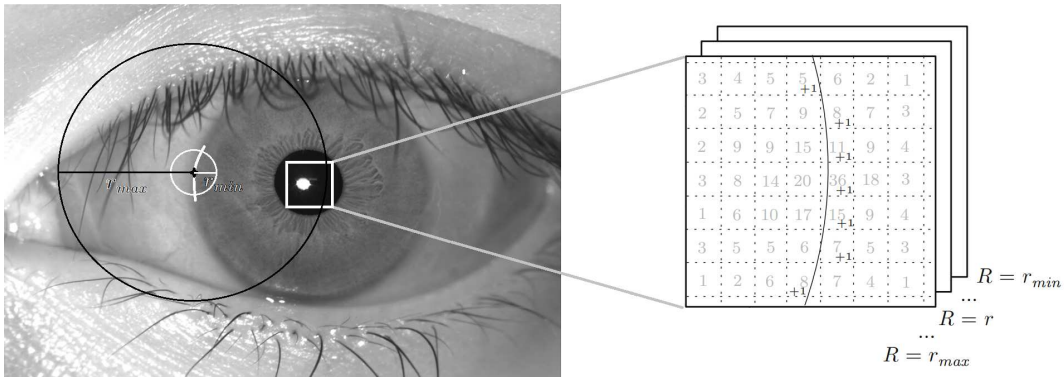


Figure 3.2: Computation of Hough 3-D matrix

is incremented. In the other case 1 is subtracted as both the pupil is darker than the iris and the iris darker than the sclera.

When the entire I_d image is scanned we look for the pixel with the most votes, i.e.

$$\arg \max_{(x_0,y_0,r)} H\left[x_0, y_0, r\right]$$

is searched. Depending on the value of r, there are 2 possibilities:

- if $r \approx r_i$, then $[x_0, y_0]$ and r denote a limbic boundary,
- we found a pupillary boundary otherwise.

We look for the second boundary now. The first option is the triple $\{x_0, y_0, r\}$ with the second greatest score; however, there are some restrictions. The second boundary should not have the center further than a pre-defined value and the radius should satisfy the ratio between pupillary and limbic boundaries, namely

$$r_p \in (0.1 \cdot r_i - \delta_1, 0.8 \cdot r_i - \delta_2)\,.$$

If we found a pupillary boundary first we search for the limbic one vice versa.

For the simplicity of the pseudocode, there are only 6-8 segments of the circumcircles voting. However, good implementations would in this case require more of them.

Algorithm: Hough Transform (1 of 4)

Input: Discrete image *image* of resolution $width \times height$, region of interest *roi* with corners tl, tr, bl, br and boundary radius limits *minRadius* and *maxRadius*.

Output: Center of pupil $[pupilX, pupilY]$, pupil radius *pupilRadius*, center of iris $[irisX, irisY]$ and iris radius *irisRadius*.

Algorithm: HOUGH TRANSFORM (2 of 4)

```
for each center in roi
  do { for i ← minRadius to maxRadius
       do hough[center, i] ← 0
for i ← 1 to width − 2
  do { for j ← tl[1] to bl[1]
       do { if Abs(image[i − 1, j] − image[i + 1, j]) > 100
            then { if image[i − 1, j] − image[i + 1, j] > 100
                   then type ← 1
                   else type ← −1
                   for k ← tl[0] to tr[0]
                   do { if ((i − k ≥ minRadius) ∧
                            ∧ (i − k ≤ maxRadius))
                        then hough[[k, j], i − k] ←
                            ← hough[[k, j], i − k] − type
                        else if ((k − i ≥ minRadius) ∧
                            ∧ (k − i ≤ maxRadius))
                        then hough[[k, j], k − i] ←
                            ← hough[[k, j], k − i] − type
for i ← tl[0] to tr[0]
  do { for j ← 1 to height − 2
       do { if Abs(image[i, j − 1] − image[i, j + 1]) > 100
            then { if image[i, j − 1] − image[i, j + 1] > 100
                   then type ← 1
                   else type ← −1
                   for k ← tl[1] to bl[1]
                   do { if ((j − k ≥ minRadius) ∧
                            ∧ (j − k ≤ maxRadius))
                        then hough[[i, k], j − k] ←
                            ← hough[[i, k], j − k] − type
                        else if ((k − j ≥ minRadius) ∧
                            ∧ (k − j ≤ maxRadius))
                        then hough[[i, k], k − j] ←
                            ← hough[[i, k], k − j] − type
```

Algorithm: HOUGH TRANSFORM (3 of 4)

```
for i ← 1 to width − 2
do { for j ← 1 to height − 2
     do { if Abs(image[i − 1, j − 1] − image[i + 1, j + 1]) > 100
          then { if image[i − 1, j − 1] − image[i + 1, j + 1] > 100
                   then type ← 1
                   else type ← −1
                 for k ← tl[1] to bl[1]
                 do { if k < j
                      then type ← (−1) · type
                      radius ← Round(Sqrt(2(k − j)^2))
                      if ((radius ≥ minRadius) ∧
                              ∧ (radius ≤ maxRadius) ∧
                              ∧ ([i + (k − j), k] ∈ roi))
                      then hough[[i + (k − j), k], radius] ←
                              ← hough[[i + (k − j), k], radius]−
                              − type
                      else if ((radius ≥ minRadius) ∧
                              ∧ (radius ≤ maxRadius) ∧
                              ∧ ([i − (k − j), k] ∈ roi))
                      then hough[[i − (k − j), k], radius] ←
                              ← hough[[i − (k − j), k], radius]+
                              + type
                    }
               }
        }
for i ← 1 to width − 2
do { for j ← 1 to height − 2
     do { if Abs(image[i + 1, j − 1] − image[i − 1, j + 1]) > 100
          then { if image[i + 1, j − 1] − image[i − 1, j + 1] > 100
                   then type ← 1
                   else type ← −1
                 for k ← tl[1] to bl[1]
                 do { if k < j
                      then type ← (−1) · type
                      radius ← Round(Sqrt(2(k − j)^2))
                      if ((radius ≥ minRadius) ∧
                              ∧ (radius ≤ maxRadius) ∧
                              ∧ ([i − (k − j), k] ∈ roi))
                      then hough[[i − (k − j), k], radius] ←
                              ← hough[[i − (k − j), k], radius]−
                              − type
                      else if ((radius ≥ minRadius) ∧
                              ∧ (radius ≤ maxRadius) ∧
                              ∧ ([i + (k − j)] ∈ roi))
                      then hough[[i + (k − j), k], radius] ←
                              ← hough[[i + (k − j), k], radius]+
                              + type
                    }
               }
        }
```

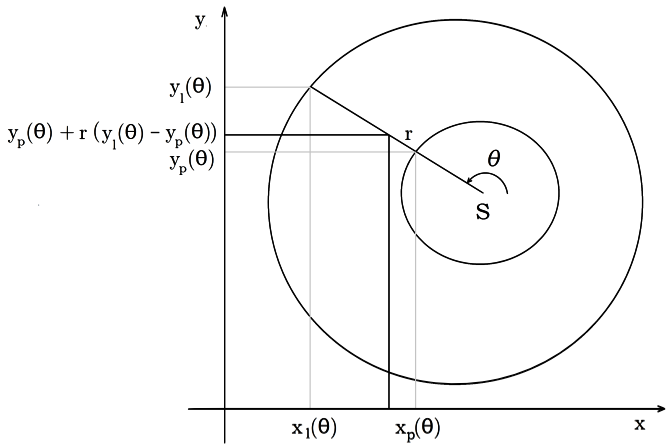


Figure 3.3: Pupillary and limbic borders parametrization

Algorithm: HOUGH TRANSFORM (4 of 4)

$max \leftarrow -\infty$
for each *center* **in** *roi*
do { **for** $radius \leftarrow minRadius$ **to** $\text{Round}(minRadius + 2\delta)$
do { **if** $hough[center, radius] > max$
then { $max \leftarrow hough[center, radius]$
$[pupilX, pupilY, pupilRadius] \leftarrow$
$\leftarrow [center[0], center[1], radius]$
$max \leftarrow -\infty$
for each *center* **in** *roi*
do { **for** $radius \leftarrow \text{Round}(1.2 \cdot minRadius)$ **to** $maxRadius$
do { **if** $hough[center, radius] > max$
then { $max \leftarrow hough[center, radius]$
$[irisX, irisY, irisRadius] \leftarrow$
$\leftarrow [center[0], center[1], radius]$
return $(pupilX, pupilY, pupilRadius, irisX, irisY, irisRadius)$

3.2 Iris Normalization

Now both the outer and inner iris boundaries are parametrized and we can prepare an iris pattern for the feature extraction. The inner – pupillary – boundary is parametrized by

$$b_p(\theta) = (x_p(\theta), y_p(\theta)) \tag{3.3}$$

and the outer one – limbic boundary – is parametrized by

$$b_l(\theta) = (x_l(\theta), y_l(\theta)) \tag{3.4}$$

where $\theta \in [0, 2\pi)$ denotes a counterclockwise rotation around the center of gravity $S = (x_0, y_0)$ of the pupillary circle (see Figure 3.3).

Figure 3.4: Normalized iris texture

These boundaries are not generally concentric – the nasal displacement can be as much as 15%.[7]. Moreover, they are not equidistant over time as the pupil dilates in dependence to the amount of light entering the eye.

However, the iris has characteristics of a rubber sheet that stretches and contracts with the pupillary reflex and its texture and markings stretch and shrink accordingly.[6;13;24] We only need to normalize the iris in a way that the resulting image maintains all the data space-invariant, regardless of the actual image acquisition conditions including the brightness of the environment at the time of the eye capture.

For these purposes a polar transformation is applied. The transformation $T : \mathbb{R}^2 \to [0, 2\pi) \times [0.1, 0.9]$ is defined by

$$T \begin{pmatrix} \theta \\ r \end{pmatrix} := \begin{pmatrix} x_p(\theta) + r\left[x_l(\theta) - x_p(\theta)\right] \\ y_p(\theta) + r\left[y_l(\theta) - y_p(\theta)\right] \end{pmatrix}$$

and it transforms the iris texture from the Cartesian coordinates into the doubly dimensionless image in the polar coordinates (see Figure 3.4).

We preserve only 80% of the iris texture as the information around the borders inclines to be more blurred and also inaccurate due to the segmentation faults.[13;25]

The images vary in their acquisition resolution and the effective number of pixels per iris radii vary accordingly. However, there is no need for unification of either the dimensions of the resulting image or its ratio. The prevailing aspect ratio used by important systems lies between $1 : 5$ and $1 : 9$.[16;23;24;26;27] However, we must not forget the size and ratio and take it into consideration during the feature extraction phase, especially at the time of the wavelet frequency specification.

Algorithm: Normalization (1 of 2)

Input: Discrete image *image* of resolution $width \times height$, center of pupil $[pupilX, pupilY]$, pupil radius $pupilRadius$, center of iris $[irisX, irisY]$ and iris radius $irisRadius$.

Output: Discrete image *normImage* of pre-defined resolution $width \times height$.

Algorithm: NORMALIZATION (2 of 2)

$$
\begin{array}{l}
\textbf{for } i \leftarrow 0 \textbf{ to } width - 1 \\
\quad \textbf{do} \begin{cases}
pupilX\theta \leftarrow pupilX + pupilRadius \cdot \sin(\frac{i}{width} 2\pi) \\
pupilY\theta \leftarrow pupilY + pupilRadius \cdot \cos(\frac{i}{width} 2\pi) \\
irisX\theta \leftarrow irisX + irisRadius \cdot \sin(\frac{i}{width} 2\pi) \\
irisY\theta \leftarrow irisY + irisRadius \cdot \cos(\frac{i}{width} 2\pi) \\
\textbf{for } j \leftarrow 0 \textbf{ to } height - 1 \\
\quad \textbf{do} \begin{cases}
normImage[i,j] \leftarrow image[pupilX\theta \ + \ (0.1 + 0.8\frac{j}{height-1}) \\
\quad (irisX\theta - pupilX\theta), pupilY\theta + (0.1 + 0.8\frac{j}{height-1}) \\
\quad (irisY\theta - pupilY\theta)]
\end{cases}
\end{cases} \\
\textbf{return } (normImage)
\end{array}
$$

3.3 Noise Masks

While iris patterns are stable over time there are other aspects which differ between distinct eye captures. One of them which influences the effectivity of the recognition process the most is the noise. Noise is generally of two types: background errors caused by sensor, iris skew etc., and burst errors caused by occlusion of various sorts. While the first type is the reason for implementing sophisticated feature extractors, described in consecutive chapters, the latter one is minimised by using noise masks.

