Master’s Thesis

Fixed-Length Feature Vectors for Biometric Template Protection of Fingerprint Data

Department of Computer Science
Hochschule Darmstadt
&
University of Wisconsin - Platteville

Jan Hirzel

Filling Date : September 20, 2010

Committee Chair : Prof. Dr. Christoph Busch Hochschule Darmstadt
Committee Member : Prof. Dr. Harald Baier Hochschule Darmstadt
Committee Member : Prof. Dr. Mike Rowe UW - Platteville
Committee Chair - (Prof. Christoph Busch)

Committee Member - (Prof. Harald Baier)

Committee Member - (Prof. Mike Rowe)

Date of Thesis Defense: 30th of August, 2010
I hereby certify, that this masters thesis together with the corresponding source code has been created by me autonomously. No other sources than those mentioned were used.

Gjøvik, September 20, 2010

( Jan Hirzel )
Acknowledgement

It is a pleasure to thank those who made this thesis possible.

First of all I want to thank my advisor Prof. Dr. Christoph Busch whose encouragement, guidance and support from start to the very end allowed me to develop and understand the subject as well as making it possible to write the thesis at NISLab in Norway. I also want to thank Daniel Hartung for his eagerness to help, the self-evident support on every day of the week and the will to push me towards my limits.

Various other persons have contributed to the completion of this thesis. I would like to show my gratitude to my friend Tobias Nöll who spent a considerable time answering a myriad of questions about Latex as well as his enlightening discussions about algorithms, mathematics and personal matters during the time. I also want to thank my cousin Cliff Zimsen who always had an open ear for my various explanations about fingerprint recognition and the staff of NISLab who was always willing to exchange ideas and giving thoughts to current problems. Finally I want to thank my friends and my family who provided me with assistance during my stay in Gjøvik.

You all made this thesis possible!
# Contents

1. Introduction 8

2. State of the Art 11
   2.1. Minutiae Matching 12
   2.2. Fingerprint Correlation 16
   2.3. Finger Pattern Matching 18

3. Biometric Performance 20

4. Pattern Matching Using Co-Sinusoidal Triplets 24
   4.1. Image Pre-Processing 24
   4.2. Cellular Partitioning 25
   4.3. Spectral Component Selection 26
   4.4. Candidate Matrix 30
   4.5. Processing Pipeline 32
   4.6. Comparison of Triplet Sets 33
      4.6.1. Direct Comparison Method (DCM) 34
      4.6.2. Weighted Direct Comparison Method (WCM) 34
      4.6.3. Neighborhood Comparison Method (NCM) 36
   4.7. Cell Size Benchmark 38
   4.8. Parameter Evaluation 39
   4.9. Algorithm Improvements 41

5. Biometric Template Protection 42
   5.1. Introduction 42
   5.2. Template Protection Schemes 44
   5.3. Helper Data Scheme 46
   5.4. Error Correcting Codes 48
   5.5. Data Preprocessing 49
   5.6. Tests 51
      5.6.1. BCH Code Triplets 51
      5.6.2. N-Tuple Combination 52
      5.6.3. Optimum Cell Size 54
      5.6.4. Layer-Based Order 54
Contents

5.6.5. Multiple Verification Samples .......................... 55

6. Conclusion .................................................. 57
7. Further Steps .................................................. 58
Bibliography ...................................................... 60
8. Glossary ....................................................... 63
9. Appendix ......................................................... 64
A. Parameter Evaluation ......................................... 65
B. Cellsize Evaluation ........................................... 73
C. HDS Test Results ............................................. 82
Abstract:
This thesis is focused on fingerprint ridge pattern comparison in combination with biometric template protection (BTP). The feature extraction algorithm described is based on the information of the fingerprint ridges as a whole and not just singular, unique points. There are numerous ways how the pattern can be examined, one of the most prominent ones though is described in the ISO standard of 2009 [fS06b] and will be discussed herein. The approach investigated in this thesis has the advantage that features - unlike finger minutiae data - can be easily transformed to a fixed length feature vector which is required for further processing such as template protection methods. In the second part of the thesis, the reference architecture for biometric template protection is introduced. The integration of the developed feature extractor into the biometric template protection process according to the Helper Data Scheme is investigated and tested.
1. Introduction

What is Biometry?
The word biometry is derived from the Greek language and translates to "the observation of characteristics of the human body for the purpose of identification (recognition)". Hence, it addresses the identification or verification of human beings based on what they are or how they act. It is further defined as

"Automated recognition of individuals based on their behavioral and biological characteristics." [FS08]

Human individuals have a wide range of measurable characteristics like height, weight, facial expression, fingerprints, veins, iris patterns, etc. but not all of those modalities fulfill the needs for automatic recognition. A biometric modality should be universal, i.e. it is measurable on all individuals equally. It needs to be unique for each participant and up to a certain degree, persistent over time. Furthermore, it needs to be measurable, convenient in use, acceptable for the targeted audience, provide a certain degree of security, and achieve a good biometric performance.

Why use Biometrics?
Identity authentication can be categorized into three different classes. It can be knowledge based (for example a password or a PIN), possession based (for example a token or a key) or determined by what the individual is (for example a body characteristics or behavior). The problem with knowledge and possession based systems is, that the secret or token can be lost, forgotten or forwarded to other individuals, whereas biometric characteristics cannot be.

There are two ways how a biometric system can work. It can identify or verify individuals. Identification is the ability to recognize an individual amongst all enrolled subjects of a system, while verification is the comparison of a person with who he claims to be.

History of Fingerprint Authentication
Fingerprints, as a unique biometric characteristic [PPJ01], have been used for identification and verification purposes since centuries and have been proven to be reliable ever since. Even though it cannot be formally proven that they are unique for each individual, there is no record of any two persons having the same
1. Introduction

fingerprint. Albeit widely used in criminalistics, since the spreading of fingerprint sensors in home computers and laptops they gained a never seen before popularity.

In order to identify an individual it is not necessary to compare each and every part of the fingerprint but it is sufficient to use a subset of features that allow a positive identification. In traditional approaches like in Jain et al. [JHPB97a] these features are the minutiae, the ridge endings and bifurcations of the epidermal ridges of a finger. Each minutia consists of a location (x, y) and an orientation. For each fingerprint image a set of minutiae is extracted and when comparing them, the decision is made depending on their correlation. There are alternative approaches to this method which do not rely on minutiae, like fingerprint correlation [BCP03] or finger pattern comparison [Hup07].

What are Fingerprint Images
A fingerprint image is a digital 2D representation of the epidermal fingerprint ridges on a finger captured by a sensor. Fingerprint sensors, no matter on what hardware they are based on, will generate a gray scale image like the one in Figure 1.1. The dark lines in the image are called ridges and are a planar copy of the epidermal ridges found on a finger. The white components within the fingerprint are called valleys. In traditional fingerprint recognition as described by Francis Galton [Gal92], the identification is based on points which, as a set of points or point cloud, uniquely describe a fingerprint. Those points reside on the ridges of the fingerprint image and therefore a good projection of the 3 dimensional, epidermal ridges to the 2 dimensional image is the base for reliable fingerprint recognition.

Importance of Security
With the high distribution of fingerprint sensors and the coalescence and advancements of computer systems, new threats appear as well. Advantages are taken from the additional information that is typical for certain biometric characteristics and that are encoded in biometric data. Through cross referencing over different applications or even different databases new dangers arise. The only way to effectively protect a biometric system against fraudulent use and the privacy of the user is by using cryptographic counter-measures that will ensure the safety of the stored biometric data.

Identity theft is still one of the biggest challenges of law enforcement and according to [Cen10] only 1.2% of all purloined identity data had either encryption or other security features protecting exposed data. With digital paperwork on the
1. Introduction

Figure 1.1.: A fingerprint image as captured by a fingerprint sensor. Source: Newsweek (http://current.newsweek.com/budgettravel/fingerprint.jpg) & FVC DB2002 DB1_a File: 1_1.tif

rise and more and more user accounts mandatory in each sector of everyday lives, it is clear that a high level of protection is inevitable especially for such personal data like biometric templates.
2. State of the Art

As mentioned in Chapter 1, there is a wide variety of different fingerprint recognition systems and techniques currently established on the market. All of the biometric systems work on the same simplified principle as shown in Figure 2.1. A prerequisite to use a biometric system is that a user has to be enrolled in the system first. The enrollment step is mandatory in all systems; no matter what biometric characteristic, which sensor or feature extraction algorithm is used. The enrollment only has to be done once and a biometric reference will be created which is used for authentication. The verification of users operates very similar to the enrollment. In the enrollment step, a sensor takes a biometric sample and passes it to the feature extraction stage. The feature extraction will generate the biometric data that will comprise the biometric reference. If the biometric data is of sufficient quality, it will be store as the reference in the database. All verifications (or in the case of an identification system, identifications) will be based on this reference, therefore it is very important that it is of utmost quality. After this step the user may authenticate at the system as often as desired. For the authentication (or verification), the user presents the same biometric characteristic as used for the enrollment to a sensor. The sensor then takes a biometric sample and passes it to the feature extractor. The feature extractor gathers the biometric data from the sample and passes it to a comparison and decision unit. The comparison and decision unit also takes the biometric reference which is stored in the database as an input and depending on the outcome of the examination the user is accepted or rejected.

Currently there are three different, well established techniques on the market to extract features from fingerprint samples, namely Minutiae Matching (see [JHPB97a]), Fingerprint Correlation (see [BVG*00]) and Cell Pattern Matching (introduced in Chapter 4). In this chapter, each of the three different techniques will be introduced.
2. State of the Art

![Diagram of fingerprint identification system](image)

Figure 2.1.: The base of each fingerprint identification or verification system.

2.1. Minutiae Matching

The technique of minutiae matching is based on the unique points located on the fingerprints, called minutiae. A minutia is a significant point on a fingerprint, either a bifurcation of a ridge or a ridge ending (see Figure 2.2). A minutia is described by its position on the fingerprint and its orientation. Those minutiae points are sufficient to uniquely describe a fingerprint since the amount of minutiae, their positions, orientations and classifications vary from finger to finger and of course person to person. A typical minutia based fingerprint comparison algorithm works as follows:

1. Image Preprocessing
2. Feature extraction
3. Count corresponding minutiae
4. Determine comparison score

*Image Preprocessing:* The image preprocessing step has a big impact on the performance of the system and should not be underestimated. When processing images of low quality, it is very likely that less minutiae points will be extracted and therefore, the probe will have fewer conformance with a reference consequently leading to a lower score. The three most important image preprocessing methods
2. State of the Art

Figure 2.2.: The minutiae points on a fingerprint. On the left is a typical ridge ending, on the right a typical bifurcation. Source: [JHPB97b]

are histogram stretching, normalization and image smoothing.

Histogram stretching, as explained by Alparslan and Fuatince [AF81], is a contrast enhancing technique based on the intensity histogram of the fingerprint image. By simply rearranging the present gray scale values, one can increase the distance between neighboring brightness ranges. This leads, in cases of low contrast images, to significant improvements.