Occlusion is mainly caused by eyelids, eyelashes and light reflection. Eyelids are always overlapping the iris from the upmost and the downmost parts. This is eliminated by using only the sides of the iris. If the eyelids cover most of the eye the recognition process will fail anyway. However, if one wants to be precise, these can be described by parabolas[7] or by line Hough transform.[16]

Light reflections are generally small objects with high luminance values. In implementations they are usually found as regions of higher intensity than the surrounding area.

Let I be a discrete eye image, $c \in [0, 255]$ be a threshold value and $M(x, y)_r$ be a mean of the surrounding area of the point $[x, y]$ within a radius of r. c usually has values around 60 and the radius used varies around 10 pixels.[16]

Then *noise mask* is a function $m : I \rightarrow \{0, 1\}^{height \times width}$ defined as

$$m(x, y) := \begin{cases} 0 & \text{if } I(x, y) \geq M(x, y)_r + c, \\ 1 & \text{otherwise.} \end{cases}$$

However, there are also computationally less demanding methods:[26]

Let $N(x, y)$ be a set of intensity values of reliable (i.e. with brightness between 10 and 240) neighboring pixels of the point $[x, y]$. Then the high-brightness noise mask is computed accordingly:

$$m_h(x, y) := \begin{cases} 0 & \text{if } \{I(x, y) \geq 240\} \vee \{[I(x, y) \geq 160] \wedge [\max(N(x, y)) < 50]\}, \\ 1 & \text{otherwise.} \end{cases}$$

Figure 3.5: An example of a noise mask

The approach is just the same with eyelashes the only difference being that we are looking for regions of intensity, which this time is lower than the surrounding area:

$$m_l(x,y) := \begin{cases} 0 & \text{if } \{I(x,y) \leq 10\} \vee \{[I(x,y) \leq 80] \wedge [\min(N(x,y)) > 200]\}, \\ 1 & \text{otherwise.} \end{cases}$$

Another possibility is to find eyelashes by using 1-D Gabor filters,[16] which will be introduced in the chapter concerned with Feature Extraction.

Aggregate noise mask is then obtained by eliminating all the unreliable pixels:

$$m(x,y) := m_l(x,y) \wedge m_h(x,y).$$

For obtaining better results, small and large areas of zeros can be set to 1 in order not to lose important parts of information. Essentially, it means connected zero-components in m which contain less then 10 or more than 1,000 pixels.[16]

Apart from a noise mask generation, all points on the original image proclaimed to be noise are overwritten by the local mean intensity to influence a convolution with filters as little as possible.[26]

The noise mask described is a special kind of image and it can be treated the same way including determination of reliability of the bits via normalization and use of localized filters. We will use this fact in the later phases, namely in the Iris Comparison chapter.

In the pseudocode, we will describe the second method of the noise mask generation described.

Algorithm: Noise Mask Generation (1 of 2)

Input: Discrete image *image* of resolution $width \times height$.
Output: Discrete binary image *mask* of resolution $width \times height$.

Algorithm: NOISE MASK GENERATION (2 of 2)

```
procedure LOW(centerX, centerY)
 low ← 1
 for i ← −1 to 1
      do { for j ← −1 to 1
           do { if ((i = 0 ∧ j = 0) ∨ (centerX + i < 0) ∨ (centerX + i ≥
                  ≥ width) ∨ (centerY + j < 0) ∨ (centerY + j ≥ height))
                then break
              if image[centerX + i, centerY + j] ≥ 50
                then { low ← 0
                     { exit
 return (low)

procedure HIGH(centerX, centerY)
 high ← 1
 for i ← −1 to 1
      do { for j ← −1 to 1
           do { if ((i = 0 ∧ j = 0) ∨ (centerX + i < 0) ∨ (centerX + i ≥
                  ≥ width) ∨ (centerY + j < 0) ∨ (centerY + j ≥ height))
                then break
              if image[centerX + i, centerY + j] ≤ 200
                then { high ← 0
                     { exit
 return (high)

main
 for i ← 0 to width − 1
      do { for j ← 0 to height − 1
           do { if ((image[i, j] ≥ 240) ∧ (image[i, j] ≤ 10))
                then mask[i, j] ← 0
                else mask[i, j] ← 1
 for i ← 0 to width − 1
      do { for j ← 0 to height − 1
           do { if mask[i, j] = 0
                then break
              if (Low(i, j) = 1 ∧ image[i, j] ≥ 160)
                then mask[i, j] ← 0
              if (High(i, j) = 1 ∧ image[i, j] ≤ 80)
                then mask[i, j] ← 0
 return (mask)
```

Chapter 4

Feature Extraction

Once we have transformed the iris texture into a doubly dimensionless polar coordinate system, our goal is to extract the unique iris information. We want to do this in a way that will achieve as high a level of independency to eye-capture conditions as possible. The transformed image is still highly influenced by noise, light conditions during the acquisition, motion blur and off-gaze that the pure XORing would not lead to satisfactory results. Our motivation is to find a method for feature extraction only sensitive to elements resilient against different eye-capture conditions.

The method most widely used is the wavelet transform. Wavelets are sensitive to edges and these are characteristic to specific iris textures in general. Localized wavelet properties, together with the convolution operation, also reduce the importance of precise alignment of the acquired biometric texture to the database one.

Let us start with the mathematical background of wavelets and frames first:

4.1 Mathematical Background

Definition 7. *A* wavelet ψ *is a function in Hilbert space* $\mathcal{L}^2(\mathbb{R}^n), n \in \mathbb{N}$, *such that the system*

$$\left\{\psi_{j,k} := 2^{j/2}\psi\left(2^j x - k\right), j, k \in \mathbb{Z}\right\}$$

is an orthonormal basis for $\mathcal{L}^2(\mathbb{R}^n)$.

Note. The wavelet $\psi_{0,0} =: \psi$ is generally called a *mother wavelet*, number j *dilation parameter* and the number k specifies the *translation* of the mother wavelet.[28]

Example. Haar wavelet (see Figure 4.1) is a function defined on the real line $\mathbb{R}$ as

$$H(t) := \begin{cases} 1 & \text{if } t \in \left[0, \frac{1}{2}\right), \\ -1 & \text{if } t \in \left[\frac{1}{2}, 1\right], \\ 0 & \text{otherwise.} \end{cases}$$

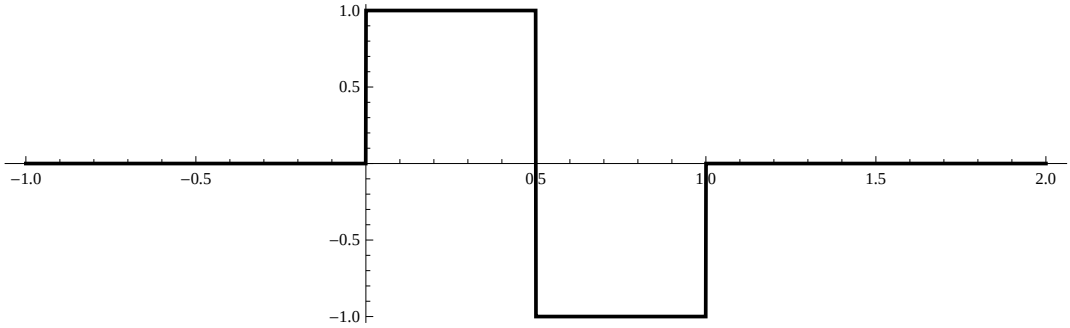


Figure 4.1: Haar wavelet

Definition 8. *A sequence* $\{\xi_n\}$ *in a Hilbert space* $\mathcal{L}^2(\mathbb{R}^2)$ *is a* frame *iff there exist numbers* $A, B \in \mathbb{R}^+$ *such that for all* $x \in \mathcal{L}^2(\mathbb{R}^2)$ *we have*

$$A\|x\|^2 \leq \sum_n |\langle x, \xi_n \rangle|^2 \leq B\|x\|^2. \tag{4.1}$$

Note. Considering an above-mentioned definition, we can also establish:

- The numbers A, B are called the *frame bounds*.
- Let A_0 be a supremum over all lower frame bounds A. Then A_0 is called *optimal lower frame bound*. Analogically for *optimal upper frame bound*.
- Number B_0/A_0 is a *tightness* of the frame. We say that the frame is tight iff $B_0/A_0 = 1$.
- Number $(A_0 + B_0)/2$ is a *redundancy* of the frame.
- The frame is *exact* iff it ceases to be a frame whenever any single element is deleted from the sequence.

Definition 9. 2-D Gabor wavelet *with center in an origin of a plane is a function*

$$g_{\theta,\phi,\sigma^2}(x, y) := e^{-\frac{x^2+y^2}{\sigma^2}} e^{-2\pi i(\phi x + \theta y)}, \tag{4.2}$$

where σ *is a standard deviation of the Gaussian kernel,* $\arctan\left(\frac{\theta}{\phi}\right)$ *planar orientation of the sinusoid and* $\sqrt{\theta^2 + \phi^2}$ *its spatial frequency.*

Theorem 1. *Let* $g_{\theta,\phi,\sigma^2}(x, y)$ *denote a 2-D Gabor wavelet with fixed values for* σ^2, θ *and* ϕ.

Then the system

$$\left\{ g^{j,k}_{\theta,\phi,\sigma^2} := 2^{j/2} g_{\theta,\phi,\sigma^2}\left(2^j x - k\right), j, k \in \mathbb{Z} \right\}$$

forms a frame.

Proof. Proof of this theorem is too complicated for the scope of this thesis. However, the tenor is a direct corollary of the Density Theorem, which can be, together with its proof, found in Christopher Heil's Chapter 7 – *The Density Theorem and the Homogeneous Approximation Property for Gabor frames* – of the book *Representations, Wavelets, and Frames.*[29]

□

Theorem 2. *Let* $\{\xi_n\}$ *be a frame in* $\mathcal{L}^2(\mathbb{R}^2)$. *Then*

$$\dim(\{\xi_n\}) \geq \dim\left(\mathcal{L}^2\left(\mathbb{R}^2\right)\right).$$

Proof. Let $\dim(\{\xi_n\}) < \dim(\mathcal{L}^2(\mathbb{R}^2))$. Then there exist images in $\mathcal{L}^2(\mathbb{R}^2)$ which cannot be generated by $\{\xi_n\}$. At least one of them – let it be $0 \neq I \in \mathcal{L}^2(\mathbb{R}^2)$ – must be orthogonal to all elements ξ_n. Then $\langle I, \xi_n \rangle = 0$ for all n, hence

$$A\|I\|^2 \leq \sum_n |\langle I, \xi_n \rangle|^2 = 0.$$

But this holds from the definition of norm only for $I = 0$ as $A > 0$, which is a contradiction.

□

Definition 10. *Let* $f : \mathbb{R}^n \to \mathbb{R}$ *and for all* $a, x \in \mathbb{R}^n$ *and* $l \in \mathbb{N}$ *exist*

$$f_a^{(l)}(x) := \frac{\partial^l}{\partial a^l} f(x)$$

and $\left|f_a^{(l)}(x)\right| < \infty$.

Then we call f *to be* Schwartz function *in* $\mathbb{R}^n$ *iff*

$$\forall_{a \in \mathbb{R}^n} \quad \forall_{m,l \in \mathbb{N}} \quad \sup_{x \in \mathbb{R}^n} \left|x^m f_a^{(l)}(x)\right| < \infty.$$

Definition 11. *Let* f *and* g *be Schwarz functions in* $\mathbb{R}^2$.

Then we define operation of 2-D convolution *as follows:*

$$(f * g)(s,t) := \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\sigma,\tau) g(s-\sigma, t-\tau) d\sigma d\tau.$$

Corollary. Let $I \in \mathcal{L}^2(\mathbb{R}^n)$ be an image and $g_{\theta,\phi,\sigma^2}(x,y)$ denote a 2-D Gabor wavelet with frequency θ and orientation ϕ.

Then

$$G_{\theta,\phi,\sigma^2}(x,y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} I(\xi,\psi)\, g_{\theta,\phi,\sigma^2}(x-\xi, y-\psi)\, d\xi d\psi$$

represents the response of g_{θ,ϕ,σ^2} to an image I at point (x,y) of an image plane.