Normalization is a mathematical tool that, when applied to the brightness channel of an image, leads to a better contrast. It projects a co-domain onto another, allowing further increase in contrast to suit specific needs of following algorithms. One of the most renowned normalizations is the binarization, where all present values will be projected to 0 or 1.

Image smoothing is a low-pass filter in the spatial domain of the image and allows minimizing the presence of noise and outliers. This can be achieved by looking at a neighborhood (for example 3x3) of a pixel and calculating the median or average of that area. Another famous representative for image smoothing techniques is the Gaussian Blur.

The feature extraction can be broken down into five small steps: creation of a directional image, extraction of the region of interest, detection of the singularity points, skeletonizing, and minutia detection.

A directional image assigns an orientation to each pixel of an image. For a fingerprint image this is the direction in which the ridges flow at the specific position. The directional image can be retrieved with techniques explained in [MMKC87] and will be used to determine the orientation of the minutiae later.

In general, when a sensor captures a fingerprint, there are some areas that contain
2. State of the Art

ridges and areas that do not hold any information. This is for example the case when the area of the fingerprint is smaller than the area that is captured by the sensor (see Figure 2.3). In order to be more efficient and reduce errors, the region that does not contain any information about the fingerprint will be discarded. The area that does not contain any information is also called background and the area that holds information is called foreground. There are numerous ways how this can be achieved - one way is by analyzing the certainty field of the orientation map (as described in [JHPB97b]) or by image contrast analysis.

![Figure 2.3: The captured fingerprint image on the left and the fingerprint with discarded background information on the right. Source: [Hup07]](image)

The next step is the detection of singularity points, namely the core and delta points. Depending on the fingerprint class, there can be up to two core points. Core and delta points can be used as reference points in the fingerprint image and are used to help building up point relations for minutiae. See Figure 2.4 for examples of core and delta points in a fingerprint image.

A core point is defined by [FS06a] as "A singular point in the fingerprint, where the curvature of the ridges reaches a maximum".

A delta point is defined by [FS06a] as "structure where three fields of parallel ridge lines meet".

With the orientation image already present, it is easy to extract the core and delta points from it with the help of the Poincaré index (see [HJ99]).

Skeletonizing or thinning is the process of reducing the ridges (or black lines) in the fingerprint image to the minimum width of one pixel. The lines will be
2. State of the Art

Figure 2.4.: The two different reference points on a fingerprint. There are classes of fingerprints that do not have a delta and others that have two cores. Source: [http://geradts.com/anil/ij/vol_002_no_001/papers/paper005.html](http://geradts.com/anil/ij/vol_002_no_001/papers/paper005.html)

reduced to the center of the ridges. With that, the minutiae can be detected on pixel level. Skeletonizing can be achieved through ridge line tracing algorithms in gray scale images or simple erosion algorithms in binary images.

Finally the minutiae points can be detected in the resulting image. The localization can be achieved by looking at N4 neighborhoods for each pixel. A point on a ridge with at least one white pixel in its neighborhood is called a border pixel and is thus a potential candidate for a minutia point. As mentioned above, ridge endings and bifurcations are possible minutiae points. In order to determine which of the two the border point is, further investigations have to be made. When following the dominant ridge flow direction which can be obtained from the orientation image, the center of orthogonal crossing lines leave a clear path when they meet the border. Then the best minutia position is found and the point is a ridge ending. Bifurcations can be detected by following the dominant ridge flow (as when detecting a ridge ending) and when the center paths of three branches meet in a single point or form a small triangle within a ridge at the border point, a bifurcation has been found. The center-of-mass of this triangle is the best position of the minutia.
2. State of the Art

After the minutiae are detected, a quality check that counts the number of minutiae will decide whether they are enough to distinctively describe the fingerprint (see Figure 2.5 for an example of detected minutiae points). If the user is in the enrollment phase, the minutiae points will be stored in the reference archive. Otherwise, the minutiae points will be compared to the points from the reference. Correlations between the point clouds will be determined by looking at delta neighborhoods around each minutia point. In the case that enough points from the probe correlate to points in the reference, the comparison will result in a match for the user.

Figure 2.5.: The detected minutiae points marked in a fingerprint image.
Source: http://fingerchip.pagesperso-orange.fr/biometrics/types/fingerprint_algo.htm

2.2. Fingerprint Correlation

The fingerprint correlation method was introduced due to the problems of minutiae based matching that arise with bad image quality. When a fingerprint image is of low quality through low contrast or distortion which cannot be corrected by image processing algorithms, the minutiae based approach may fail due to the fact that it simply cannot extract enough minutiae points. The correlation
2. State of the Art

The method introduced by [BVG00] is based on the gray level information present in the image. Even when an image is of bad quality, the gray levels of the image still contain enough discriminatory information to allow verification.

The correlation approach creates small cells on the fingerprint that will contain enough information to uniquely identify a fingerprint. Here, a template consists of a sufficient number of these small areas which, in the verification, will be compared to the areas at the same positions. Bazen suggests three different methods on how to select the cells. Minutia based template selection, coherence based template selection and correlation based template selection.

Minutia based template selection extracts minutiae points similar to the approach in the previous chapter. Instead of saving the minutiae points directly, it selects an area around the minutiae points and stores the area. This will only be done in the enrollment. In the verification phase, the areas will be taken from the reference and the comparison algorithm will look for this area in the probe. Of course the problems that arise with minutiae matching are also affecting this approach during the enrollment and therefore it is not the recommended template selection approach.

In the coherence based template selection approach, the areas that will represent the fingerprint will be chosen by looking for areas of low coherence. The coherence of an area is defined as the measure of how many of the ridges passing through flow in the same direction. For example a high coherence is present when all ridges flow in the same direction, whereas low coherence is at hand when ridges are flowing in different directions in the same area. Areas of high coherence are not very distinctive since they contain only little to no unique information. As a matter of fact, the areas that are most suitable are located around minutiae points, even though the minutiae are not explicitly extracted beforehand.

Correlation based matching selects the template areas according to the following approach. An area is selected and then it will be compared against all other possible cell positions on the fingerprint. If there is another position that is very similar or exactly the same to the selected one, the selected area is not very distinctive. If the area can only be found once it is very individual for the print and a good candidate for an area that can substitute the print. This process is repeated with different areas until sufficient distinctive cells have been found. This approach produces the most suitable results since it satisfies the template requirements best. The drawback, of course, is that the approach costs a lot of resources which makes it less attractive.
Bazel suggests a hybrid approach for the template selection, combining all three methods. After the user is enrolled, the system can verify the user by using the preselected areas from the reference and comparing them to the areas that will be found at those positions on the probe. Naturally, in order to be able to compare the same positions, the probe has to be aligned with the reference. This is achieved by sliding the templates over the probe and finding the position at which the distance of the squared gray levels is smallest. This is of course also very costly on resources, but the advantage is that it makes the system less error prone to distortions. Besides, the process can be speedup when the image is normalized to the range of $[0 \ 1]$. Then a convolution or a filter can be used to find the best position for the area. Furthermore, if the size of the fingerprint is chosen correct, a 2D Fast Fourier Transformation can be used.

2.3. Finger Pattern Matching

Finger pattern algorithms are based on spectral information that is present in fingerprints rather than certain local point patterns. What all finger pattern matching algorithms have in common is, that the fingerprint will be divided into small cells and each cell will be represented by a substitute. Depending on the used algorithm, the substitute consists of different parameters for the employed models.

The fingerprint image is divided into cells (tessellation) of variable size and shape. The cells do not have to cover the whole information, but usually cover the whole Region of Interest (ROI) of a fingerprint, i.e. parts of the image where ridges are present. There are three different ways proposed by [FS06b] on how to encode the spectral information of the cells, Gabor filters, Discrete Fourier Transformation (DFT) or Quantized Co-Sinusoidal Triplets (QCT). The encoding of the spectral information of each cell is based on the given underlying mathematical model that allows the representation of the whole cell through a small set of parameters. The complete ensemble of cells will be stored as the biometric reference.

In the enrollment phase, the fingerprint image is taken and an image preprocessing, like in Chapter 2.1, is carried out if required. The image is then divided into the cells and the spectral information is extracted. The resulting parameters will be stored in the reference archive.

For the verification, a probe image is taken from a sensor, preprocessed and then the image has to be aligned to the reference. The alignment needs to cover translations as well as rotations and deformations. This can be a challenging task and is still subject of current research. It can be performed through a global or local
2. State of the Art

alignment. A global alignment will translate and rotate the probe according to a reference that can be extracted from the fingerprint image (for example core or delta points). A local alignment can be done on cellular basis, where the information in the cells from the references will be used to align the probe. After the alignment, the fingerprint is tessellated and the information from the cells extracted. The cells of the reference and the probe will be compared and if they are sufficiently conforming, result in a match.

In respect of the topic of this thesis, the creation of fixed-length feature vectors, the finger pattern matching algorithms are the most promising approaches. They do not rely on volatile data like minutiae points or stable areas but already offer the opportunity to fix the amount of template data for all participants. Furthermore the QCT approach has been chosen as the spectral data extraction mechanism since it has already been proven by Bioscrypt Inc. at the FVC 2000 to provide reliable results (see [Bio10] for more information). According to the 'Introduction' chapter of [FS06b], the Quantized Co-Sinusoidal Triplet method has been patented by Bioscrypt Inc.
3. Biometric Performance

To determine how well a system performs, it is necessary to have metrics that can objectively quantify the "how well". Biometric metrics were introduced to provide an objective and universal method to compare the performance of biometric systems, no matter on which modality they are based upon. The result of a comparison of two biometric samples is a comparison score that can be projected to the range of $[0, 1]$. The closer the score is to 1, the higher is the probability that the samples originated from the same source. The closer the comparison score is to 0, the smaller is the certainty of the system that the samples originated from the same source. The decision whether the comparison leads to a match or a non-match, is made based on a threshold parameter $t$. If a comparison score is equal to, or above the threshold, the comparison is considered a match. Is it below $t$ it is regarded a non-match.

A biometric system can commit two different types of errors. It can falsely determine two samples to be from different biometric sources, even though they are from the same and it can falsely ascertain that two samples are from the same biometric source, even though they are from different sources. These errors can be caused by several reasons and therefore the classification can be further narrowed down and are defined in the Harmonized Biometric Vocabulary [FS08].

*Failure to Capture (FTC)*: Errors that occur during the capturing of the biometric sample. This includes errors that are caused by the user (e.g. the finger is not placed properly on the sensor), errors caused by the image sensor, but also samples rejected by an automated quality control method.

*Failure to Extract (FTX)*: FTX errors occur if the feature extraction algorithm is not able to generate features for the biometric template from the provided source. This could be either caused by the fact, that the feature extraction was not able to identify an adequate amount of features or that the processing time exceeded the specified limits.

*Failure to Enroll (FTE)*: When a system is not capable of creating a biometric reference for verification, a failure to enroll error occurred. This metric describes
3. Biometric Performance

the amount of persons that are unable to participate in the system. This can be caused by non-existence of the biometric characteristic on an individual or by the disability to generate a template of sufficient quality from the individual’s characteristics.

**Failure to Acquire (FTA):** The FTA is the inability to generate biometric features from a probe during the verification process. Therefore, it combines the FTC and FTX during the verification phase.

The phases in which the errors will be introduced can be seen in Figure 3.1. With every advancement towards the enrollment (or verification) the error possibilities accumulate.

![Diagram: Phases in which errors can be introduced into the system](image)

**False Match Rate (FMR):** The FMR is the number of cases in which the system falsely accepts a user, even though he is not enrolled in the system or is enrolled as a different user (impostor). The FMR is not associated to the above mentioned failure rates and therefore only evaluates the intrinsic algorithm. This metric is system critical, since it is concerned with the security of the system.

**False Accept Rate (FAR):** like the FMR, the FAR also measures the cases where a user is falsely accepted in a system. Additionally to the intrinsic algorithm it also considers the extrinsic parameters that are defined by the above mentioned errors (FTC, FTX, FTE and FTA).
3. Biometric Performance

*False Non Match Rate (FNMR):* The FNMR is the number of cases in which a user was falsely rejected even though he is a valid user of the system (Genuine). The FNMR is, like the FMR, not associated with any other error rates and therefore evaluates the intrinsic algorithm rather than the whole system. The FNMR is considered a comfort criterion, since no further harm can be done from a falsely rejected user.

*False Reject Rate (FRR):* The false reject rate is, same as the FNMR, a measure of how many users are falsely rejected from the system. Same as the FAR, it is based on the performance of the intrinsic algorithms and the extrinsic parameters of the system (FTC, FTX, FTE and FTA).

The FAR and FRR are typically used to evaluate the performance of a whole biometric system, since they also take errors caused by users and sensors into account. Thereupon, the FAR and FRR are highly dependent on the user group and the hardware, whereas the FMR and FNMR exclude all sensor and user dependent errors and thus are perfect for comparing different algorithmic approaches and are the key to system testing. Typically, the FAR and the FRR are drawn into a single plot for further evaluation (see Figure 3.3). That way, it is easy to determine how the system performs for different applications and how the threshold \( t \) should be chosen.

*Equal Error Rate (EER):* The EER is reached when the FAR and the FRR are exactly the same. This allows a comparison between different biometric systems and algorithms without considering the threshold \( t \) that significantly affects the view on the performance of a biometric system. It also allows comparison of biometric systems that utilize different biometric modalities, thus making it the most important metric for evaluating algorithms and systems. The EER can be obtained from the ROC Curve. It is at the point where the axis bisection intersects with the curve.

*Receiver Operating Characteristic (ROC):* The ROC curve combines the information of the FMR and the FNMR in a single plot which allows fast insight into the behavior of the system in different applications (see Figure 3.2). Usually at least one of the axes is plotted in the logarithmic scale.
3. Biometric Performance

Figure 3.2.: The Receiver Operating Characteristic (ROC) Curve with both axes in logarithmic scale.

Figure 3.3.: Score Distribution with threshold t on the x axis and the probability on the y axis. The green curve indicates the genuine access attempts, the red curve the impostor attempts.
4. Pattern Matching Using Co-Sinusoidal Triplets

The first step in the generation of a fingerprint template is the conversion of the fingerprint to spectral data. The steps suggested in the ISO standard [FS06b] are as follows:

- Image Pre-Processing
- Cellular Partitioning
- Spectral Component Selection

The image preprocessing step is optional but can have a great effect on the results, since poor image quality leads to missing and spurious features.

4.1. Image Pre-Processing

Due to the nature of the individual characteristic of a human finger (scars, medical skin condition, etc.) and the fact that a fingerprint image taken by a sensor is spoiled by the current condition (dry, wet, oily, etc.) of the finger, the image quality may differ drastically depending on the time of the acquisition. Moreover, the image might also be differently distorted, when different sensors are used (for example contact based or contactless). This leads to fingerprint images with widely varying qualities as can be seen in Figure 4.1. In order to overcome this problem and increase the biometric performance of the system, the quality of the images provided by a fingerprint sensor might be enhanced before any features are extracted. A basic improvement that does not consume much processing power is the histogram normalization of the fingerprint image (as for example proposed by Alparslan and Fuatince [AF81]) which already leads to a more consistent image quality. Further enhancements include Gabor filtering [YLJF03], image segmentation [HJ04], binarization [Tha03], and thinning [Tha03] of the fingerprint images.
4. Pattern Matching Using Co-Sinusoidal Triplets

Figure 4.1.: Fingerprint images with different image qualities. The quality depends on the characteristic and condition of the finger, the image sensor, and the pressure used when obtaining the image. Source: http://www.secugen.com/images/faq03.gif

4.2. Cellular Partitioning

Figure 4.2.: The tessellation of a fingerprint with an optional X- and Y-Offset.

After image preprocessing, a tessellation of the fingerprint image has to be done. Therefore, the fingerprint will be divided by a grid into cells of size $w \times w$. The ISO standard [fS06b] does not give any recommendation about the specific size of the cells but when using quantized co-sinusoidal triplets for the spectral component selection it is suggested to have a maximum of approximately two ridges per cell. Thus, it varies depending on the resolution of the sensor how big the cells should
be. Vannfält and Åström [Vs06] suggested in their thesis a cell size of $5 \times 5$ pixels for a sensor resolution of 250ppi.

If there is space at the top and left border that does not contain any fingerprint information, an offset can be chosen to get a maximum coverage of the fingerprint area. Any cells that cannot be fully filled with image information (the rightmost and bottommost cells) will be discarded.

4.3. Spectral Component Selection

Next, the spectral information of each cell needs to be approximated. There are three different, already established retrieval methods. They are: Quantized co-sinusoidal triplets, Discrete Fourier Transformation (DFT) and Gabor filters. There are numerous publications about the use of the DFT (like Bracewell [Bra89], Nussbaumer [Nus82] or Winograd [Win76]) as well as the use of Gabor filters (like Yang [YLJF03] or Huppmann [Hup07]) but only the master’s thesis of Vannfält and Åström [Vs06] was concerned with QCT, even though an algorithm using QCT won the FVC of 2000. It is proven to produce reliable results and is rather simple to implement, hence the quantized co-sinusoidal triplet method was chosen as the spectral component selector.

Quantized Co-Sinusoidal Triplets

The method of quantized co-sinusoidal triplets is based on the approximation of each finger pattern cell through a cosine triplet ($\theta$, $\lambda$, and $\delta$). As seen in Figure 4.3, the three parameters describe an angle of propagation, the wavelength and the frequency. The range of the parameters can be restricted to a minimum and maximum to get the highest variance without repeating structures.

Angle of propagation $\theta$: represents the directional information of a cell. The angle of propagation is measured perpendicular to the crest of the co-sinusoidal function. If the crest is parallel to the vertical axis $x$, the angle is 0 and it increases with counter-clockwise rotation. It is specified in radians and therefore the interval $[0, \pi]$ describes all possible orientations of the function.

Wavelength $\lambda$: describes the occurrence of ridges and the distance between them for one cell. The valid interval is: $[\text{minimum spatial wavelength, } \infty]$, whose lower end is in practice limited by twice the length of the longest diagonal for a cell. The frequency $f$ is directly related to this parameter since it is defined as $\lambda = \frac{1}{f}$. The range of the frequency is $[0, \text{maximal spatial frequency}]$ where the maximal spatial frequency is the Nyquist frequency (as described for signal theory by
4. Pattern Matching Using Co-Sinusoidal Triplets

Figure 4.3.: Left: A cell taken from the fingerprint. Right: The approximation through the co-sinusoidal triplets. The $\delta$ parameter is defined as $\delta = \frac{d}{\lambda} \cdot 360$ degrees.

Grenander [Gre59]. For 2D signal processing the Nyquist frequency is equal to the length of the image diagonal divided by two.

Phase $\delta$: describes the distance of the first crest to origin of the cell. It is specified in angular coordinates and therefore is in the interval of $[0, 360]$. It is defined as $\delta = \frac{d}{\lambda} \cdot 360^\circ$, where $d$ is the distance.

The 2D co-sinusoidal function to approximate the cell is defined as follows:

$$\text{Cell}_{\theta, \lambda, \delta}(s, t) = \cos(P \cdot 2\pi \cdot f + \delta),$$

where $P = s \cdot \cos(\theta) - t \cdot \sin(\theta)$,

and $f = \frac{1}{\lambda}$

The parameters $s$ and $t$ of the function describe the position of each pixel inside the cell and the valid interval for $s$ and $t$ respectively is defined as $s = [1, \text{image width}]$ and $t = [1, \text{image height}]$ where $s, t \in \mathbb{N}$. The resulting values of the function will be quantized and are accurate enough to reconstruct the ridges of the fingerprint depending on the precision for each parameter. The ISO standard [fS06b] suggests a linear distribution for the quantization and is defined for each value. The chosen values are arranged such that they lie in the defined intervals. The amount of possible values of the quantization (bit-depth) depends highly on the cell size. The smaller the cells are the less information is necessary to approximate a cell adequately. However, smaller cells also denote that more cells are necessary in order to reproduce the fingerprint image. Therefore, it is necessary to find a suitable bit-depth depending on the resolution of the cells. The bit-depth determines the precision of the approximation, where a higher bit-depth gives more information that can be used to recreate the original image (see Figure 4.4 for an example).
4. Pattern Matching Using Co-Sinusoidal Triplets

Figure 4.4.: On the left is the original fingerprint image, on the right is the synthetic resemblance using the previously gathered triplets. Source: FVC2000 DB.1a

In order to select the most suitable triplet to approximate the cell, the following approach is chosen. First a normalization of the fingerprint values to the range $[-1, 1]$ is done. Afterwards, the distance between the fingerprint cell and all possible synthetic cells (depending on the chosen bit-depths) will be calculated and the synthetic cell structure with the minimum distance will be used to represent the information of the fingerprint cell. The euclidian distance function (see Equation 4.2) is used as suggested in the standard, even though different distance functions (like for example hamming distance) can be used.

$$D = |A_{1,1} - B_{1,1}| + |A_{1,2} - B_{1,2}| + \cdots + |A_{m,n} - B_{m,n}|$$

For $A$ being the set of all pixels in the fingerprint cell and $B$ being the set of all pixels in the candidate cell structures.

In the case that there is more than one cell with the same distance, the following prioritization shall be employed.