Proposition 3. *2-D convolution is commutative.*

Proof. Let f and g be arbitrary Schwarz functions in $\mathbb{R}^2$. By substitutions $u := s - \sigma$ and $v := t - \tau$ we get

$$(f * g)(s,t) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\sigma,\tau) g(s-\sigma, t-\tau) d\sigma d\tau =$$

$$= \int_{\infty}^{-\infty} \int_{\infty}^{-\infty} f(s-u, t-v) g(u,v) - du - dv =$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(s-u,t-v)g(u,v)dudv =$$
$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(u,v)f(s-u,t-v)dudv = (g*f)(s,t)$$

□

Proposition 4. *2-D convolution is associative.*

Proof. Let f, g and h be arbitrary Schwarz functions in $\mathbb{R}^2$. Then

$$\begin{aligned}
&(f*(g*h))(s,t) = \\
&= f(s,t) * \left(\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(\sigma,\tau)h(s-\sigma,t-\tau)d\sigma d\tau\right) = \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\xi,\psi)\left(\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(\sigma,\tau)h(s-\sigma-\xi,t-\tau-\psi)d\sigma d\tau\right)d\xi d\psi = \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\xi,\psi)g(\sigma,\tau)h(s-\sigma-\xi,t-\tau-\psi)d\sigma d\tau d\xi d\psi = (*)
\end{aligned}$$

By substitutions $\sigma := \sigma - \xi$ and $\tau := \tau - \psi$ we get

$$\begin{aligned}
(*) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\xi,\psi)g(\sigma-\xi,\tau-\psi)h(s-\sigma,t-\tau)d\sigma d\tau d\xi d\psi = \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\left(\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\xi,\psi)g(\sigma-\xi,\tau-\psi)\right)h(s-\sigma,t-\tau)d\sigma d\tau d\xi d\psi = \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\left((f*g)(\sigma,\tau)\right)h(s-\sigma,t-\tau)d\sigma d\tau = \\
&= ((f*g)*h)(s,t)
\end{aligned}$$

□

Definition 12. *Let $f : \mathbb{R}^2 \to \mathbb{R}$ and $f_x, f_y : \mathbb{R} \to \mathbb{R}$.*
We call f a separable function *iff*

$$\forall x, y \in \mathbb{R} \quad f(x,y) = f_x(x) \cdot f_y(y).$$

Theorem 5. *Let f and g be Schwarz functions in $\mathbb{R}^2$ and let g be separable in a way that $g(x,y) = g_x(x) \cdot g_y(y)$.*
Then

$$(f*g)(x,y) = (f(x,y) * g_x(x)) * g_y(y) = (f(x,y) * g_y(y)) * g_x(x).$$

Proof. By a commutativity of convolution we get

$$\begin{aligned}
(f*g)(x,y) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\xi,\psi)g(x-\xi,y-\psi)d\xi d\psi = \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x-\xi,y-\psi)g(\xi,\psi)d\psi d\xi = (*)
\end{aligned}$$

Separability of g gives us

$$(*) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x-\xi, y-\psi)g_x(\xi)g_y(\psi)d\psi d\xi = (\heartsuit)$$

and the associativity of convolution finishes the proof:

$$(\heartsuit) = \int_{-\infty}^{\infty}\left(\int_{-\infty}^{\infty} f(x-\xi, y-\psi)g_x(\xi)d\xi\right) g_y(\psi)d\psi = (f(x,y) * g_x(x)) * g_y(y),$$

$$(\heartsuit) = \int_{-\infty}^{\infty}\left(\int_{-\infty}^{\infty} f(x-\xi, y-\psi)g_y(\psi)d\psi\right) g_x(\xi)d\xi = (f(x,y) * g_y(y)) * g_x(x).$$

□

Theorem 6. *2-D Gabor wavelet is separable.*

Proof. Using the equation 4.2, we get

$$g_{\theta,\phi,\sigma^2}(x,y) = e^{-\frac{x^2+y^2}{\sigma^2}} e^{-2\pi i(\phi x+\theta y)} = \\ = e^{-\frac{x^2}{\sigma^2}} e^{-2\pi i\phi x} \cdot e^{-\frac{y^2}{\sigma^2}} e^{-2\pi i\theta y} = g^x_{\theta,\phi,\sigma^2}(x) \cdot g^y_{\theta,\phi,\sigma^2}(y)$$

□

Corollary. The response of Gabor wavelet g_{θ,ϕ,σ^2} to an image I from Corollary 4.1 can be calculated by separate convolutions with $g^x_{\theta,\phi,\sigma^2}$ and $g^y_{\theta,\phi,\sigma^2}$, i.e.

$$G_{\theta,\phi,\sigma^2}(x,y) = \left(I * g_{\theta,\phi,\sigma^2}\right)(x,y) = \\ = \left(\left(I * g^x_{\theta,\phi,\sigma^2}\right) * g^y_{\theta,\phi,\sigma^2}\right)(x,y) = \left(\left(I * g^y_{\theta,\phi,\sigma^2}\right) * g^x_{\theta,\phi,\sigma^2}\right)(x,y).$$

Definition 13. *Gray code is an ordering of 2^n binary numbers such that only one bit changes from one entry to the next.*

4.2 Deployment

As described in the mathematical section, it is worth pointing out once again that 2-D wavelets have a useful property of being generators of the Hilbert space $\mathcal{L}^2(\mathbb{R}^2)$. Hence, every image $I \in \mathcal{L}^2(\mathbb{R}^2)$ can be uniquely written as

$$I = \sum_{j\in\mathbb{Z}}\sum_{k\in\mathbb{Z}} c_{jk}\psi_{j,k}, \tag{4.3}$$

where $\psi_{j,k}$ are members of the wavelet family ψ and c_{jk} are uniquely chosen scalars for all $j,k \in \mathbb{Z}$. Any such an image I can be therefore reconstructed just by knowing ψ and $c_{jk}, j,k \in \mathbb{Z}$ and a Gabor wavelet (see Definition 9) was considered to be one of these wavelets for a long period of time.[30]

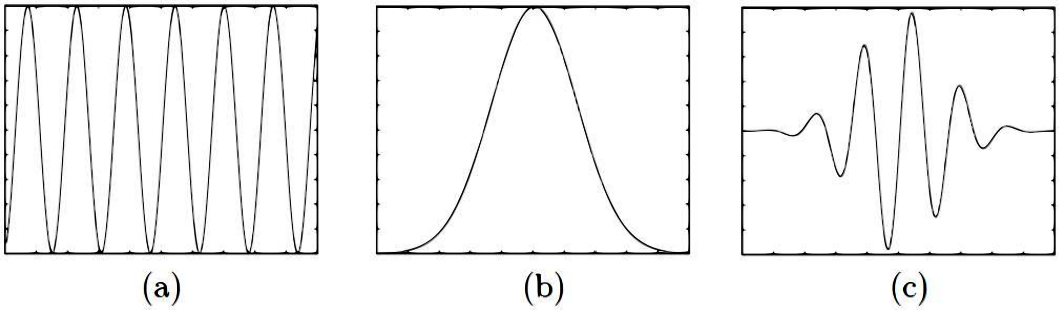


Figure 4.2: Modulation of 1-D Gabor wavelet: (a) 1-D sinusoid, (b) Gaussian function, (c) 1-D Gabor wavelet

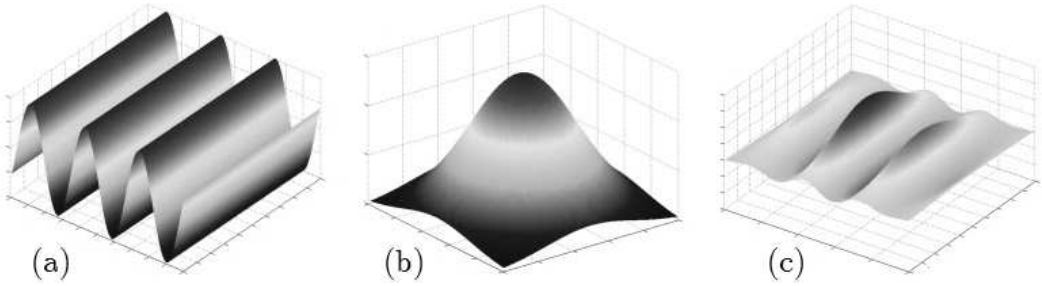


Figure 4.3: Modulation of 2-D Gabor wavelet: (a) 2-D sinusoid, (b) Gaussian kernel, (c) 2-D Gabor wavelet

A Gabor wavelet is obtained by modulation of a sinusoid with a Gaussian function.[31] The case of one dimension is illustrated in Figure 4.2. 1-D Gabor wavelet is frequently deployed in image processing because of its usefulness for obtaining a response to the signal in its localized part and edge detection capabilities. In the case of 2-D, a Gabor wavelet is retrieved by modulating 2-D sinusoid with a Gaussian kernel, which can be seen in Figure 4.3. Figure 4.4 shows 2-D plots of 2-D Gabor wavelets at various frequencies and orientation.

It is 2-D Gabor wavelet which was chosen as a filter in the feature extraction phase in iris recognition. The reason sources from neurophysiology. In 1962, Hubel and Wiesel discovered that the receptive fields in the back region of a cat's brain have a characteristic shape.[32] In 1980, Marčelja showed that this shape can be best modeled by a family of Gabor wavelets.[33] This area of a brain, where receptive fields occur (reffered to as primary visual cortex), is the principal projection area for visual information and consists of simple cells that receive input from neurons in the eye (via an area called lateral geniculate nucleus).[30] It is hypothesized that it works the same for all the mammals, including human.[34] Moreover, John Daugman reports satisfactory results gained by implementation of 2-D Gabor wavelets in iris recognition.[7;13].

Despite the fact that he refers to Gabor wavelets in his patent only at its general form (4.2)[13], the formula of the Gabor wavelets used in nature is believed to be[34]

$$g_{\theta,\phi,\sigma^2}(x,y) = \frac{\theta}{\sqrt{2\pi}\kappa}\left\{e^{i(\theta x\cos\phi+\theta y\sin\phi)} - e^{-\frac{x^2}{2}}\right\} e^{-\frac{\theta}{8\kappa^2}\left[4(x\cos\phi+y\sin\phi)^2+(-x\sin\phi+y\cos\phi)^2\right]},$$

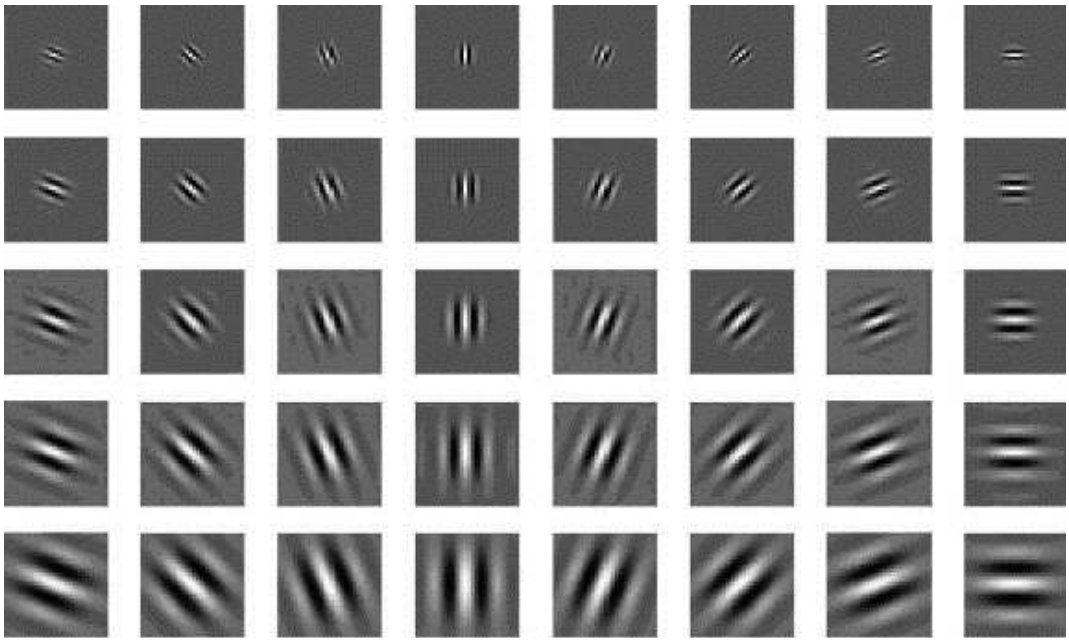


Figure 4.4: 2-D Gabor wavelets at different frequencies and orientation

where $3 \approx \kappa \in \mathbb{R}$.

However, Gabor *wavelet* is in fact not wavelet as this family of functions does not respect the condition of orthogonality.[35;36] This was originally proven by Balian-Low Theorem in early 1980s.[37;38] Nevertheless, Gabor wavelets dedicate a vital property of frames for generating all the Hilbert space $\mathcal{L}^2(\mathbb{R}^2)$ (see Theorem 2). So having a Gabor wavelet ξ, we can write any image $I \in \mathcal{L}^2(\mathbb{R}^2)$ as

$$I = \sum_{j \in \mathbb{Z}} \sum_{k \in \mathbb{Z}} c_{jk} \psi_{j,k}.$$

In difference with the equation (4.3), this time the scalars c_{jk} are not unique due to non-orthogonality of individual frame functions. Nevertheless, an image I can still be reconstructed just by knowing ξ and $c_{jk}, j, k \in \mathbb{Z}$.

The response of a Gabor wavelet to an image is obtained via 2-D convolution operation:

$$\int \int I(p,q)\, g_{\theta,\phi,\sigma^2}(x-p, y-q)\, dpdq. \tag{4.4}$$

(see Corollary 4.1). The expression (4.4) is a complex-valued number representing both the phase and the amplitude of a response of the image to the wavelet. A phase shows the direction and an amplitude shows the intensity of the response. An amplitude is sensitive to the extraneous factors of the image, such as contrast or illumination. Hence, only the phase information is taken into consideration, i.e. coordinates of the quadrant of the complex plane, in which the phase number lies, are saved as a bit tuple. These coordinates are obtained simply as signs of the real and imaginary parts of the 2-D integral (4.4).

This approach has a significant advantage in comparison to the binary representation of the quadrant. The phase code is cyclic now, i.e. there is always only a single bit change between two adjacent quadrants. Hence it is a 2-bit Gray code (see Definition 13).[39]

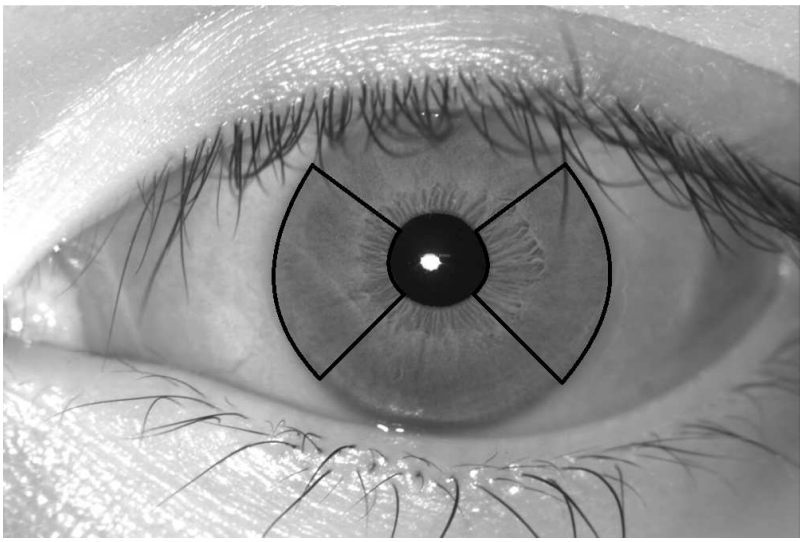


Figure 4.5: Segments of iris texture proposed by Kang et al. to be only used

This way we obtained 2 bits of information characteristic for frequency, orientation and deviation based on $(\theta, \phi, \alpha, \beta)$ at polar position (x, y).

For the same angular distance we can obtain more phase information given by convolution with another wavelets. Daugman suggests to use 8 self-similar wavelets for every sample angular distance.[7] Despite their partial overlap, correlation of each pair of them is usually only negligible.[13] Depending on the position, this wavelet family can differ in deviation and frequency. In Daugman's case, α and β parameters span identically from about 14% to 112% of the iris radii, while the wavelet frequency spans in their inverse proportion to maintain the self-similarity.[7]

However, standard implementations consider 8 same wavelets with the sample origins laying on the horizontal lines spanned linearly across the transformed picture,[13;23] i.e. the lines described by

$$y = \frac{1 + 2i}{16} h,$$

i = 0, ..., 7, where h represents the height of the normalized image.

This way we obtained 16 bits of information for each angular distance. However, not a uniform distribution of angular distances over all the circle is used. According to lower reliability of areas potentially occluded by eyelids or more sensitive to noise such as eyelashes and light reflection, the set of sample origins is restricted to 2 segments. While Daugman suggests these segments of interest to be symmetric over the horizontal meridian (we have already implemented this restriction in the pseudocode of integro-differential operator), Kang et al. propose to use only segments $\left[\frac{7\pi}{4}, \frac{\pi}{6}\right]$ and $\left[\frac{5\pi}{6}, \frac{5\pi}{4}\right]$ [25] (see Figure 4.5). Each origin then standardly lies on one of 128 angular distances uniformly distributed over these segments of the circle.

This way we obtained $2 \cdot 8 \cdot 128 = 2,048$ bits of information. This is the prevalent approach to feature extraction. However, other methods have also been proposed in the past. Predominant resultant data designated for comparison is a binary code, nevertheless, real-valued approaches are also suggested. While these

have an advantage of containing more information, they require significantly more storage space. They are often investigated to be used for a multibiometric recognition together with the face techniques. Some of these methods with references are collected in the Iris Biometrics book. [16]

In the feature extraction pseudocode, we stick to the prevalent method for obtaining information. By contrast to Daugman's method, we use always the same Gabor filter, responding to the areas uniformly distributed over specified segments of an iris.

For the simplicity of the code, we use a classic 2-D convolution of image and filter. However, in the case of implementation of the algorithm, wavelet separation (suitable for Gabor wavelets – see Theorem 6) would carry significantly faster computations, especially for large filters.

Algorithm: FEATURE EXTRACTION (1 of 2)

Input: Discrete image *image* of resolution $width \times height$ and Gabor wavelet determining factors θ, ϕ and σ^2.
Output: Discrete binary image $iris$ of resolution 256×8.

procedure CONVOLUTION($centerX, centerY$)
$sum \leftarrow 0$
for $x \leftarrow -size$ **to** $size$
do { **for** $y \leftarrow -size$ **to** $size$
do { **if** $(centerY + y \geq 0 \wedge centerY + y \leq height - 1)$
then { $sum \leftarrow sum + image[\mathrm{Mod}(centerX + x, width),$
$centerY + y] \cdot gabor[-x, -y]$
return (sum)

Algorithm: FEATURE EXTRACTION (2 of 2)

main

$size \leftarrow \text{Round}(\frac{8}{3}\sigma)^{[40]}$
for $x \leftarrow -size$ **to** $size$
do { **for** $y \leftarrow -size$ **to** $size$
  do $gabor[x, y] \leftarrow e^{-(x^2+y^2)/\sigma^2} e^{-2\pi\imath(\phi x+\theta y)}$
for $i \leftarrow 0$ **to** 127
do { **if** $i \leq 15$
  then $firstX \leftarrow 0$
  else { **if** $i \leq 79$
    then $firstX \leftarrow \text{Round}(\frac{1}{4}width)$
    else $firstX \leftarrow \text{Round}(\frac{1}{2}width)$
for $j \leftarrow 0$ **to** 7
do { $response \leftarrow \text{Convolution}(firstX + Round(\frac{i}{256}width),$
    $Round(\frac{2i+1}{16}height))$
  $iris[2i, j] \leftarrow \text{Sign}(\Re(response))$
  $iris[2i + 1, j] \leftarrow \text{Sign}(\Im(response))$
for $i \leftarrow 0$ **to** 255
do { **for** $j \leftarrow 0$ **to** 7
  do { **if** $iris[i, j] = -1$
    then $iris[i, j] \leftarrow 0$
return $(iris)$

Chapter 5

Iris Code Comparison

There are several ways of comparing the code created in the previous section with the database codes. The fact influencing the character the most is whether the data is real or binary. We will stick to the latter option as it prevails in use.

Another aspect is the length of the code. Most common is to use 2,048 bits of information saved in a 8×256 matrix which was patented as IrisCode^{TM}(see Figure 5.1).[13] However, both longer (4,096-bit) and shorter codes were proposed.[16]

There are two possible reasons for the process of comparison:

- *Identification* ($1 : n$ comparison) is a process of finding a database sample code which belongs to the same person as an investigated code. If there are more decission arguments available, then more output sample codes are better than none. In identification a high pressure is put on the speed of the process.

- *Authentication* ($1 : 1$ comparison) is a process of deciding whether an investigated iris code belongs to the same person as the identity (together with the database code) of the claimed person. The output is a decision whether an investigated person is the owner of the database iris code or he is an impostor. A truthfulness probability of the decission can be a part of the output. In authentication a high pressure is put on the reliability of the answer.

Let us define the process of iris code comparison mathematically now.

Definition 14. *Let $C \subseteq \{0,1\}^n, n \in \mathbb{N}$ be a set of all potential iris codes. Then* similarity function *(or* comparator*) is a function $\Gamma : C \times C \to \mathbb{R}$ returning a level of similarity between two investigated codes.*

Note. Standardly $\Gamma : C \times C \to [0,1]$.

Figure 5.1: Iris code consisting of 2,048 bits of information

Note. In the case of the iris code obtained in previous sections:

$$|C| = 2^{8\times 256} \cong 3.2 \cdot 10^{616}$$

Definition 15. *Let Γ be a similarity function, $c \in C$ an investigated code, $D \subseteq C$ a database of iris codes, $|D| = n$ and $d_1, \ldots, d_n \in D$. Let t be a threshold parameter.*

Then identification *is a function $F : C \times D \to \mathbb{N}_0$ defined in a following way:*

$$F(c,t) := \begin{cases} i & \text{if } i = \arg_j \max \{\Gamma(c, d_j)\}, j = 1, \ldots, n \wedge \Gamma(c, d_i) \geq t, \\ 0 & \text{otherwise.} \end{cases}$$

Note. A part of an output can be also $\Gamma(c, d_i)$.

Note. If we are interested only in finding the nearest-sample-code's owner (i.e. we know that investigated person has an image in the database), we define *identification* in a following way:

$$F(c) := \arg_j \max \{\Gamma(c, d_j)\}, \quad j = 1, \ldots, n.$$

Definition 16. *Let Γ be a similarity function, $c_i \in C$ an investigated code and $c_d \in C$ a database sample code. Let t be a threshold parameter.*

Then authentication *is a function $A : C \times C \to \{0, 1\}$ defined in a following way:*

$$A(c_i, c_d, t) := \begin{cases} 1 & \text{if } \Gamma(c_i, c_d) \geq t, \\ 0 & \text{otherwise.} \end{cases}$$

Note. A part of an output can be also $\Gamma(c_i, c_d)$.

The human eye finds it easy to recognize two similar – but not the same – iris images regardless of a their tiny tilt, different contrast, small difference of illumination level or partial occlusion.[24] However, it is not so obvious in the case of computer-driven programs. Each of these factors can lead to a strict rejection of 2 irides of the same owner.

Hence, our goal is to find a metric on the space of potential iris codes which mathematically represents our intuition or – better – a metric in which two irides of the same owner are as 'close' as possible while each different pair have sufficiently long distance between.

5.1 Hamming Distance

Definition 17. *Let $C \subseteq \{0, 1\}^n$ be a set of all potential iris codes. Then* Hamming distance *between $c_1, c_2 \in C$ is a number of coefficients $(i, j), i, j = 1, \ldots, n$ in which they differ.*

Proposition 7. *Hamming distance is a metric.*

Proof. Let $\mathrm{HD}(c_1, c_2)$ denote a Hamming distance of codes $c_1, c_2 \in C$.

(a) $\mathrm{HD}(c_1, c_2) \geq 0$ trivially. $\mathrm{HD}(c_1, c_2) = 0$ if and only if c_1 and c_2 agree in all coordinates and this happens if and only if $c_1 = c_2$.

(b) The number of coordinates in which c_1 differs from c_2 is equal to the number of coordinates in which c_2 differs from c_1.

(c) $\mathrm{HD}(c_1, c_2)$ is equal to the minimal number of coordinate changes necessary to get from c_1 to c_2 and $\mathrm{HD}(c_2, c_3)$ is the minimal number of coordinate changes necessary to get from c_2 to c_3. So $\mathrm{HD}(c_1, c_2) + \mathrm{HD}(c_2, c_3)$ changes turn c_1 into c_3. But

$$\mathrm{HD}(c_1, c_2) + \mathrm{HD}(c_2, c_3) \geq \mathrm{HD}(c_1, c_3),$$

because $\mathrm{HD}(c_1, c_3)$ is the minimal number of coordinate changes necessary to get from c_1 to c_3.