1. The triplet with the lowest frequency ($\delta$) value shall be chosen
2. The triplet with the highest wavelength ($\lambda$) value shall be chosen
3. The triplet with the lowest angle of propagation ($\theta$) value shall be chosen

Concluded, the number of the possible cells is dependent on the bit-depth chosen for the parameters and is calculated as follows $2^{l+m+n}$ where $l, m$ and $n$ respectively define the bit depth for $\theta$, $\lambda$, and $\delta$. This is further specified in Table 4.1, 4.2 and 4.3.
4. Pattern Matching Using Co-Sinusoidal Triplets

<table>
<thead>
<tr>
<th>Value (l-bits)</th>
<th>θ</th>
</tr>
</thead>
<tbody>
<tr>
<td>0...000</td>
<td>0</td>
</tr>
<tr>
<td>0...001</td>
<td>$\frac{1}{2^l} \cdot 180$</td>
</tr>
<tr>
<td>0...010</td>
<td>$\frac{2}{2^l} \cdot 180$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>1...110</td>
<td>$\frac{2^{l-2}}{2^l} \cdot 180$</td>
</tr>
<tr>
<td>1...111</td>
<td>$\frac{2^{l-1}}{2^l} \cdot 180$</td>
</tr>
</tbody>
</table>

Table 4.1.: l-bit representation for the angle of propagation $\theta$. Source: [FS06b]

<table>
<thead>
<tr>
<th>Value (m-bits)</th>
<th>$f$</th>
<th>$\lambda$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0...000</td>
<td>0</td>
<td>$\infty$</td>
</tr>
<tr>
<td>0...001</td>
<td>$\frac{1}{2^m} \cdot \text{Nyquist}$</td>
<td>$\frac{1}{7}$</td>
</tr>
<tr>
<td>0...010</td>
<td>$\frac{2}{2^m} \cdot \text{Nyquist}$</td>
<td>$\frac{1}{7}$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>1...110</td>
<td>$\frac{2^{m-2}}{2^m} \cdot \text{Nyquist}$</td>
<td>$\frac{1}{7}$</td>
</tr>
<tr>
<td>1...111</td>
<td>$\frac{2^{m-1}}{2^m} \cdot \text{Nyquist}$</td>
<td>$\frac{1}{7}$</td>
</tr>
</tbody>
</table>

Table 4.2.: m-bit representation for the wavelength $\lambda$. Source: [FS06b]

<table>
<thead>
<tr>
<th>Value (n-bits)</th>
<th>$\delta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0...000</td>
<td>0</td>
</tr>
<tr>
<td>0...001</td>
<td>$\frac{1}{2^n} \cdot 360$</td>
</tr>
<tr>
<td>0...010</td>
<td>$\frac{2}{2^n} \cdot 360$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>1...110</td>
<td>$\frac{2^{n-2}}{2^n} \cdot 360$</td>
</tr>
<tr>
<td>1...111</td>
<td>$\frac{2^{n-1}}{2^n} \cdot 360$</td>
</tr>
</tbody>
</table>

Table 4.3.: n-bit representation for the phase $\delta$. Source: [FS06b]
4. Pattern Matching Using Co-Sinusoidal Triplets

4.4. Candidate Matrix

When following the algorithm with a precision of 3 bits for l, 3 bits for m, and 3 bits for n then the resulting cell candidates can be seen in Figure 4.5 when concatenated. After selecting the triplets with the closest distance to a fingerprint cells, the resulting synthetic print looks like Figure 4.6(b). The original print can be seen in Figure 4.6(a) and the synthetic print when using a lower precision (2 bits for each parameter) in Figure 4.6(c). This was achieved by calculating the synthetic cells at the respective positions. The size of the cells was 5x5 as suggested by [Vs06].

Figure 4.5.: This is a candidate Matrix, where $\theta$ increases in each 8x8 quadrant, $\lambda$ increase in every row and $\delta$ increase in every column.
4. Pattern Matching Using Co-Sinusoidal Triplets

Figure 4.6.: The original fingerprint and two synthetic restorations.
4.5. Processing Pipeline

The following processing pipeline was used which summarizes the mandatory steps to transform a fingerprint image into the co-sinusoidal triplet representation. With the premise to support an unknown number of fingerprint scanners which use different types of fingerprint image formats and different scales, image preprocessing is inevitable. Since the focus of this thesis is on fixed-length feature vectors, image preprocessing will be handled by the VeriFinger SDK. In order to produce a fixed set of feature vectors, the information gathered from a fingerprint must be limited to a mutual size. The most obvious approach is, to find a reference point in the fingerprint that is stable for all occurrences. There are several points in a fingerprint that fulfill this definition, namely the core point and the delta. Cores are the most common used reference points even though there are fingerprint classes, that do not contain a core (plain arch) and some that contain more than one. However, the plain arch type has a likelihood of occurrence of less than 5% and therefore is still a prudent selection as a reference point. A different promising approach is introduced by [AJ07] which selects the point of the highest curvature as the reference point. This method will result in a single point for plain arches as well as a single point for double loops, which have two cores.

1. After the reference point is extracted, the fingerprint image will be translated and rotated according to the position and orientation of the reference point. The result of the transformation will be a rotated image with the reference point at the center of the image.

2. The rotated and translated image is then segmented at the reference point according to the tessellation pattern.

3. Triplets for each cell of the segmentation are extracted.

As can be seen in Figure 4.7 an already preprocessed fingerprint enters the processing pipeline. Several enhancement algorithms are introduced in Chapter 4.1 but the whole preprocessing step is done by VeriFinger as mentioned earlier. This includes a contrasts enhancement, image smoothing, binarization, and core point detection.

The fingerprint will be translated according to the position of the core so the core is in the center of the image. Furthermore, the image will be rotated around the orientation of the core therefore reducing the rotation problem to a minimum.

1http://www.neurotechnology.com/verifinger.html
4. Pattern Matching Using Co-Sinusoidal Triplets

The information about the position and orientation of the core is extracted in the preprocessing step.

After the fingerprint is aligned at the core, the tessellation (or segmentation) is done. The center of the tessellation is the spatial middle of the fingerprint image (where the core is now situated). According to the size of the cells and the number of cells for each print, the number of cells produced by the tessellation differs.

The last step is the approximation of each cell through a triplet (as described in 4.3). The most suitable cell candidate for each cell will be determined and used to represent it.

4.6. Comparison of Triplet Sets

In the following chapter three different comparison methods will be presented together with results for tests of each.
4. Pattern Matching Using Co-Sinusoidal Triplets

4.6.1. Direct Comparison Method (DCM)

There are various approaches on how to compare two co-sinusoidal triplet sets. One way is to directly compare one cell of a probe with the corresponding cell of the reference. The distance is calculated by using the difference between each parameter of each cell of the biometric reference with the ones from the biometric probe. So for example the angle of propagation $\theta$ of the first cell in the first row of the probe will be compared to the angle of propagation of the first cell in the first row of the reference. The difference of these values will be divided by the maximum possible distance for this parameter and then subtracted from 1, so the result is a similarity score between 0 and 1.

The density of the similarity scores can be plotted for genuine and impostor comparisons as described in [BCP+03]. The resulting genuine and impostor score distribution can be seen in Figure 4.8. The data was raised by using the FVC 2000 DB1 which consists of 800 individual prints where 717 could be processed by VeriFinger. Images which are below the quality threshold of VeriFinger could not be processed due to the lack of a reference point. The whole set consists of a total of 800 fingerprint images from 100 different users with 8 sessions for each. The test was conducted with the cell size of $8 \times 8$ and a precision of 5, 4 and 5 bits for theta, lambda and gamma as proposed by [Vs06].

When looking at the plot it can be deducted that this approach does not give enough information to successfully distinguish between an impostor and genuine comparison without excluding a high percentage of genuine access attempts. Therefore, it is not possible to specify a threshold that allows for a useful distinction. In order to draw the right conclusion the test has to be widened to a different range of precisions, cell sizes, and tessellation but a first deduction suggests to use a different, more complex comparison approach.

4.6.2. Weighted Direct Comparison Method (WCM)

A more complex approach can have different weights depending on the significance of the parameter and its probability to be affected by noise. According to [Vs06] the significance of each parameter varies very strongly. He showed that for example the frequency part of a triplet is less substantial than the orientation of the cell.

$$\text{score}(i,j) = \Gamma_\theta \cdot \text{score}_\theta(i,j) + \Gamma_\delta \cdot \text{score}_\delta(i,j) + \Gamma_\lambda \cdot \text{score}_\lambda(i,j)$$
Figure 4.8.: The comparison scores for the first approach. The density on the Y-Axis is given as a relative frequency. The similarity score on the X-Axis is given as a value between 0 and 1. The genuine and impostor approaches are superimposed.
4. Pattern Matching Using Co-Sinusoidal Triplets

4.6.3. Neighborhood Comparison Method (NCM)

A typical problem with the direct matching is, that occasionally a cell produces a match, even though its surrounding neighborhood does not match and vice versa, therefore introducing errors to a homogeneous area (see Figure 4.9). In order to enhance the accuracy of the results further, the neighborhood around each cell is taken into account as well. When considering for example the orientation part $\theta$ of the triplet in a vector field, a single misaligned cell will be surrounded by a neighborhood of matching cells. Therefore, a decision about the matching of a cell is not solely based on each individual cell, but the cell plus the surrounding N8 neighborhood reducing the error due to the presence of outliers and noise.

![Figure 4.9: Neighborhood problem](image)

Figure 4.9.: Neighborhood problem: Where the homogeneous area in the left is correctly captured, an error is introduced in the probe that will lead to a non-match at this position.

As can be seen in Equation 4.3, for the comparison of the angle of propagation $\theta$, the neighborhood at the point $\theta(i,j)$ is considered and if enough surrounding cells match as well, an extra weight is added to stabilize the decision.

$$
\theta_{\text{neighborhood}}(i,j) = \sum_{u=-1}^{i+1} \sum_{v=-1}^{j+1} \begin{cases} 
1 & \text{if } |\theta_{\text{reference}}(u,v) - \theta_{\text{probe}}(u,v)| \leq \Gamma_{\theta_1} \\
0 & \text{if } |\theta_{\text{reference}}(u,v) - \theta_{\text{probe}}(u,v)| > \Gamma_{\theta_1}
\end{cases}
$$

$$
\theta_{\text{score}}(i,j) = \begin{cases} 
2 & \text{if } |\theta_{\text{reference}}(i,j) - \theta_{\text{probe}}(i,j)| \leq \Gamma_{\theta_2} \land \theta_{\text{neighborhood}}(i,j) > \Gamma_{\theta_3} \\
1 & \text{if } |\theta_{\text{reference}}(i,j) - \theta_{\text{probe}}(i,j)| \leq \Gamma_{\theta_2} \\
0 & \text{if } \theta_{\text{neighborhood}}(i,j) > \Gamma_{\theta_3} \\
-1 & \text{else}
\end{cases}
$$

Where $\Gamma_{\theta_1}$ is the threshold used to determine if each individual neighboring cell matches, $\Gamma_{\theta_2}$ is the threshold used to determine if the currently investigated cell matches, and $\Gamma_{\theta_3}$ is the threshold used to determine if the neighborhood of $\theta(i,j)$ as a whole matches.
4. Pattern Matching Using Co-Sinusoidal Triplets

The approach to calculate the absolute difference between the neighboring cells is also applied to the wavelength parameter $\lambda$. If enough of the surrounding area matches, $\lambda(i,j)$ gets an extra weight (see Equation 4.4).

$$
\lambda_{\text{neighborhood}}(i,j) = \sum_{u=i-1}^{i+1} \sum_{v=j-1}^{j+1} \begin{cases} 
1 & \text{if } |\lambda_{\text{reference}}(u,v) - \lambda_{\text{probe}}(u,v)| \leq \Gamma_{\lambda 1} \\
0 & \text{if } |\lambda_{\text{reference}}(u,v) - \lambda_{\text{probe}}(u,v)| > \Gamma_{\lambda 1}
\end{cases}
$$

$$
\lambda_{\text{score}}(i,j) = \begin{cases} 
2 & \text{if } |\lambda_{\text{reference}}(i,j) - \lambda_{\text{probe}}(i,j)| \leq \Gamma_{\lambda 2} \land \lambda_{\text{neighborhood}}(i,j) > \Gamma_{\lambda 3} \\
1 & \text{if } |\lambda_{\text{reference}}(i,j) - \lambda_{\text{probe}}(i,j)| \leq \Gamma_{\lambda 2} \\
0 & \text{if } \lambda_{\text{neighborhood}}(i,j) > \Gamma_{\lambda 3} \\
-1 & \text{else}
\end{cases}
$$

(4.4)

Where $\Gamma_{\lambda 1}$ is the threshold used to determine if each individual neighboring cell matches, $\Gamma_{\lambda 2}$ is the threshold used to determine if the currently investigated cell matches, and $\Gamma_{\lambda 3}$ is the threshold used to determine if the neighborhood of $\lambda(i,j)$ as a whole matches.

According to Vannfält and Åström [Vs06], unlike the orientation and frequency, the phase is shifting very rapid which makes absolute differences between cells unsuitable. This was supported by additional tests (see Figure 4.10 where a direct comparison of similar triplets produces reliable results in the orientation and wavelength, but fails for the phase. Instead of the comparison of absolute values, relative differences will be used which are based on the difference of each neighboring cell to the center.

$$
\delta_{\text{neighborhood}}(i,j) = \sum_{u=i-1}^{i+1} \sum_{v=j-1}^{j+1} \begin{cases} 
1 & \text{if } ||\delta_{\text{reference}}(i,j) - \delta_{\text{reference}}(u,v)|| - ||\delta_{\text{probe}}(i,j) - \delta_{\text{probe}}(u,v)|| \leq \Gamma_{\delta 1} \\
0 & \text{if } ||\delta_{\text{reference}}(i,j) - \delta_{\text{reference}}(u,v)|| - ||\delta_{\text{probe}}(i,j) - \delta_{\text{probe}}(u,v)|| > \Gamma_{\delta 1}
\end{cases}
$$

$$
\lambda_{\text{score}}(i,j) = \begin{cases} 
1 & \text{if } \delta_{\text{neighborhood}}(i,j) \geq \Gamma_{\delta 2} \\
0 & \text{if } \delta_{\text{neighborhood}}(i,j) < \Gamma_{\delta 2}
\end{cases}
$$

(4.5)
4. Pattern Matching Using Co-Sinusoidal Triplets

Figure 4.10.: The reference is compared with the probe with direct comparison for each parameter. The differences for each cell are shown in the bottom three images. Where a black cell is equivalent to a small difference between the reference and the probe at this position for the specific parameter.

4.7. Cell Size Benchmark

In order to investigate the influence of the cell size, the False Match Rate (FMR), the False Non Match Rate (FNMR) and the Equal Error Rate (EER) for cell sizes in the range of $5 \times 5$ pixels up to $18 \times 18$ pixels were studied. The comparison algorithm used is based on the similarity of the cell triplets in the reference and the probe. Each cell triplet of the probe will be compared to the corresponding cell triplet in the reference.

After conducting a research on the dimension for each quantized co-sinusoidal triplet, the following results were discovered. The acquisition is based on the
FVC 2000 database DB1_A. The results show the EERs for each cell size. The full evaluation can be found in the Appendix (B). The full evaluation consists of a score distribution, a Receiver Operating Characteristic (ROC) curve and the Equal Error Rate (EER) for the corresponding cell size over the whole dataset. The analysis was conducted for each individual parameter and thus results in three score distributions, ROC curves and EERs for each cell size. For comparison, the VeriFinger performance based on minutiae matching and the implementation of the Xu/Veldhuis approach from Martin Olsen are included as well.

For the tests, all images were preprocessed by the VeriFinger SDK 6.0. Different image enhancement filters followed by a binarization led to results as those in Figure 4.11. Additionally the orientations and positions of the cores were extracted in order to overcome translation and orientation problems. Images with no cores present and images that VeriFinger could not process were excluded from the test. Therefore, not the whole set of the databases could be used. The chosen bit-depths were 5 for the angle of propagation $\theta$, 4 for the wavelength $\lambda$ and 5 for the phase $\delta$.

As can be seen in Figure 4.12, the cell size that results in the least average error rate and therefore the optimal cell size for the FVC dataset DB1_a is 16.

4.8. Parameter Evaluation

After selecting the most suitable cell size (in this case 16 x 16) depending on the resolution of the sensor, the influence of the different parameters on the
4. Pattern Matching Using Co-Sinusoidal Triplets

unambiguity was investigated. The EERs for each parameter were compared to the EERs of all different combinations of the parameters. The entire outcome of this can be found in Appendix (A).

<table>
<thead>
<tr>
<th>Parameter Combination</th>
<th>EER</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \theta )</td>
<td>0.1803</td>
</tr>
<tr>
<td>( \lambda )</td>
<td>0.2379</td>
</tr>
<tr>
<td>( \delta )</td>
<td>0.1639</td>
</tr>
<tr>
<td>( \theta ) and ( \lambda )</td>
<td>0.1946</td>
</tr>
<tr>
<td>( \theta ) and ( \delta )</td>
<td>0.1691</td>
</tr>
<tr>
<td>( \lambda ) and ( \delta )</td>
<td>0.2036</td>
</tr>
<tr>
<td>( \theta, \lambda ) and ( \delta )</td>
<td>0.1886</td>
</tr>
</tbody>
</table>

Table 4.4.: Influence of the three parameters on the equal error rate. The data of FVC2000 DB1_A was used to conduct the evaluation.

The summary of this examination is shown in table 4.4. The single use of \( \delta \) provides the lowest equal error rate. The most fruitful combination is between \( \theta \) and \( \delta \), where the equal error rate is only slightly above the use of \( \delta \) alone. The use of two parameters instead of one can help to stabilize the results when employed on different datasets, therefore a combination of \( \theta \) and \( \delta \) is suggested.
4. Pattern Matching Using Co-Sinusoidal Triplets

Figure 4.13.: The Gauss Bell will increase the weight for values close to the middle and decrease the weight for values further away from it.

4.9. Algorithm Improvements

When running the tests, it was noticeable that the triplet creation algorithm takes a considerable time to process. This is mainly due to the problem that the ISO standard [FS06b] suggests to compare each cell with all possible cell candidates. The algorithm can be speed up by calculating the cell candidates upfront and keeping them in the memory for the usage of the triplet creation.

To further improve the speed, the complexity for the searching of the most suitable triplet can be reduced. This is achieved by calculating a orientation image for the fingerprint like described in [MMKC87]. With the help of the orientation image it is possible to predict the orientation value of a cell. This is done by averaging all orientations of ridge pixels in a cell. There are two ways how the search for the optimal triplet can be enhanced. The first option is to simply round the orientation value to the nearest orientation quantization value and then search through all combinations of $\lambda$ and $\delta$ for this orientation. The second option is to round the orientation down and up to the next orientation quantization values and look for both orientations through all combinations of $\lambda$ and $\delta$. This limits the search to $2^{l+m+n}$ or $2^{2^m+n}$ instead of $2^{l+m+n}$ where $l$, $m$ and $n$ is the bit depth of $\theta$, $\lambda$ and $\delta$.

Another conspicuousness when analyzing the results, especially when looking at the synthetic prints was, that the disparity of two genuine triplet sets increased with increasing distance from the core. As a result, the comparison of two genuine triplet sets can be enhanced by using a Gauss Bell (see Figure 4.13. This leads to a stronger weight for values closer to the core and a weaker weight for values at the corners and borders of the triplet set. However, this will also affect the comparison of impostors as well and may lead to a higher similarity score for them too. It needs to be evaluated whether the effect on the similarity is beneficial for the comparison.
5. Biometric Template Protection

5.1. Introduction

When using a biometric fingerprint system the biometric feature (template) representing a fingerprint needs to be somehow digitally stored. Traditional systems, like the ones described in chapter 1, store them unencrypted in a database. Therefore, anyone with access to the database is able to gain access to the fingerprint data stored for each user. The Chaos Computer Club [Clu10] has proven that with the help of this data it is possible to create a fake fingerprint which can be used for profiling (for example medical conditions can be extracted) or to gain access to the biometrically secured asset. As template forgery being the security risk, the involved privacy risk is not just identity theft, but also profiling and non-cooperative usage. For example an attacker is able to link information of different databases together with the one piece of information that undeniably identifies him as one and the same person, the biometric characteristic. An example for non-cooperative usage is tracking or surveillance using biometrics (like it was tested at Mainzer Hauptbahnhof\(^1\)).

The basic principle of the problem can be compared to a regular login system, like any e-mail system. The password for a user is not stored in plain text in a database, but rather as an encrypted hash of the word originally used as the password. By using a mathematical one-way function, like a hash function, a person with access to the database, may it be an attacker or a legitimate user of the system, will not be able to retrieve the password from the hash. This is caused by the strong causality underlying the hash function, where slightly different input values lead to completely different output values.

The requirements for secure biometric templates are defined in [BBGK08], whereof the main demands are the following:

*Protected Templates:* It shall be impossible to retrieve or decode the original biometric sample from the secure biometric features,

\(^1\)http://www.heise.de/newsticker/meldung/Foto-Fahndung-im-Mainzer-Hauptbahnhof-169943.html
5. Biometric Template Protection

Revocable: As opposed to passwords, biometric characteristics cannot be changed or altered if necessary due to the nature of being a behavioral or, as in the case of fingerprints, biological characteristic of a person. Once a biometric template is compromised it cannot be simply renewed. Therefore, one of the biggest demands for biometric templates is the ability to renew or revoke them. When incorporating this requirement, cross reference matching is also suppressed since it allows different protected templates from the same characteristic, ultimately leading to diversification.

Interoperability: The biometric system shall be based on a predefined format and method that should be compatible with a wide range of sensor types and feature extraction components.

The most practical procedure is a two-stage approach where at first a biometric sample is converted to the preexisting biometric feature data formats and then the data is converted to a protected template.

In order to prevent the misuse of biometric features, they need to be encrypted to facilitate security even when retrieved by an attacker. The aim of BTP is to have a similar performance in comparison to traditional biometric systems that do not employ BTP. The performance does not need to be better and can be slightly worse, since the gain in security through protected templates can outweigh the loss in performance. The following chapter will introduce the most renowned template protection schemes and describe the Helper Data Scheme, which was implemented, in detail.

Why standard cryptography is not enough
Any cryptographic system has to face the threats of disclosure and tampering. Biometric systems however enclose additional threats that endanger the security of the biometric templates. First and foremost there is the possibility of cross reference matching or cross comparison, which establishes unwanted links between different applications or different databases. As stated above, biometric characteristics are unchangeable and cannot be revised. So once compromised, the characteristic is compromised forever. With standard encryption systems, all biometric characteristics can be recovered if the databases secret key is compromised.