□

In the case of arbitrary binary codes c_1 and c_2, Hamming distance is nothing else than an L^1-norm of a vector $c_1 \oplus c_2$, where $\oplus$ denotes a boolean Exclusive-OR operator (XOR):

Definition 18. *Let $\mathcal{V}$ be a vector space over $\mathbb{C}$. A function $\|\cdot\| : \mathcal{V} \to \mathbb{R}$ is called a* norm *iff it satisfies following 3 properties for any elements $x, y \in \mathcal{V}$ and $t \in \mathbb{C}$:*

- $(\|x\| \geq 0) \wedge (\|x\| = 0 \Leftrightarrow x = 0)$,
- $\|t \cdot x\| = |t| \, \|x\|$,
- $\|x + y\| \leq \|x\| + \|y\|$.

Definition 19. *Let $c = (x_1, \ldots, x_n) \in \mathbb{C}^n$. Then an L^1-*norm *is a function $\|\cdot\|_1 : \mathbb{C}^n \to \mathbb{R}_0^+$ defined as*

$$\|c\|_1 := \sum_{i=1}^{n} |x_i|.$$

Proposition 8. *$\|\cdot\|_1$ is a norm.*

Proof. Let $c = (c_1, \ldots, c_n), d = (d_1, \ldots, d_n) \in \mathbb{C}^n$ and $t \in \mathbb{C}$. Then

- $(\|c\|_1 = \sum_{i=1}^{n} |c_i| \geq \sum_{i=1}^{n} 0 = 0) \wedge (\|c\|_1 = 0 \Longleftrightarrow c = \bar{0})$,
- $\|t \cdot c\|_1 = \sum_{i=1}^{n} |t \cdot c_i| = \sum_{i=1}^{n} |t| \cdot |c_i| = |t| \sum_{i=1}^{n} |c_i| = |t| \cdot \|c\|_1$,
- $\|c + d\|_1 = \sum_{i=1}^{n} |c_i + d_i| \leq \sum_{i=1}^{n} (|c_i| + |d_i|) = \sum_{i=1}^{n} |c_i| + \sum_{i=1}^{n} |d_i| = \|c\|_1 + \|d\|_1$.

□

Theorem 9. *Let $C \subseteq \{0,1\}^n, n \in \mathbb{N}$ be a set of all potential iris codes and $c_1, c_2 \in C$. Then*

$$HD(c_1, c_2) = \|c_1 \oplus c_2\|_1 . \tag{5.1}$$

Proof. Results directly from the discussion above.

□

Note. $\|\cdot\|$ will always denote $\|\cdot\|_1$ hereafter.

Hamming distance is a very natural metric for measuring the distances between iris codes. The distance 0 would represent a perfect match, while the distance n lies between two inverse codes. In order to achieve a normalized distance in an interval $[0, 1]$ we can rewrite (5.1) in a following way:

$$\mathrm{HD}_n(c_1, c_2) := \frac{\|c_1 \oplus c_2\|}{n}.$$

A serious disadvantage of a Hamming distance lies in its sensitivity to tiny head tilts. Only a fine misalignment can lead to a total misleading information as can be seen in the following example:

Example. Let $c_1 = (0, 1, 0, 1, \ldots, 0, 1)$ and $c_2 = (1, 0, 1, 0, \ldots, 1, 0)$. Then

$$\mathrm{HD}_n(c_1, c_2) = 1.$$

A significant advantage of the Gabor-wavelets-based feature extraction lies in its relation between the iris texture and the iris code. Since an iris code is composed of localized features, bit shifts in an iris code correspond to angular shifts of the underlying iris texture (except for the boundaries of used segments). Hence, a misalignment of the iris textures can be partially eliminated by computing a distance between a genuine iris code and subsequently circulary shifted investigated iris code. We will denote an iris code c shifted i positions to the left by $c <<< i$.

Since an eye was localized during the preprocessing phase and normalized to be 'upside up' there is no need to test all the circular shifts. Only a constant k is chosen, usually $k < 10$. There are 2 reasons not to shift the iris code all around.

The first one is the speed of the algorithm. A XOR operation on two binary iris codes is a single machine operation and a 2,4-GHz CPU can process almost 1,000,000 comparisons per second.[7] However, there is a higher-order digit of codes involved in identification in projects such as Aadhaar.[16]

The second reason is the discriminalibity of the results. We always store only the best match of the codes, hence, the mean distance between distinct codes decreases with every increment of the shift level. We expect no more than a tiny tilt of an eye. Therefore, there is no need to shift the code all around. John Daugman reports a distance mean between different codes to decrease from 0.5 to 0.458 on the sample of 9.1 million comparison of 4,258 different eyes when allowing 7 circular shifts.[7] Meanwhile, the mean distance between two genuine samples lowers to 0.11.

Hence, we can further upgrade the distance formulated in (5.1):

$$\mathrm{HD}_{ns}(c_1, c_2) := \min_{i=-k,\ldots,k} \frac{\|c_1 \oplus (c_2 <<< i)\|}{n}.$$

Another improvement is a use of the noise mask generated during the image preprocessing phase. Let m_1 denote a noise mask code for the iris code c_1 and m_2 that one for c_2. Let $c_1, c_2 \in C \subseteq \{0, 1\}^{n \times m}$. Then $m_1, m_2 \in C$. Noise mask codes consist of 1s at positions determined to be compared and 0s on those considered to be occluded.

Hence we finalize the formula (5.1) subsequently:

$$\mathrm{HD}_{nsm}(c_1, c_2) := \min_{i=-k,\ldots,k} \frac{\|c_1 \oplus (c_2 <<< i) \cap m_1 \cap (m_2 <<< i)\|}{\|m_1 \cap (m_2 <<< i)\|}.$$

Note. If no confusion arises we will use notation HD for HD_{nsm} hereafter.

Proposition 10. *The time complexity for Hamming-distance-based comparison is $O(k \cdot n)$, where n is the length of the code and k is the maximal number of shifts.*

The space complexity is $O(k + n)$.

Proof. In order to obtain a Hamming distance we need to execute a single XOR operation on each of n positions of the codes. Hence it is operated in $O(n)$. The time complexity of counting the bits with a value 1 is operated also in $O(n)$. We need to repeat this process $(2k + 1)$-times – once for every shift of an investigated code. Hence,

$$C_t = (2k + 1) \cdot O(n) \cdot O(n) = O(m \cdot n).$$

From the storage point of view we need to save both the codes $(2 \cdot n)$, the dissimilarity vector (n) and the current and the smallest distance between the 2 codes $(O(1))$. Hence,

$$C_s = 2 \cdot n + n + O(1) = O(n).$$

□

Hamming-distance-based comparison in the last stated version is partially circular-misalignment-proof. However, it is still sensitive to segmentation inaccuracies and non-linear distortions. It can either shift all the code linearly or leave it as is. The solution for non-linear distortions provides *Levenshtein distance*.

5.2 Levenshtein Distance

Levenshtein distance is a metric originally proposed by Vladimir Levenshtein in 1965,[41] nowadays used predominantly for correction of mistyping. It is an extension of Hamming distance operating also on the codes of different lengths, allowing 3 operations – beside a substitution also an insertion and a deletion:

Definition 20. *Let $C \subseteq \{0,1\}^*$ be a set of all potential iris codes and $c_1 \in \{0,1\}^m, c_2 \in \{0,1\}^n, c_1, c_2 \in C, m, n \in \mathbb{N}$ are 2 binary vectors of iris code, not essentially of the same length. Let*

- *p_s be a cost function of a substitution,*
- *p_i a cost of an insertion,*
- *p_d a cost of a deletion*

of single bit, $p_s, p_i, p_d \in \mathbb{R}^+$.

Then the Levenshtein distance *is a function* $LD : C \times C \to \mathbb{R}_0^+$ *defined as*

$$LD(c_1, c_2) := ld_{c_1,c_2}(|c_1|, |c_2|),$$

where $ld_{c_1,c_2} : \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{R}_0^+$ *is a function given by*

$$ld_{c_1,c_2}(i,j) := \begin{cases} i \cdot p_i & \text{if } j = 0, \\ j \cdot p_d & \text{if } i = 0, \\ \min \begin{cases} ld_{c_1,c_2}(i-1,j) + p_i, & \text{(i)} \\ ld_{c_1,c_2}(i,j-1) + p_d, & \text{(d)} \\ ld_{c_1,c_2}(i-1,j-1) + p_s\left(c_1^i, c_2^j\right) & \text{(s)} \end{cases} & \text{otherwise.} \end{cases}$$

The definition by itself looks a bit incomprehensible, let me show that it is nothing difficult though. We can imagine the entire process as a filling in of a $(m+1) \times (n+1)$ matrix. Each position (i, j) denotes the lowest cost of transferring first i bits of a vector c_1 into the first j bits of a vector c_2.

First, we insert 0 to a position $(0, 0)$ – changing any code into the same one is for free. The cost on all positions $(0, j)$ is $j \cdot p_d$ then – we need to pay j-times the cost of a deletion to transfer j-dimensional vector into an empty one. Symetrically with the first column and insertion of the bits.

The other elements of the matrix are filled in the following way. Each element (i, j) will obtain the cheapest price of three posibilities:

- The cost of transferring the first j bits of the vector c_2 into first $i-1$ bits of vector c_1 plus one cost of insertion (in this case we are appending the bit c_1^i) – possibility (i).
- The cost of transferring the first $j-1$ bits of the vector c_2 into first i bits of vector c_1 plus one cost of deletion (in this case we are deleting the bit c_2^j) – possibility (d).
- The cost of transferring the first $j-1$ bits of the vector c_2 into first $i-1$ bits of vector c_1 plus the cost of substitution of the bit c_2^j to the bit c_1^i – possibility (s). The cost function p_s has a zero value if $c_2^j = c_1^i$ and a strictly positive value otherwise.

The value on the position (m, n) is then the minimal cost we need to pay to transfer a vector c_2 into a vector c_1. In the case of two codes of the same length $(m = n)$ and in the case of the same cost of both insertion and deletion $(p_i = p_d)$, it is also a minimal cost of transferring a vector c_1 into a vector c_2.

Proposition 11. *Levenshtein distance is a metric on the codes of the same length.*

Proof. (a) Every edit's cost is a positive number; hence, $\mathrm{LD}(c_1, c_2) \geq 0$ immediately. Since every cost is strictly positive the only situation when $\mathrm{LD}(c_1, c_2) = 0$ occures when there is no edit required. No edit to transfer a code c_1 into c_2 is needed if and only if $c_1 = c_2$. Hence,

$$\mathrm{LD}(c_1, c_2) = 0 \Leftrightarrow c_1 = c_2.$$

(b) Let $\mathrm{LD}(c_1, c_2) = k$ and x_k be a sequence of edits that yields to a change of c_1 into c_2 with a cost of k. Then x_k contains the same number of insertions and deletions since we consider only codes of the same length.

Let x_k^{-1} be a sequence of edits, reverse to x_k, i.e. when insertion is used in x_k, deletion is integrated to x_k^{-1} instead. Substitution is self-inverse as an operation. Hence, x_k^{-1} is a valid path from c_2 to c_1 with the same cost as x_k. Therefore $\mathrm{LD}(c_2, c_1) \leq \mathrm{LD}(c_1, c_2)$.

Let us consider $\mathrm{LD}(c_2, c_1) = l < k$ now. Let x_l be a path from c_2 to c_1 with a cost of l. Then x_l^{-1} is a valid path from c_1 to c_2 of the same cost. But such a sequence of edits cannot exist because Levenshtein distance is the smallest cost of a transformation of c_1 into c_2 and $\mathrm{LD}(c_1, c_2) = k > l$. This is a contradiction and such a path x_l does not exist.

Hence,

$$\mathrm{LD}(c_1, c_2) = \mathrm{LD}(c_2, c_1).$$

(c) Let $\mathrm{LD}(c_1, c_2) = k, \mathrm{LD}(c_2, c_3) = l$ and x_k, x_l be sequences of edits that yield to a change of c_1 into c_2, c_2 into c_3 respectively, with costs of k and l. Let $||$ denote an operation of concatenation. Then $x_k||x_l$ is a valid sequence of edits transforming c_1 to c_3. Levenshtein distance is defined as shortest such a path. Hence,

$$\mathrm{LD}(c_1, c_3) \leq k + l = \mathrm{LD}(c_1, c_2) + \mathrm{LD}(c_2, c_3).$$

□

Now we will show that Levenshtein distance is in fact an extension of a Hamming distance:

Theorem 12. *Let $C \subseteq \{0,1\}^n, n \in \mathbb{N}$ be a set of iris codes of the same length n. Let $p_i = p_d = n + 1$ and $p_s = 1$. Then*

$$\forall c_1, c_2 \in C \qquad LD(c_1, c_2) = HD(c_1, c_2).$$

Proof. All codes in C have the same length. Therefore, Hamming distance is a valid metric on C. The cost of substitution is the same as in Hamming distance while the cost of insertion and deletion is higher then the code length. The cost on the diagonal of the Levenshtein matrix will not be influenced by insertions or deletions now, because it would be cheaper to replace all the vector by the other one than to make a single insertion or deletion. Hence, on the position (n, n) is the number of bits we need to substitute. This is the number of positions in which the codes c_1 and c_2 differ. And this is the Hamming distance.