The biggest problem of the combination of cryptography and biometrics is the difference of what cryptographic parameters should be and what biometric templates are. Biometric data is fuzzy, continuous and to a high degree uncertain, whereas cryptographic parameters need to be discrete and demand zero uncer-
tainty. Biometric templates are at best similar but never equal. Therefore, the requirements to cryptography for biometric template protection as developed by [BBGK08] are as follows:

**Secrecy:** Biometric templates shall be stored encrypted in the database without the need to be decrypted for comparison.

**Diversifiability:** It shall be possible to derive different protected biometric templates from the very same characteristic.

**Noise-Robustness:** Due to the high noise that occurs when acquiring a biometric template, the cryptographic system needs to allow a certain degree of variation in the comparison.

**Privacy Preserving:** It shall be impossible to retrieve data about the original biometric sample from the binary string representation. Furthermore, it shall prohibit leakage conjugation, i.e., typical weak spots of certain biometric characteristics shall not be revealed.

### 5.2. Template Protection Schemes

The computation of cryptographic keys from biometric data was first proposed by Bodo [Bod94] in 1998 and evolved over the time according to the recent constraints. Today there are numerous different template protection schemes in research like Pseudo Identities, Cancelable Biometrics, Fuzzy Vault, Fuzzy Extractors, Helper Data Scheme, Fuzzy Commitment, Biometric Encryption, Shielding Functions, Secure Syndrome, Biocryptics, Extended PIR or Secure Sketch to name but a few. All of these schemes are represented in a consistent architecture, the ”Reference Architecture for Biometric Template Protection based on Pseudo Identities” [BBGK08].

**Reference Architecture**

The reference architecture as described in [BBGK08] is technology neutral, thus providing a generic framework for biometric template protection. It is designed to facilitate the requirements described in Chapter 5.1. The reference architecture is based on so-called pseudo identities (PI). PIs are diversifiable, protected identity verification strings. A PI does not reveal any information about the original biometric measurement data, the biometric template or the identity of its owner and does not allow retrieval of such. The PI life cycle can be defined in 4 steps.
5. Biometric Template Protection

Creation: First, a biometric capture device retrieves a biometric sample (like a fingerprint sensor acquiring a fingerprint image). Then a feature extractor creates biometric features from the captured sample (like QCT). In the last step, a pseudo identity encoder (PIE) generates a pseudo identity and if necessary, additional auxiliary data (AD). The PIE takes supplementary data (SD) as an input which is used for security enhancement. The SD can be a possession or knowledge-based secret by the user or a system specific secret. The SD is not stored with the template.

As an output of the PIE, the combination of PI and AD is the protected template. There are three different ways how to store the PI and AD, each providing different benefits and drawbacks. They are: central storage, local storage and hybrid storage. Central storage keeps both, PI and AD in a database, whereas local storage keeps both on a token. Hybrid storage keeps either one in the database and the other one on a token. The big advantages that arise with hybrid storage are blacklist and audit functionalities as well as easy revocation.

![Reference Architecture based on Pseudo Identities](image)

Figure 5.1.: Reference Architecture based on Pseudo Identities. This Figure shows the PIR approach.

Verification: There are two different classes of verification processes with only subtle differences. The Pseudo Identity Recorder (PIR) approach and the Pseudo identity Verification approach. The PIR approach is based on recreation of a
5. Biometric Template Protection

A new PI* is created by going through the three phases of triplet creation (biometric sample retrieval, feature extraction, PI encoding). The newly created PI* is transferred to a Pseudo Identity Comparator (PIC) which compares the PI* to the PI that is stored in the database and if and only if they are exactly equal, the result of the verification is a match. An overview of this approach can be found in Figure 5.1.

The Pseudo Identity Verifier (PIV) approach is not based on recreation of a PI* but rather on direct verification of a PI based on a provided recognition sample. The PIV takes the protected template, a sample from a biometric sensor and the SD as an input and provides a verification result. The benefit of this technique over the PIR is that no exchange of template information is required when the PIV and the protected template are implemented on the same device.

Expiration: The expiration of a protected template can be caused by different reasons, where the most prominent one is a compromised template. Apart from that, the system may require renewal of the reference from time to time, for example due to the impact of aging effects (like face recognition).

Revocation: The revocation of a biometric reference is as easy as deleting the PI from the database. If the subject is still a legitimate user of the system, a new biometric reference has to be created (reenrollment).

5.3. Helper Data Scheme

The Helper Data Scheme is based on the reference architecture described in Chapter 5.2 and consists of two different phases. In order to be able to use the system, a person has to be enrolled first. The work flow of the Helper Data Scheme can be seen in Figure 5.2 for the enrollment and verification.

Enrollment
When enrolling, the biometric sensor will take multiple samples and the feature extractor will obtain biometric features of constant length for each sample.

In the binarization step, the average feature vector for the whole population of enrollees $\mu$ is calculated and through the case differentiation 5.1 with the created feature vectors combined.

$$Q(X(i, j)) = \begin{cases} 0, & \text{if } X(i, j) \leq \mu \\ 1, & \text{if } X(i, j) > \mu \end{cases}$$

(5.1)
5. Biometric Template Protection

$X_{i,j}$ is the extracted feature $j$ for the selected user $i$ where $j$ ranges from 1 to $M$, and $M$ is the number of feature vectors. This will create a bit matrix (QBV) for each feature vector with the information if the current position of the feature vector was above or below the average.

The Reliable Bit Selector (RBS) will then select the $k$ most stable features for this enrollee according to the multiple feature vectors that were extracted. $k$ is the length of the codeword and the most stable positions for this enrollee. A position is called stable if a feature for all sessions is equal. For users with less than $k$ reliable components, soft components can be defined, where less than all sessions are equal. These $k$ stable components are called Auxiliary Data 1 (AD1) and can be stored unencrypted in the database. The order of the positions can also be randomized, making it harder to gather information from AD1. Then the RBS selects the most reliable components, RBV, from the QBV.

Meanwhile, a Random Number Generator (RNG) will create a Secret Binary Vector (SBV). First, a mathematical one-way function (Hash Function) will create an encrypted version of the SBV that will be stored in the database as the PI. Additionally, the SBV will be combined with an Error Correcting Code (ECC) that will be used to add redundancy to correct possible errors which will be introduced by the biometric sample. The resulting Code Binary Vector (CBV) will be selected to have the same length as the RBV. The CBV and RBV will be combined with an XOR and the result will be stored in the database as Auxiliary Data 2 (AD2). Through the combination with an XOR, it is impossible to retrieve neither the secret in the CBV nor the biometric information of the sample without the knowledge of one or the other.

**Verification**

For the verification, a new (single) biometric sample is taken from a sensor and the features will be extracted. In the binarization step of the verification, the real-valued feature vectors will be compared against the mean value of the whole population and a binarized bit vector QBV' is passed to the RBS. The RBS of the verification will then select the most reliable bits, RBV', of QBV' with the help of AD1 from the database. Afterwards, the RBV' will be combined with AD2 from the database with an XOR, leading to CBV'. The ECC will then take care of potential bit errors which might be introduced in the noisy fingerprint images. The result of the ECC, SBV' will then be converted to PI' through the hash function. If and only if PI' is consistent with the PI from the database, the result of the comparison is a match. If PI' and PI are different in any element, it will result in a non-match for this access attempt. Additional to the security in the database, the binary outcome of the comparison has the advantage to prevent hill-climbing attacks [MDFAAF*06].
5. Biometric Template Protection

Revocation
When a protected template is revoked, a new SBV needs to be created and combined with one new feature vector. From that, AD2 and PI will be recalculated. Is it not necessary to recalculate the reliable positions for the same biometric characteristic.

5.4. Error Correcting Codes

In coding theory, Error Correcting Codes (ECC) are techniques that allow reliable transfer of data over an error prone medium. ECCs deal with error detection and error correction, where error detection is limited to the exposure of existing errors, whereas error correction is able to reconstruct the original signal when errors occurred. Error detection is indispensable in any form of digital data transfer and storage. Error detection and correction is achieved by adding redundant information to the useful information which will be used to determine that an error occurred and, in error correction, where it occurred. Error correction can be handled in two ways, with Automatic Repeat Request (ARQ) or Forward Error Correction (FEC). ARQ is a technique where a detected error in a transmitted message leads to retransmission of the whole message. It is therefore also called backward error correction. A FEC is present when the sender already adds information to correct possible errors to a message. Hybrid approaches exist, where
5. Biometric Template Protection

Small error correcting capabilities are encoded in the message and if greater errors are detected than can be fixed, an ARQ will be sent. Since in biometric applications errors are already introduced when the sample is taken, an FEC approach is the only way to allow correction of existing noise. There exist dozens of FECs, for example Constant-Weight Code, Convolutional Code, Group Codes, Golay Codes, Hagelbarger Code, Hamming Code, Online Code, or BCH. The BCH (Bose-Chaudhuri-Hocquenghem) codes are widely used in academics and are forward error correcting codes for 1-bit errors used to correct around 25% of the total codeword length. The advantage of BCH Codes above others is that they can be decoded via syndrome decoding and the high variability for selection of the codeword length and correcting capabilities. BCH Codes can be classified as general and simplified BCH Codes. The mathematical definition, in pursuance of [Gil10], is as follows

BCH Codes are cyclic codes that make use of field theory and polynomials of that field. $GF(q)$ is defined as the finite field, where $q$ is a prime power. The codes are defined by a $(d - 1) * n$ check matrix over $GF(q^m)$. The length of the useful information encoded by the BCH is defined as $n$. Furthermore, $a$ is an element of $GF(q^m)$ of order $n$, $b$ is an integer between $0 \leq b < n$ and $d$ is an integer within the range $2 \leq d \leq n$. A polynomial code over $GF(q)$ can be generated that has the codeword length $n$ and the minimal hamming distance of at least $d$.

$$H = \begin{bmatrix}
1 & a^b & a^{2b} & \cdots & a^{(n-1)b} \\
1 & a^{b+1} & a^{2(b+1)} & \cdots & a^{(n-1)(b+1)} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & a^{b+d-2} & a^{2(b+d-2)} & \cdots & a^{(n-1)(b+d-2)}
\end{bmatrix} \quad (5.2)$$

The BCH Code is called primitive if the code is using only a primitive $a$, which limits the maximum possible block length to $n = q^m - 1$. An excerpt of possible codeword length and sizes of the polynomial can be found in Chapter 5.6.1 in Table 5.1.

5.5. Data Preprocessing

The HDS demands that in a verification two fingerprint datasets (feature vectors) can be compared by directly comparing each element of the probe feature vector with each element of the reference feature vector. The verification of QCTs, as explained in Chapter 4, is not just based on direct matching, but rather on more complex methods including relative neighborhood comparison. This signifies, that
the values created by the quantized co-sinusoidal triplets method need to be converted to a utilisable format that allows direct comparison.