□

The logical value of the costs would be $p_i = p_d = p_s(x, y) = 1, x \neq y$. However, these values can be chosen arbitrarily depending on the preferences – whether to non-linearly shift or to prefer the substitutions.

We have proved that Levenshtein distance is a stronger tool for aligning the iris codes than the Hamming distance. It means that by using Levenshtein distance, both the codes fit better onto each other. At the same time, we must not forget that it also lowers the distance between two codes extracted from different irides.

Hence, Levenshtein distance can be a useful tool for non-linear distortions such as an off-gaze; for a linear shift Hamming distance still gives better results.

Proposition 13. *The time complexity for Levenshtein-distance-based comparison is $O(n^2)$, where n is the length of the code.*

The space complexity is also $O(n^2)$.

Proof. In order to compute a Levenshtein distance all we need to do is to create an $(n+1) \times (n+1)$ matrix. For each of the coordinate pair we need to compute the cost of insertion, deletion and substitution of a single bit, or one of them in the case of a first row and column. Even in the worst case, the algorithm operates in an $O(1)$ complexity. Hence,

$$C_t = (n+1) \cdot (n+1) \cdot O(1) = O\left(n^2\right).$$

From the space complexity point of view we need to store both the codes $(2 \cdot n)$, the costs of the operations $(O(1))$ and the Levenshtein matrix $((n+1) \times (n+1))$. Hence,

$$C_s = 2 \cdot n + O(1) + (n+1) \cdot (n+1) = O(n) + O(1) + O\left(n^2\right) = O\left(n^2\right).$$

□

The $O(n^2)$ would not be sufficient for competitive identification systems. However, the time complexity can be lowered down to $O(m \cdot n), m \in \mathbb{N}$ with the maintenance of the same results in most cases.

Similarly as with the Hamming distance, we assume the iris not to be rotated more than some angle, say $k_a << \pi$. Consequently, we do not have to compute the rotations over a higher angle. It means we do not have to consider the values of the matrix in a distance from the diagonal higher than a constant k,

$$k := \frac{k_a n}{2\pi}.$$

Let L denote the Levenshtein matrix for the codes c_1 and c_2. We simply compute only such coordinates (x, y) of L for which it holds $|x - y| \leq k$. We define all the other values as a number higher than the biggest distance between any two codes can be – say ∞.

We can define such a matrix L_C in a mathematical way:

$$L_C(i,j) := \begin{cases} L(i,j) & \text{if } |x-y| \leq k \\ \infty & \text{otherwise.} \end{cases}$$

This matrix still has an $O(n^2)$ space-complexity, though it can also be lowered by not saving the values distant more than $k+1$ from the diagonal or integrating this constraint into the matrix definition. However, the biggest advantage lies in the lowered time complexity.

Proposition 14. *Above-mentioned algorithm has a time complexity $O(k \cdot n)$, where n is the length of the code and k the maximal number of shifts.*

Proof. In comparison with computing a full Levenshtein matrix, this time we only need the values on the diagonal and adjacent k lines. For each of them the complexity is $O(1)$. Hence, the overall complexity is

$$\begin{aligned} C_t &= (n+1) + 2n + 2(n-1) + \cdots + 2(n-k) \\ &= (n+1) + 2\left[\frac{n(n+1)}{2} - \frac{(n-k-1)(n-k-2)}{2}\right] \\ &= (n+1) + \left(n^2 - n\right) - \left(n^2 - kn - 2n - kn + k^2 + 2k - n + k + 2\right) \\ &= 2kn + 3n - k^2 - 3k - 1 = O(k \cdot n) \end{aligned}$$

□

Part II

Iris Cryptosystems

There is one significant difference between iris cryptosystems and standard iris recognition systems. While an output of the latter one consists of *Yes* and *No* in the case of authentication (or a database sample in the case of identification) biometric cryptosystems retrieve biometric keys and these can be either correct or incorrect.

The FRR of a biometric cryptosystem therefore represents the rate of incorrect keys returned to genuine users. Analogically, the FAR corresponds to the rate of correct keys untruly generated for the impostors.

Every security approach has its drawbacks. Passwords might be guessed, tokens reverse-engineered and biometrics might be compromised. However, their joint use provide the best security presently available.

In comparison with traditional password schemes, biometric cryptosystems bring remarkable security benefits since it is significantly more difficult to forge, copy, share and distribute them.[42] Moreover, iris biometrics provide an equal level of security for every sample since physiological characteristics are not user selective.

On the other hand, iris patterns are inherent and stable life-long. Once the iris is compromised a system has to be replaced since it is difficult to change the biometric characteristics. Users are therefore reluctant to have raw biometric data stored in databases. Hence, the task is to find a way to store the samples *irreversibly* (it should be computationally hard to reconstruct the original biometric sample from the stored reference data) and *unlinkably* (one reference data should be hard to guess by knowing another one).[16]

Moreover, users wish to have different keys for different purposes: two separate keys for two distinct bank accounts makes it possible to revoke one without affecting the other.

Hence, our goal is to design a system which does not store biometric samples, but only a reference data from which the biometric characteristics cannot be derived and from which the key is not revealed unless the iris is physically present.

Iris cryptosystems generally exhibit noticeably inferior performance in a comparison with standard iris recognition systems. This is because the enrolled template is not seen and, therefore, cannot be properly aligned at the comparison phase.

However, cryptography requires the keys to be exactly right or the protocol will fail. This is an argument for the use of error-correcting codes.

Chapter 6

Error-Correcting Codes

Error correction is a technique that enables restoration of the data delivered over unreliable communication channels. Any data misrepresented by the channel can be – within certain limitations – detected and corrected based on the remaining data.[43;44]

Any error-correcting code is characterised by a triplet (n, k, d), where

- n is a length of codewords,
- $k := \log_q |C|$, where $|C|$ is a cardinality of the code and q is a cardinality of the field over which is the code defined,
- d is a minimal distance between any 2 codewords.

6.1 Hadamard Codes

Definition 21. *A square matrix is called to be* orthogonal *iff inner product of any two distinct rows or columns is equal to* 0.

Definition 22. *A* Hadamard matrix *of order n is a square orthogonal matrix H of dimensions $n \times n$ with elements* 1 *and* -1.

Corollary. For every Hadamard matrix of order $n > 1$, n is an even number.

Note. For every Hadamard matrix, every two distinct rows or columns agree in precisely $n/2$ components.

Proposition 15. *Let H be a Hadamard matrix of order $n > 2$. Then n is divisible by* 4.

Proof. Let us consider two distinct rows $\mathcal{R}_a, \mathcal{R}_b$ in H, $1 \leq a, b \leq n$, both of them containing at least one element -1.

Let A is a set of column indexes in which the row $\mathcal{R}_a$ contains a value 1 and symetrically for a set B and the row $\mathcal{R}_b$. Then

$$|A| = \frac{n}{2} = |B| .$$

Rows $\mathcal{R}_a$ and $\mathcal{R}_b$ differ in $|(A \setminus B) \cup (B \setminus A)| = \frac{n}{2}$ elements. Simultaneously,

$$|(A \setminus B) \cup (B \setminus A)| = |A| + |B| - 2\,|A \cap B| = \frac{n}{2} + \frac{n}{2} - 2\,|A \cap B| .$$

It means that

$$\frac{n}{2} = n - 2\left|A \cap B\right| \implies n = 4\left|A \cap B\right|.$$

hence, n is divisible by 4.

□

Theorem 16. *Let H be a Hadamard matrix of order n. Then*

$$\begin{pmatrix} H & H \\ H & -H \end{pmatrix}$$

is a Hadamard matrix of order $2n$.

Proof. Obvious.

□

Definition 23. *Let $H_0 := \begin{pmatrix} 1 \end{pmatrix}$ and*

$$H_{k+1} := \begin{pmatrix} H_k & H_k \\ H_k & -H_k \end{pmatrix},$$

$k \in \mathbb{N}$.

Then H_n is a Hadamard matrix of order 2^n constructed by a Sylvester method.

Note. Let $H = (h_{ij})_{0 \le i,j \le 2^n - 1}, h_{ij} \in \{-1, 1\}$ be a Hadamard matrix of order 2^n constructed by Sylvester method. Let $i = \sum_{k=0}^{n-1} c_k 2^k$ and $j = \sum_{l=0}^{n-1} c_l 2^l$ be binary representations of i and j.

Then

$$h_{ij} = (-1)^{\sum_{k=0}^{n-1} c_k d_k}.$$

Definition 24. *Let H be a Hadamard matrix of order $n = 2^k, k \in \mathbb{N}$ and*

$$C_H := \begin{pmatrix} H \\ -H \end{pmatrix}.$$

Then the rows of C_H with replaced -1s by 0s are codewords and form a Hadamard code *C.*

Theorem 17. *Hadamard code of the length $n = 2^m, m \in \mathbb{N}$ has a minimal distance between the codewords $d = n/2$ and $k = m + 1$.*

Proof. (a) We will stick to the version described by Peterson and Weldon.[44]

Let $c_1, \ldots, c_n, c_{-1}, \ldots, c_{-n}$ be all the codewords generated by a Hadamard matrix. Then all the corresponding components of c_i and c_{-i}, $i \in \{1, \ldots, n\}$ are different and the distance between c_i and c_{-i} is n. Since $\pm v_i$ and $\pm v_j$ are orthogonal for all $i, j \in \{1, \ldots, n\}, i \neq j$ they match exactly in half the positions and differ in the other half. Thus, the corresponding binary vectors are at a distance $d = n/2 = 2^{m-1}$.

(b) Hadamard code is binary and its cardinality is $2n = 2^{m+1}$. Then it is straightforward that $k = \log_2(2^{m+1}) = m + 1$.

□

6.2 Reed-Solomon Codes

Reed-Solomon codes were described for the first time in 1960 by Irving Reed and Gus Solomon.[45]

Definition 25. *Let $q = p^k, k \in \mathbb{N}$ and p be a prime number. Let $\mathcal{F}_q$ be a finite field, $\zeta_1, \dots \zeta_{q-1}$ all non-zero elements of $\mathcal{F}_q$ in a fixed order and Ξ a set of all polynomials over $\mathcal{F}_q$ of degree $\leq k-1$.*

Then Reed-Solomon code *is defined as follows:*

$$\mathcal{RS}_{q,k} := \{(f(\zeta_1), \dots, f(\zeta_{q-1})), f \in \Xi\}.$$

Definition 26. *Error-correcting code C of length n is said to be* linear *over a finite field $\mathcal{F}_q$ iff C is an arbitrary subspace of $\mathcal{F}_q^n$.*

Proposition 18. *Reed-Solomon codes are linear.*

Proof. Let us denote

$$[f] := (f(\zeta_1), \dots, f(\zeta_{q-1}))$$

for any polynomial $f \in \Xi$ and all non-zero elements $\zeta_1, \dots \zeta_{q-1} \in \mathcal{F}_q$. Let $f, g \in \Xi$ and $\zeta \in \mathcal{F}_q$. Then

- $[f+g] = [f] + [g]$,
- $\zeta[f] = [\zeta f]$.

□

Theorem 19 (Singleton bound)**.** *Let C be a code characterised by a triplet (n, k, d). Then*

$$n \geq k + d - 1.$$

Proof. In other words, we want to prove that $d \leq n - (k-1)$. Let us project all the codewords on the first $(k-1)$ coordinates now. Since there are q^k different codewords at least two of them should agree on these $(k-1)$ coordinates. But these then disagree on at most the remaining $n - (k-1)$ ones. Hence, $d \leq n - (k-1)$.