Converting the angle of propagation $\theta$: With the method of co-sinusoidal triplets, the similarity of two angles is restricted. When considering a ridge orientation as a vector, it is clear that the orientation of the vector is irrelevant when approximating the cell with the co-sinusoidal triplet. An angle of $180^\circ$ is equivalent to the angle of $0^\circ$, so is $181^\circ$ equivalent to $1^\circ$, $182^\circ$ to $2^\circ$, and so forth. Additionally, a maximum dissimilarity is reached at $90^\circ$ and everything above is increasing the similarity of the two angles. In order to allow the direct match as described above, the absolute value of the sine of $\theta$ is used resulting in: $|\sin(\theta)|$

In order to consider the surrounding neighborhood, an average of the N8 neighborhood around each cell is calculated and used as the value of the center. This should lower the impact of noise and outliers in a homogeneous area and will have a blurring effect on edges. At the borders of the fingerprint where no complete N8 neighborhood is present, only the available neighbours are used to determine the average. The formula is as follows:

$$\text{Cell}(i,j) = \sum_{u=i-1}^{i+1} \sum_{v=j-1}^{j+1} \theta_{\text{reference}}(u,v)/9$$

Converting the frequency $\lambda$: Same as the angle of propagation $\theta$ the neighborhood of $\lambda$ is calculated as an average of the surrounding N8 neighborhood.

$$\text{Cell}(i,j) = \sum_{u=i-1}^{i+1} \sum_{v=j-1}^{j+1} \lambda_{\text{reference}}(u,v)/9$$

Converting the phase $\delta$: The similarity of the phase, as calculated in chapter 4.6.3 is dependant on the relative distance of the neighborhood to its center. Therefore the conversion of the phase is slightly different to the calculation of the calculation of the parameters above.

$$\text{Cell}(i,j) = \sum_{u=i-1}^{i+1} \sum_{v=j-1}^{j+1} |\delta_{\text{reference}}(i,j) - \delta_{\text{reference}}(u,v)|/8$$

When the three parameters plus the three neighborhoods have been converted, they are concatenated to a single feature vector of the length (number of cells) * 6. Further improvements to the HDS input data will be explained in the following Chapter 5.6.
5. Biometric Template Protection

5.6. Tests

5.6.1. BCH Code Triplets

One of the adjustment parameters of the HDS can be found in the ECC. The BCH Codes, that were chosen for the ECCs allow to choose the length of the codeword \( n \) and the size of the BCH polynomial \( k \) (also called secret). A codeword is the combination of the payload plus a block of check bits. Depending on the selected parameters, the BCH code can cover a limited amount of 1-bit errors. The size of the polynomial for a given codeword length has a direct impact on the error correcting capabilities. The smaller the polynomial, the less payload can be encoded but the more errors can be corrected. The bigger the polynomial, the more payload can be added to the codeword but the less errors can be corrected. For the implementation of the BCH codes, the BCH Encoder and Decoder of the Matlab Communications Toolbox were used. An excerpt of all possible valid combinations of BCH Codes can be found in Table 5.1. The codeword length can be calculated from \( 2^x - 1 \), where \( x \) is arbitrary. The number of correctable errors is derived from the length of the codeword and the size of the polynomial.

<table>
<thead>
<tr>
<th>codeword length ( n )</th>
<th>polynomial size ( k )</th>
<th>correctable errors ( t )</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>15</td>
<td>11</td>
<td>1</td>
</tr>
<tr>
<td>15</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>15</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>31</td>
<td>26</td>
<td>1</td>
</tr>
<tr>
<td>31</td>
<td>21</td>
<td>2</td>
</tr>
<tr>
<td>31</td>
<td>16</td>
<td>3</td>
</tr>
<tr>
<td>31</td>
<td>11</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 5.1.: An excerpt of the valid combinations for BCH codes.

In order to find the most suitable BCH code combination, a statistical analysis over the FVC 2000 DB2\_a and FVC 2002 DB2\_a database was conducted. As already mentioned in Chapter 4 only a part of the dataset could be used (656 prints in FVC 2000 DB2\_a and 730 prints in FVC 2002 DB2\_a) due to extraction
5. Biometric Template Protection

problems of VeriFinger. This lead to roughly 7 sessions per user (originally 8). All users were enrolled with the HDS approach where 5 sessions per user have been used for the statistical analysis of the stable components and the remaining 2 sessions were used for identification. All feature vectors have been encoded (and later decoded) with all possible combinations of BCH codes from a codeword length of 7 up to 511. The results of this test can be seen in Figure C.1.

It was observed, that the majority (90.16%) of the scatter plot is located at the y-axis which is equivalent to a FMR of 0. Since the FMR is a system critical criterion, it is desirable to have it as low as possible. When searching for a good codeword length, the assumption is made that the best codewords will be found where the FMR is 0. The codeword lengths with the lowest FNMRs for the fixed FMR of 0 were retrieved for each cell size. The codewords that led to the three lowest FNMRs for each cellsize were selected and plotted against each other as can be seen in Figure C.2. The most stable codeword length for any given cell size is 63. This codeword length was present in 43% of all the lowest three FNMR values. Of course the worst case has to be observed as well. Thus, the three codeword lengths that produced the highest FMRs for each cell size were examined and the results can be seen in Figure C.3. The results with the worst FMRs consist predominantly of low codeword lengths (66.66% for codeword lengths 7 and 15).

Of course the codeword length itself is not the only parameter of the BCH code. The size of the polynomial also has to be considered and therefore the results were further investigated. As before, the polynomial sizes that led to the lowest FNMRs with the FMR fixed at 0 were considered to provide the best results. They were also calculated for all cellsizes between 1 and 14. The best results are shown in Figure C.4 and the worst in Figure C.5. Polynomials of small size proved to be unfeasible, which is expected since they do not manage to store enough unique information about a fingerprint. It was observed over the whole field of BCH codeword lengths, that polynomial sizes \( \leq 5 \) are accountable for around 66% of the worst FMR to FNMR rates. All of the three worst results for each cellsize had a polynomial size of \( \leq 10 \). The polynomial sizes that caused the best results were between 15 and 18, which were responsible for 66.66% of the best results.

5.6.2. N-Tuple Combination

In order to find the parameters that are most feasible, all combinations of the three QCT parameters as well as their neighborhoods were investigated. This way it is possible to examine which parameters are stable and which parameters have a high fluctuation. The HDS automatically selects parameters with the lowest
5. Biometric Template Protection

intra-class variability and the highest inter-class variability as part of the biometric template. This means that a position in the feature vector that is measured with the same amplitude on different sessions of the same user and at the same time has a high variance between different users is preferential to a feature that deviates from this definition. The position is considered stable if fulfills the above definition.

In order to evaluate the significance of the different parameters, $\theta$, $\lambda$, $\delta$, the $\theta$ neighborhood, the $\lambda$ neighborhood, and the $\delta$ neighborhood were concatenated and used as a single feature vector. After the HDS selected the most suitable positions in the feature vector, the positions can be projected back to the original parameters. As can be seen in the result of this test in Figure C.6, the $\theta$ parameter proved to be most reliable. In fact, when looking at the exact numbers the two most stable positions are without exception always $\theta$ parameters. As explained in Chapter 4 the delta parameter is unsteady and is hence the least stable. The fact that the neighborhoods cover an area and are combined to a single value in the preprocessing step makes them an averaging function for the surroundings of a cell. Naturally, an averaging function projects a wide range to a narrow range which leads to a lower inter-class variability of the neighborhood parameters. This explains why the neighborhoods are not considered the most stable positions.

Two practical problems arise with these results that need to be considered. First, when the HDS is scanning for stable positions and finds two equally stable positions, the one that is found first in the feature vector will be considered more stable. For example all positions for $\theta$ have the same maximum possible intra-class variability (between 0 and 1), so it is very likely that two or more equally stable positions exist. This problem is existent for all parameter groups. As a result, the order of the parameters in the feature vector is important.

The second problem is, that positions that do not contain any fingerprint information (areas without ridges), are considered very stable since they have the maximum intra-class variability and typically a high inter-class variability. In order to cope with this problem, the helper data will be rearranged after the HDS selected the stable positions. Positions that do not contain fingerprint information will be pushed back to the end of the helper data, thus making them the least stable ones. The results of this is shown in Figure C.7. As can be seen, previously stable positions of $\theta$ and $\lambda$ moved to the end, improving the least stable $\theta$ and $\lambda$ values by more than 100%.
5.6.3. Optimum Cell Size

When using the HDS, it is conceivable to calculate and concatenate the QCT parameters for multiple cell sizes to one single, extensive feature vector. The HDS will automatically take care of the selection of the most stable positions over different cell sizes. If there is enough information in one cell size though, the processing performance (not the biometric performance) can be increased by generating only the information necessary. To investigate if there is a cell size that stands out enough to rely only on one specific size, all parameters of all cell sizes were concatenated and used as a feature vector for the HDS. The resulting stable positions were projected onto the respective cell size and the occurrences of a singular size at the top positions gave information about the existence of such. For the test, the 6-tuple for the cell sizes from 5 to 17 were calculated and concatenated, leading to a feature vector with around 30,000 positions for each user. Figure C.8 shows the results for this test without removal of cells that do not contain any fingerprint information and Figure C.9 shows results after the elimination of cells without information. The test shows that no specific cell size is advantageously selected for having the most stable cells and therefore the cell size that produced the lowest FNMR at the FMR zero was chosen as the best cell size.

5.6.4. Layer-Based Order

As already discovered in Chapter 4, the most stable cells for the QCT are situated around the core of a fingerprint. This knowledge was verified for the HDS (see Figure 5.3) and can be used to improve the search for stable cells. It was implemented by rearranging the input data for the HDS. The Order in which the cells were selected can be seen in Figure 5.4. The innermost cell will be placed in front of the feature vector, moving outside layer by layer, the cells will be appended to the new feature vector. The order on a layer is from top to bottom and from left to right. This means that the leftmost column of the current layer is appended first, then the top entry followed by the bottom entry of the next column. This is repeated until the rightmost column is appended. Then the dimension of the layer is extended by one cell in each direction and the cells will be appended in the previously mentioned order.

fig:stableCells
5.6.5. Multiple Verification Samples

The results found in 5.6.1 are in no sense comparable to the verification output of state of the art verification algorithms that do not employ BTP. To improve the performance, one idea is to use more than just one verification attempt. The user provides the biometric system with a series of probes and if one of the probes match, the user is accepted. It is supposable that the fingerprint sensor takes a continuous shooting while the users places the finger on the sensor. Then the whole set of fingerprint images is examined and compared to the reference. This
5. Biometric Template Protection

Figure 5.4.: Rearranging the QCT, the cells in the middle will be copied to the first positions of the feature vector, going layer by layer to the outside cells. The order is from top to bottom and from left to right.

was tested by using 3 or 5 of the 7 sessions per user for enrollment and later use 4 or 2 of the sessions for one verification. This allows a maximum performance boost by 100\% of the prior accepted users (when 5 sessions were used for enrollment). Therefore, if the FNMR is at around 70\%, it can be lowered to a maximum of 40\% FNMR when using 2 sessions for enrollment. In worst case there will be no change in the results at all. A thing that should be kept in mind is, that this is only valid if the FMR is exactly zero. Otherwise it will increase all impostor attempts, leading to a much higher FMR. Since the selected BCH Code produces results for both FVC databases (FVC2000 DB2 a and FVC2002 DB2 a) with zero FMR, this is a sound approach. The results (see Figure C.10 and Figure C.11) illustrate the decrease of the minimum FNMR from 68.83\% to only 62.34\%.
6. Conclusion

The QCT approach was implemented and the biometric performance was compared to state of the art system that do not employ BTP (namely NeuroTechnology’s VeriFinger software and Spectral Minutiae). Different comparison methods were implemented to further deepen the understanding of biometric fingerprint verification and the necessities of a successful fingerprint verification system. The features produced by the QCT were converted to a fixed-length feature vector which is mandatory to employ BTP. The HDS was implemented as the BTP system and the results investigated and documented. Since there was only very limited literature available on the QCT approach, the requirements had to be established and tested first. The biometric performance of the HDS is significantly dependent on solid fixed-length feature vectors. The biometric performance of the commercial software VeriFinger, which is based on the traditional minutiae detection (see Chapter 2.1), could not be reached in the tests with the QCT. The results were slightly below the spectral minutiae approach (refer to Appendix B) which were a solid foundation for the HDS. The challenges that arise with the HDS lie in the trivial comparison method for two feature vectors. The more complex comparison logic that could be used in the QCT verification could not be simply projected onto unidimensional feature vectors. All improvement methods that were implemented (see Chapter 5.6 for more details) did not lead to a better biometric performance which can compete with VeriFinger or the Spectral Minutiae method. Nevertheless, the combination of QCT and HDS has the potential to perform at an acceptable rate, based on the analysis in this thesis. Further improvements as mentioned in Chapter 7 can help to increase the biometric performance to a level at which it is comparable to commercial products which do not utilize BTP.
7. Further Steps

There are numerous possibilities that need to be investigated in order to completely explore the topic of QCT in combination with the HDS. The preprocessing of the QCT for the HDS as explained in Chapter 5.5 is a crucial step and has a considerable impact on the biometric performance of the whole BTP system. Therefore, one of the most promising sections to review and improve is the preprocessing of the QCT. Different combination possibilities of the parameters can be employed for example to the $\delta$ parameter that performed especially for during QCT comparisons but performed poor for the HDS. Furthermore, the parameters can receive a different weight, depending on their position of the fingerprint by stretching the original co-domain.

Another idea is to use different cell sizes on different regions of the fingerprint. For example on areas that are known to produce very stable results (for example around the core) the grid could be refined, whereas the outer regions can have a rougher approximation allowing to cover for more noise. Also different cell sizes could be used for the different parameters. If there is a significant difference in the performance of singular parameters in one cell size over another, one parameter could be used as a feature from a different cell size, then another.

Another idea is to use a different tessellation of the fingerprint. Huppmann suggested in his thesis [Hup07] about Gabor Filters a circular tessellation (see Figure 7.1). This approach requires a more sophisticated and complex triplet selection process but instead of the suggested brute force triplet selection, the values could be mathematically derived from the ridges in the cells with the Hough transform [Hou59].

More research can be done in the statistics part of the HDS. The amount of fingerprints necessary to determine the stable positions is not specifically defined and should be further investigated. The maximum possible number of fingerprints that could be used for enrollment was 5 and it is not certain that it provides enough information for the statistics module to choose the most stable positions.

Different ECCs then the BCH can be used to encode and decode that protected
7. Further Steps

Figure 7.1.: A circular tessellation automatically provides a finer grid for areas around the reference point.

data. The BCH code can only handle 1-bit errors well. If fingerprints should be predominantly error prone to block errors, the BCH code will fail to cover the noise. At last, the co-sinusoidal function itself could be improved or extended to specifically generate features that work well together with the HDS.
Bibliography


Bibliography


Bibliography


8. Glossary

**Enrollment**: The registration in a biometric system. During the enrollment, a reference of the biometric data will be stored that is necessary for later authentication.

**Verification**: Verification is the comparison of a person with who he claims to be.

**Identification**: Identification is the ability to recognize an individual amongst all enrolled subjects of a system.

**Protected Templates**: Biometric data that is secured against exploitation by a BTP system.

**BTP**: Biometric Template Protection

**PI**: Pseudo Identifier

**AD**: Auxiliary Data

**SD**: Supplementary Data

**PIE**: Pseudo Identity Encoder

**QCT**: Quantized Co-Sinusoidal Triplets

**PIC**: Pseudo Identity Comparator

**PIV**: Pseudo Identity Verifier

**PIR**: Pseudo Identity Recorder

**FCT**: Failure-to-Capture

**FTX**: Failure-to-Extract

**FTE**: Failure-to-Enroll

**FTA**: Failure-to-Acquire

**FMR**: False-Match-Rate

**FAR**: False-Acceptance-Rate

**FNMR**: False-Non-Match-Rate

**FRR**: False-Reject-Rate

**ECC**: Equal Error Rate

**ROC**: Receiver Operating Characteristic

**FEC**: Forward Error Correction

**ARQ**: Automated Resend Request

**BCH**: A group of cyclic, error correcting block codes named after their discoverers Bose, Chaudhuri and Hocquenghem
9. Appendix
A. Parameter Evaluation
A. Parameter Evaluation

Figure A.1.: Evaluation of the influence of $\theta$

(a) $\theta$ score distribution

(b) $\theta$ ROC curve
A. Parameter Evaluation

Figure A.2.: Evaluation of the influence of $\lambda$

(a) $\lambda$ score distribution

(b) $\lambda$ ROC curve
A. Parameter Evaluation

(a) $\delta$ score distribution

(b) $\delta$ ROC curve

Figure A.3.: Evaluation of the influence of $\delta$
A. Parameter Evaluation

(a) Combination of $\theta$ and $\lambda$ score distribution

(b) Combination of $\theta$ and $\lambda$ ROC curve

Figure A.4.: Evaluation of the influence of the combination of $\theta$ and $\lambda$
A. Parameter Evaluation

(a) Combination of $\theta$ and $\delta$ score distribution

(b) Combination of $\theta$ and $\delta$ ROC curve

Figure A.5.: Evaluation of the influence of the combination of $\theta$ and $\delta$
A. Parameter Evaluation

(a) Combination of $\lambda$ and $\delta$ score distribution

(b) Combination of $\lambda$ and $\delta$ ROC curve

Figure A.6.: Evaluation of the influence of the combination of $\lambda$ and $\delta$
A. Parameter Evaluation

(a) Combination of $\theta$, $\lambda$ and $\delta$ score distribution

(b) Combination of $\theta$, $\lambda$ and $\delta$ ROC curve

Figure A.7.: Evaluation of the influence of the combination of $\theta$, $\lambda$ and $\delta$
B. Cellsize Evaluation
B. Cellsize Evaluation

Figure B.1.: The evaluation of the VeriFinger software as well as the Xu/Veldhuis approach

Figure B.2.: Score distribution, ROC curve and EERs for the cell size 5 x 5
B. Cellsize Evaluation

Figure B.3.: Score distribution, ROC curve and EERs for the cell size 6 x 6

Figure B.4.: Score distribution, ROC curve and EERs for the cell size 7 x 7
B. Cellsize Evaluation

Score Distributions and ROC Curves for Cellsize 8x8

Figure B.5.: Score distribution, ROC curve and EERs for the cell size 8 x 8

Score Distributions and ROC Curves for Cellsize 9x9

Figure B.6.: Score distribution, ROC curve and EERs for the cell size 9 x 9
B. Cellsize Evaluation

Score Distributions and ROC Curves for Cellsize 10x10

Figure B.7.: Score distribution, ROC curve and EERs for the cell size 10 x 10

Score Distributions and ROC Curves for Cellsize 11x11

Figure B.8.: Score distribution, ROC curve and EERs for the cell size 11 x 11
B. Cellsize Evaluation

Figure B.9.: Score distribution, ROC curve and EERs for the cell size 12 x 12

Figure B.10.: Score distribution, ROC curve and EERs for the cell size 13 x 13
B. Cellsize Evaluation

Score Distributions and ROC Curves for Cellsize 14x14

Figure B.11.: Score distribution, ROC curve and EERs for the cell size 14 x 14

Score Distributions and ROC Curves for Cellsize 15x15

Figure B.12.: Score distribution, ROC curve and EERs for the cell size 15 x 15
B. Cellsize Evaluation

Figure B.13.: Score distribution, ROC curve and EERs for the cell size 16 x 16

Figure B.14.: Score distribution, ROC curve and EERs for the cell size 17 x 17

80
B. Cellsize Evaluation

Figure B.15.: Score distribution, ROC curve and EERs for the cell size 18 x 18
C. HDS Test Results

Figure C.1.: Codeword Evaluation for the BCH ECC. The analysis was conducted for different cell sizes and BCH codeword lengths of 7, 15, 31, 63, 127, 255, 511 and all possible error correcting capabilities of those parameters.
C. HDS Test Results

Figure C.2.: The three codeword lengths for each individual cell size that produced the best FNMRs.
Figure C.3.: The three codeword lengths for each individual cell size with the worst FMRs
Figure C.4.: The three polynomial sizes for each individual cell size that produced the best FNMRs.
Figure C.5.: The three polynomial sizes for each individual cell size with the worst FMRs
Figure C.6.: The most stable QCT parameters as selected by the HDS. On the x-axis are the positions in the resulting feature vector with the most stable position at 1 and the least stable position at 1944. A fingerprint was comprised by 18 * 18 cells and 6 parameters describing each cell (therefore 18 * 18 * 6). On the y-axis is the number of users. For this test, only the FVC2002 DB2_a was used which contained 90 users with 7 sessions each, therefore leading to a maximum of 90 on the y axis.
Figure C.7.: The most stable QCT parameters as selected by the HDS. The areas with no ridge information were pushed back to the end of the feature vector. On the x-axis are the positions in the resulting feature vector with the most stable position at 1 and the least stable position at 1944. A fingerprint was comprised by $18 \times 18$ cells and 6 parameters describing each cell (therefore $18 \times 18 \times 6$). On the y-axis is the number of users. For this test, only the FVC2002 DB2.a was used which contained 90 users.
Figure C.8.: The most stable QCT parameters as selected by the HDS. On the x-axis are the positions in the resulting feature vector with the most stable position at 1 and the least stable position at 29424. A fingerprint was comprised by 20*20 to 18*18 cells (depending on the cell size), 6 parameters describing each cell, and 13 different cell sizes. On the y-axis is the number of users. For this test, only the FVC2002 DB2_a was used which contains 90 users. The cell sizes range from 5 to 18, where blue represents the smallest cell size and red the biggest.
C. HDS Test Results

Figure C.9.: The most stable QCT parameters as selected by the HDS. The areas with no ridge information were pushed back to the end of the feature vector. On the x-axis are the positions in the resulting feature vector with the most stable position at 1 and the least stable position at 29424. A fingerprint was comprised by $20 \times 20$ to $18 \times 18$ cells (depending on the cell size), 6 parameters describing each cell, and 13 different cell sizes. On the y-axis is the number of users. For this test, only the FVC2002 DB2a was used which contained 90 users. The cell sizes range from 5 to 18, where blue represents the smallest cell size and red the biggest.
Figure C.10.: The verification outcome of all cell sizes with all BCH codeword combinations. Only one probe was used for the verification.
Figure C.11.: The verification outcome of all cell sizes with all BCH code-word combinations. Multiple (4 and 2) samples were used for the verification.