□

Definition 27. *Codes, which match the Singleton bound, are called to be* maximum distance separable *(MDS).*

Theorem 20. *Reed-Solomon codes are MDS.*

Proof. The proof is based on the simple fact that a non-zero polynomial of degree l can have at most l zeroes.[46] For Reed-Solomon code two codewords (with corresponding polynomials f and g) agree at i-th coordinate if and only if $(f-g)(\zeta_i) = 0, \zeta_i \in \mathcal{F}_q$. But $(f-g)$, as mentioned, can have at most $(k-1)$ zeros which means that $d \geq n - (k-1)$ and Reed-Solomon codes match the Singleton bound.

□

Corollary. Characterization of Reed-Solomon codes can be also written as

$$(n, k, n - (k-1)).$$

Chapter 7

Fuzzy Commitment Scheme

A fuzzy commitment scheme (which will be denoted as FCS hereafter) represents a method for combining iris biometry with both the token-based and the password-based cryptography. It was described for the first time in the technical report of Computer Laboratory of University of Cambridge by Hao, Anderson & Daugman in 2005.[42] It is based upon a 2,048-bit iris code described in the previous part.

Cryptography requires precise alignment of both the database and the investigated iris codes. However, there are standard differences between the codes of the same eyes reported to be 10 – 20%.[9] On the other hand, the difference between the codes of different eyes are between 40% and 60%. Consequently, we want to integrate error-correcting codes able to fix 1/5 of the code, but simultaneously unable to correct more than 2/5.

According to a robust statistical test of differences between various captures of the same iris,[42] there are two types of errors:

- *Burst errors* concentrated to particular areas caused by undetected eyelashes and specular reflections.
- *Background errors* spread all over the image caused by sensor, noise and iris skew.

Burst errors are local distortions which significantly affect one part of the image while the others are left more or less without a change. These can therefore be corrected by using Reed-Solomon codes which have already found wide application in CD or mp3 players etc.[47]

On the other hand, the rate of background errors is more or less constant across all parts of the image. This time, the good properties of Hadamard codes can be used.

The FCS proceeds in the following way: At the beginning, we have an iris code θ_i and a key K of constant length which we will compute later.

First, we divide K into blocks of fixed length m, each of the block representing one symbol of the alphabet. Then we encode the blocks using Reed-Solomon code. We receive a code word and this time we look at the blocks as words and we encode each of them separately using Hadamard code. The output is such a codeword θ_K of length 2,048,

$$\theta_K = \mathcal{H}\left(\mathcal{RS}\left(K\right)\right),$$

which looks like a genuine iris code. This code is XORed with the user's reference iris code θ_i obtained at enrollment, and a cipher code is received:

$$\theta := \theta_K \oplus \theta_i.$$

This code has a characteristic of an iris code. θ is subsequently saved to a token together with a hash value of the key $\#(K)$. The key K is then securely erased.

Hence, the encoding function looks the following way:

$$\mathcal{E} : \{K, \theta_i\} \to \{\#(K), \theta\}.$$

The decoding process starts with capturing the user's eye and extracting its feature code $\widetilde{\theta}_i$. Then this code is XORed with the cipher code θ:

$$\widetilde{\theta}_K := \theta \oplus \widetilde{\theta}_i = \theta_K \oplus \theta_e,$$

where θ_e is an error vector which we do not know at the moment. However, the closer the presented iris code is to the genuine one, the smaller the Hamming weight of the vector.

Definition 28. *Let c be a vector of length n and $\bar{0}$ denotes a vector of n zeros. Then Hamming weight HW of vector c is defined by*

$$HW(c) := HD(c, \bar{0}).$$

Now, if the Hamming weight of θ_e is greater than a threshold value given by error-correcting capabilities of Hadamard code, Reed-Solomon code respectively, then $\mathrm{HD}\left(\theta_K, \widetilde{\theta}_K\right)$ will be too great and the key will not be extracted. The other way around, if the presented code is within the distance possible to correct then the user will obtain the genuine key, i.e. if

$$\#\left(\mathcal{RS}^{-1}\left(\mathcal{H}^{-1}\left(\widetilde{\theta}_K\right)\right)\right) = \#(K),$$

then we obtained the right key and if not our access is denied.

Let us compute the size and the number of blocks before and after each encoding now:

We want the resultant vector to be 2,048-bit long. We would also like the Hadamard code to be able to correct around 25% of erroneous bits in each of the blocks.

By the Hadamard code properties the code consists of $2n$ codewords of length n. Considering Sylvester construction, let $n = 2^m, m \in \mathbb{N}$. Then there are 2^{m+1} codewords; each of them can have a unique binary representation of length $\log_2(2^{m+1}) = m + 1$. On the other hand, Hadamard code is capable of correcting $2^{m-2} - 1$ errors. Hence, $m = 6$ is a suitable choice (a greater value of m would result in a smaller total key length, while smaller value would lead to an unsufficient tolerance of background errors).[42] Then there are $2,048/2^m = 32$ blocks, each of them of the length $m + 1 = 7$ before and $2^m = 64$ after encoding. Hadamard code then corrects up to

$$\frac{2^{m-2} - 1}{2^m} = \frac{15}{64} \approx 23.4\%$$

erroneous bits.

Now most of the bits are corrected. However, if burst errors were present then a few blocks were distorted to such an extent that they were corrected into a different codeword. For this case, Reed-Solomon codes – by the Berlekamp-Massey algorithm, can correct up to 6 blocks, each of them consisting of $m+1 = 7$ bits.[42] At the beginning of the encoding process there are thus 20 blocks. Hence, the length of the secret key can be up to

$$|K| = 7 \cdot 20 = 140$$

bits. This is a length, which is sufficient for such encoding algorithms, as AES is.[48] It is also a way longer key than the longest one – 69 bits – achieved by fingerprint cryptography.[49]

For further improvement of the key security a 3-factor scheme can be implemented which initiates password protection into the process. In this scheme the password can be either used for a simple cipher code encryption or in an even safer version, the codewords generated by a Hadamard matrix can be permuted accordingly to the password.

Chapter 8

Cancelable Iris Biometrics

The biometric data of each human is given and cannot be changed. Their invariance over time – a property that makes biometrics so attractive – is also one of its major liabilities. When a password is compromised the user can simply choose a different one. When the biometric data is compromised replacement is impossible. For minimizing the impact of this weakness cancelable biometrics is introduced.

Cancelable iris biometrics consists of an intentional, repeatable distortion of a biometric signal based on a chosen transform and comparison of templates in the transformed domain.[50] The biometric signal is distorted exactly the same way at each enrollment, both in the registration and the authentication phase. The distortion transform is always selected to be non-invertible so even if both the transform function and the resulting data are known the original undistorted iris biometrics cannot be recovered.

Furthermore, if one transformed data is compromised the transform function is simply changed to re-enroll the user as a new person. These functions are chosen not only non-invertible to maintain irreversibility, but they must also reveal as little information about the original biometrics as possible in the name of unlinkability. Hence, a goal of cancellable iris biometrics is to find transforms which provide secure biometric templates while the comparison procedures in the transformed domain maintain recognition rates of the original biometric algorithm.[16]

8.1 Block Re-Mapping

The first technique to achieve the goal – block re-mapping – consists of four parts. In the first one, a normalized iris texture is partitioned into blocks of a fixed size. Subsequently, the blocks are permuted according to a fixed key. It is a good idea not to use permutations similar to identity, as their original meaning is eliminated then. However, until now all the steps were invertible operations. Anyone in possession of the permutation key can reverse this process without difficulties.

This is why the third step is introduced. That one takes some of the blocks and duplicate them over other blocks excluding them from the iris texture. The number of blocks remaining in the texture seriously influences subsequent matching process. If they are too few the recognition process will probably fail, however, if we do not overwrite enough blocks the modified texture will still reveal too much

information for the potential attacker. Simultaneously, it is a good idea not to use blocks potentially concealed by noise, such as eyelids or eyelashes and ignore them as a source for the re-mapping process. This provides the feature extraction phase a reasonable amount of iris texture instead of duplicating eyelashes all over the place.[16]

In the last phase, block edges are finely blurred to smooth the sharp edges on their join.

8.2 Mesh Warping

The second technique is called mesh warping (MW). In this approach we, in a way, lay a regular grid mesh over the iris texture and connect the vertices with the corresponding points on the iris texture. By distorting the grid according to a fixed key we are displacing the underlaying connected points and appropriately also the points inside the grid cells.

This distortion is also the reason why MW is an irreversible process, as the original data cannot be exactly recovered even if the warping key is known. However, the parameters of the key should not be chosen absolutely arbitrarily. First, if they are too similar to an identity mapping the resulting iris texture will be too similar to the original one. On the other hand, if large areas of the source texture are compressed to a few pixels there may not remain enough information for feature extraction.

Let us consider the following situation: We have a 512×64 pixel iris texture and a regular grid of 16×16 pixels. Then there are $31 \cdot 3 = 93$ vertices inside the texture. If each of them can be moved by 4 pixels to each horizontal direction and 4 pixels vertically, then there are $9^2 = 81$ replacement possibilities for each point. Thus, there are overall $81^{93} \approx 3 \cdot 10^{177}$ different transformations to choose as a key. Though some of them are similar to each other and some of them useless for the transformation, there is still enough amount of distinct transformation keys available.

Chapter 9

Potential Attacks

There are many types of attacks on iris recognition systems and cryptosystems varying in the phase in which they are exerted and a component they invade.[50] (see Figure 9.1). We will consider these types which aim particularly at influencing the final response.[16]

9.1 Attacks on Physical Biometric Data

Attacks of this type are specialized for gaining access to the system or to retrieve the key by presenting incorrect biometric data.

9.1.1 Spoofing

Spoofing is a general name for methods when an impostor attempts to fool the system and impersonate himself as someone else by intentional exposing of fake biometric data. These attacks are of 4 types:[51]

- *Artificial eye* – an impostor presents a printed iris, model of an eye or a video capture of the genuine iris
- *Printed contact lences* in impostor's eyes
- *Genuine eye*, removed from the body
- *Forced use* of genuine user

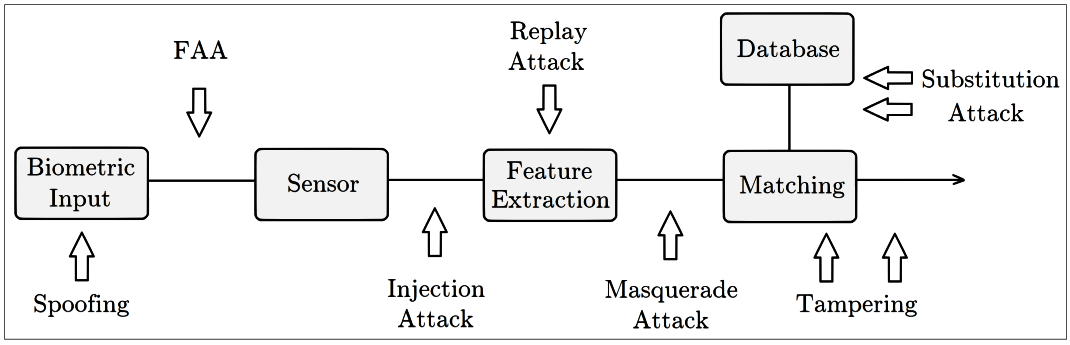


Figure 9.1: List of attacks with the phases in which they are exerted

A biometric system by itself is not protected against the last type of attack. However, no one probably expect it to be and this vulnerability can be easily eliminated by supervisioning of the image acquisition process. This can also easily reveal the first and the third type of attack. There are also other means of liveness detection.

The first one is a retinal light reflection commonly known as 'red-eye effect'. Once an eye is illuminated the light entering the eye via pupil is reflected back to the light source by the retina. In the case of an artificial eye the pupil stays black, while for the genuine one, it appears red due to blood vessels behind the retina.[51]

Light reflex is another mean of liveness detection. A live eye reacts to a change of illumination by shrinking or stretching the pupil. If the illumination is too strong a blinking also appears. The illumination reflects in the pupil of a live eye. On the other hand, in case of lences, reflection appears also on them.[51]

Last, but not least, an eye is persistently moving. This delicate unrest, called saccade, is another proof of liveness of the eye.[52] The same situation is also with the consistent pupillary unrest. This movement, hardly visible without a magnifying lense, is called hippus.[51;52]

9.1.2 False Acceptance Attack

False acceptance attack (FAA) is a special type of brute force attack for biometrics. It uses the fact that EER can only approach 0 value, but never meet it, and with this consideration has also been chosen FAR value of the system. FAA consists of presenting a sufficiently large number of different iris samples to the biometric sensor until a false accept, which the probability is always greater than zero, is realized.

To defend the system against this type of attack FAR value needs to be chosen as low as possible. On the other hand, by decreasing FAR, FRR is increasing. Creating a system with both minimal values is a general goal of iris biometry.

However, iris cryptosystems and cancelable iris biometrics are more vulnerable to FAA due to their lower performance rates and consequent higher FAR.

9.2 Attacks on Digital Biometric Data

In comparison with the previous group of attacks, digital-data-based attacks require infiltrating a biometric system at some point. This breaching is then exploited for the presentation of such data which lead to either acceptance of incorrect or rejection of genuine biometric data.

A defence against all types of digital attacks generally lies in making the system resilient to intrusion by the appropriate methods of computer security. Encryption techniques and authentication of individual devices in the biometric system are among the most useful.

9.2.1 Replay Attack

In the replay attack, an invader penetrates into the system and records genuine data leading to a successful matching. These data are then replayed either in the

same or in a slightly modified shape.

Replay attacks which do not modify the template can be detected by saving the history of acquired biometric data. By the probabilistic theory it is nearly impossible to capture two absolutely identical images within a few recordings.

9.2.2 Masquerade Attack

In the masquerade attack, an invader does not have full information about the genuine data. However, eavesdropping helps him to construct fake biometric data – either in the physical or digital form – which produce a sufficiently high match score.

However, this does not hold for iris cryptosystems and cancelable iris biometrics since their irreversibility prevents the invader from obtaining any useful information for reconstructing an original biometric input.

9.2.3 Injection Attack

In the injection attack, data acquired by a different biometric sensor are injected into the data transmission between a genuine sensor and feature extractor predending to be captured by the correct biometric sensor.

9.3 Substitution Attack

Substitution attack is a kind of attack in which a biometric sample, already registered in the system, is replaced by an invader either in the database or on the way between matcher and database. In the case of substitution attack, a genuine template is rejected and the invader is able to access the system by presenting his own biometric data.

However, it is significantly more difficult to execute such an attack in the case of iris cryptosystems since the biometric data is only used to be XORed with a cryptographic key and then securely erased. Substitution attack therefore requires additional knowledge.

A similar situation occurs with a substitution attack in the cancelable iris biometrics scheme. In this case, the transform key has to be known for the attack to be effective.

9.4 Tampering

Tampering is a type of attack where the invader gains access into the system and either overwrites the algorithm to give different results or simply replaces the final response.

However, this type of attack is hardly feasible within biometric cryptosystems since these return keys instead of binary decisions.[16]

References

[1] The Washington Post. National Geographic: Afghan Girl, a Life Revealed. 10.4. 2001.

[2] D. Denker. Along Afghanistan's War-Torn Frontier. *National Geographic*, 167(6):772–797, 6 1985.

[3] J. Daugman. How the Afghan Girl Was Identified by Her Iris Patterns. http://www.cl.cam.ac.uk/∼jgd1000/afghan.html. Retrieved: July 2013.

[4] C. Newman. A Life Revealed. *National Geographic*, 4 2002.

[5] R. Das. An Introduction to Biometrics. *Keesing Journal of Documents & Identity*, 17:3–5, 2006.

[6] L. Borovanský a kol. *Soustavná anatomie člověka*. Státní zdravotnické nakladatelství, Prague, 1967.

[7] J. Daugman. How Iris Recognition Works. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(1):21–30, 2004.

[8] C. C. Junquieira. *Základy histologie*. Nakladatelství a vydavatelství HH, Jinočany, 1999.

[9] J. Daugman. The Importance of Being Random: Statistical Principles. *Pattern Recognition*, 36(2):279–291, 2003.

[10] S. Rosypal a kol. *Nový přehled biologie*. Scientia, Prague, 2003.

[11] S. K. Modi. *Biometrics in Identity Management: Concepts to Applications*. Artech House, 2011.

[12] L. Flom and A. Safir. Iris Recognition System. Patent US 4641349 A, 1987.

[13] J. Daugman. Biometric Personal Identification System Based on Iris Analysis. Patent US 5291560 A, 1994.

[14] J. Daugman and I. Malhas. Iris Recognition Border-Crossing System in the UAE. http://www.cl.cam.ac.uk/∼jgd1000/UAEdeployment.pdf, 2004. Retrieved: 13.7. 2013.

[15] J. Daugman. Iris Recognition at Airports and Border-Crossings. http://www.cl.cam.ac.uk/∼jgd1000/Iris_Recognition_at_Airports_and_Border-Crossings.pdf. Retrieved: 14.7. 2013.

[16] Ch. Rathgeb, A. Uhl, and P. Wild. *Iris Biometry: From Segmentation to Template Security*. Springer, 2013.

[17] Humanitarian News and Analysis: A Service of the UN Office for the Coordination of Humanitarian Affairs. Afghanistan-Pakistan: UNHCR Closes Two Iris Verification Centres in Pakistan. http://www.irinnews.org/report/26361/afghanistan-pakistan-unhcr-closes-two-iris-verification-centres-in-pakistan. Retrieved: 13.7. 2013.

[18] J. Nikishkina. Global Commercial Biometrics Market to Grow Considerably as Awareness Increases. *Frost & Sullivan*, 10 2013.

[19] S. Z. Li and A. K. Jain. *Encyclopedia of Biometrics*. Springer, 2009.

[20] International Organization for Standardization. *ISO/IEC 19794-6*, Information Technology – Biometric Data – Interchange Formats – Iris Image Data edition, 2011.

[21] International Organization for Standardization. *ISO/IEC 15444-12*, Information Technology – JPEG 2000 Image Coding System edition, 2012.

[22] R. L. Davis and P. D. Becherer. Techniques for Improved Soft Lens Fitting. *Contact Lens Spectrum*, 8 2005.

[23] Nadia Othman. *A Biometric Reference System for Iris OSIRIS version 4.1*. Biosecure, 2013.

[24] M. J. Burge and K. W. Bowyer. *Handbook of Iris Recognition*. Springer, 2013.

[25] B. J. Kang, K. R. Park, J. Yoo, and K. Moon. Fuzzy Difference-of-Gaussian-Based Iris Recognition Method for Noisy Iris Images. *Optical Engineering*, 49(6):1–10, 2010.

[26] S. A. C. Schuckers, N. A. Schmid, A. Abhyankar, V. Dorairaj, Ch. K. Boyce, and L. A. Hornak. On Techniques for Angle Compensation in Nonideal Iris Recognition. *IEEE Transactions on Systems, Man and Cybernetics*, 37(5):1176–1190, 2007.

[27] J. Lee, P. S. Huang, J. Chang, Ch. Chang, and T. Tu. Iris Recognition Using Local Texture Analysis. *Optical Engineering*, 47(6), 2008.

[28] P. Wojtaszczyk. *A Mathematical Introduction to Wavelets*. Cambridge University Press, Cambridge, 1997.

[29] P. E. T. Jorgensen, K. D. Merrill, and J. A. Packer. *Representations, Wavelets, and Frames: A Celebration of the Mathematical Work of Lawrence W. Baggett*. Springer, 2008.

[30] B. W. Silverman and J. C. Vassilicos. *Wavelets: The Key to Intermittent Information?* Oxford University Press, New York, 1999.

[31] D. Gabor. Theory of Communication. *Journal of the Institution of Electrical Engineers – Part III: Radio and Communication Engineering*, 93(26):429–457, 1946.

[32] D. H. Hubel and T. N. Wiesel. Receptive Fields, Binocular Interaction and Functional Architecture in the Cat's Visual Cortex. *The Journal of Physiology*, 160(1):106–154, 1962.

[33] S. Marčelja. Mathematical Description of the Responses of Simple Cortical Cells. *Journal of the Optical Society of America*, 70(11):1297–1300, 1980.

[34] T. S. Lee. Image Representation Using 2D Gabor Wavelets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(10):1–13, 1996.

[35] J. J. Benedetto, Ch. Heil, and D. F. Walnut. Differentiation and the Balian-Low Theorem. *Journal of Fourier Analysis and Applications*, 1(4):355–402, 1994.

[36] Y. Meyer and R. D. Ryan. *Wavelets: Algorithms and Applications*. Society for Industrial and Applied Mathematics, Philadelphia, 1993.

[37] M. C. Pereyra and L. A. Ward. *Harmonic Analysis: From Fourier to Wavelets*. American Mathematical Society, 2012.

[38] I. Daubechies. The Wavelet Transform, Time-Frequency Localization and Signal Analysis. *IEEE Transactions on Information Theory*, 36(5):961–1005, 1990.

[39] P. E. Black, editor. *Gray Code*. U.S. National Institute of Standards and Technology, 2011. Retrieved: 16.6. 2013.

[40] S. S. Dias. Improved 2-D Gabor Filter. The MathWorks, Inc., 6 2008. http://www.mathworks.in/matlabcentral/fileexchange/13776-improved-2d-gabor-filter.

[41] V. I. Levenshtein. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Cybernetics and Control Theory*, 10(8):707–710, 1966.

[42] F. Hao, R. Anderson, and J. Daugman. Combining Cryptography with Biometrics Effectively. Technical report, University of Cambridge, Computer Laboratory, 2005. UCAM-CL-TR-640, ISSN 1476-2986.

[43] F. J. MacWilliams and N. A. Sloane. *The Theory of Error-Correcting Codes*. North-Holland, Amsterdam, 1977.

[44] W. W. Peterson and E. J. Weldon. *Error-Correcting Codes*. The Massachusetts Institute of Technology, 1961.

[45] I. S. Reed and G. Solomon. Polynomial Codes over Certain Finite Fields. *Society for Industrial and Applied Mathematics*, 8:300–304, 1960.

[46] A. J. Deshpande. Algorithmic Introduction to Coding Theory, Lecture 4. Massachusetts Institute of Technology, Computer Science and Artificial Intelligence Laboratory, 2002. http://people.csail.mit.edu/madhu/FT02/scribe/lect04.pdf. Retrieved: 22.6. 2013.

[47] H. Y. Hsu, J. Ch. Yeo, and A. Y. Wu. Multi-Symbol-Sliced Dynamically Recongurable Reed-Solomon Decoder Design Based on Unified Finite-Field Processing Element. *IEEE Transactions on Very Large Scale Integration Systems*, 14(5):489–500, 2006.

[48] Federal Information Processing Standards. Advanced Encryption Standard (AES), 11 2001.

[49] T. C. Clancy, N. Kiyavashand, and D. J. Lin. Secure Smartcard-Based Fingerprint Authentication. In *ACM SIGMM Workshop on Biometrics Methods and Application*, 2003.

[50] N. K. Ratha, J. H. Connell, and R. M. Bolle. Enhancing Security and Privacy in Biometrics-Based Authentication Systems. *IBM Systems Journal*, 40(3):614–634, 2001.

[51] B. Toth and U. C. von Seelen. Liveness Detection for Iris Recognition. In *Biometrics and e-Authentication over Open Networks*. National Institute of Standards and Technology, 2005.

[52] S. Schuckers, L. Hornak, T. Norman, R. Derakhshani, and S. Parthasaradhi. Issues for Liveness Detection in Biometrics. In *Center for Identification Technology Research*. West Virginia University, 2002.

List of Figures

1.1 An eye from the frontal view 5
1.2 Human eye 5
1.3 Interrelation between FAR and FRR 7
1.4 The problem of setting a suitable level of FAR and FRR 8

2.1 An example of eye captured under NIR illumination 11

3.1 Investigation of a horizontal line in the middle of the image 14
3.2 Computation of Hough 3-D matrix 22
3.3 Pupillary and limbic borders parametrization 25
3.4 Normalized iris texture 26
3.5 An example of a noise mask 28

4.1 Haar wavelet 31
4.2 Modulation of 1-D Gabor wavelet: (a) 1-D sinusoid, (b) Gaussian function, (c) 1-D Gabor wavelet 35
4.3 Modulation of 2-D Gabor wavelet: (a) 2-D sinusoid, (b) Gaussian kernel, (c) 2-D Gabor wavelet 35
4.4 2-D Gabor wavelets at different frequencies and orientation 36
4.5 Segments of iris texture proposed by Kang et al. to be only used . 37

5.1 Iris code consisting of 2,048 bits of information 40

9.1 List of attacks with the phases in which they are exerted 59

List of Abbreviations

Abbreviation	Full Name	Page
EER	Equal Error Rate	7
FAA	False Acceptance Attack	60
FAR	False Acceptance Rate	7
FCS	Fuzzy Commitment Scheme	54
FRR	False Rejection Rate	7
GRR	Genuine Rejection Rate	7
MDS	Maximum Distance Separable	53
MW	Mesh Warping	58
NIR	Nearly Infrared	10
ROI	Region of Interest	14
VW	Visible Wavelength	10