ISBN 978-82-326-1930-6 (printed version)
ISBN 978-82-326-1931-3 (electronic version)
ISSN 1503-8181

Doctoral thesis at NTNU, 2016:296

Guoqiang Li

Innovative methods for large-scale fingerprint identification systems

Facilitating searching in a large-scale database

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of
Science and Technology
Faculty of Computer Science and Media Technology
NISlab - Norwegian Information Security Laboratory

Doctoral thesis at NTNU, 2016:296

NTNU

Guoqiang Li

Innovative methods for large-scale fingerprint identification systems

Facilitating searching in a large-scale database

Thesis for the degree of Philosophiae Doctor

Trondheim, October 2016

Norwegian University of Science and Technology

Faculty of Computer Science and Media Technology

NISlab - Norwegian Information Security Laboratory

NTNU

Norwegian University of
Science and Technology

NTNU
Norwegian University of Science and Technology

Thesis for the degree of Philosophiae Doctor

Faculty of Computer Science and Media Technology

ISBN 978-82-326-1930-6 (printed version)
ISBN 978-82-326-1931-3 (electronic version)
ISSN 1503-8181

Doctoral thesis at NTNU, 2016:296

Printed by Skipnes Kommunikasjon as

Innovative methods for large-scale fingerprint identification systems

Faculty of Computer Science and Media Technology
Norwegian University of Science and Technology

First, there is a mountain, then there is no mountain, then there is.

(Traditional Buddhist saying, via Donovan)

Declaration of Authorship

I, Guoqiang Li, hereby declare that this thesis and the work presented in it is entirely my own. Where I have consulted the work of others, this is always clearly stated.

Signed:

(Guoqiang Li)

Date:

Summary

Fingerprint recognition has gained wide acceptance and great popularity after the fingerprint recognition based applications have been adopted in diverse scenarios, such as forensics area (mainly by law enforcement agencies), access control products, financial transaction systems and mobile devices, etc. However, new challenges also emerge along with the extensive deployment of these systems. During the recognition process, a shorter response time is always desirable when an individual needs to be identified in a system with a database consisting of millions of fingerprints. In some systems, the database's size is continuously growing. Meanwhile, gathering these millions of fingerprints in the database would be a high value target for the adversaries. With the hardware improvement in new products (such as the smartphone), it is also possible to incorporate fingerprint recognition into these new products in a user-friendly and low-cost manner, and further use for establishing the identity of an individual. These challenges and possibilities motivated us to investigate innovative methods which would benefit large-scale fingerprint identification systems in terms of accuracy, efficiency and security.

The performance of a large-scale fingerprint identification system can be affected in a number of aspects involved in the whole recognition process whose components generally consist of data acquisition, sample pre-processing, template creation, feature extraction, comparison algorithm and data storage. It is difficult to investigate all the research aspects involved in these components in one dissertation. We chose to work on several research aspects that we consider are either rarely studied or crucial for a large-scale fingerprint identification system.

The performance of the fingerprint identification system is sensitive to the sample quality, hence the first research aspect that we studied is to assess the quality of fingerprint samples taken from a smartphone's camera. The smartphone has become as a part of our daily lives. Most smartphones contain a high resolution camera, network connectivity, powerful processor and large memory. These advanced hardware make a smartphone possible to act as a fingerprint sensor without adding extra resources. However, the quality of samples captured by such general-purposed cameras under an uncontrolled environment is unstable due to defocusing, poor illumination, or camera motion during the data acquisition process. In this dissertation, a quality assessment approach is designed to qualify the fingerprint samples taken from the smartphones' cameras. In a practical scenario, a re-capturing action will be activated in order to obtain a good quality sample when the quality of a captured sample is considered poor by the proposed approach. In the end, a higher quality sample can contribute to the system recognition accuracy.

In order to accurately and efficiently establish the identity of an individual in large-scale fingerprint identification systems, fingerprint indexing algorithm plays a crucial role in these systems. The second research aspect that we worked on is fingerprint indexing whose purpose is to output a short list of candidate identities which will be further used by a verification algorithm or even a human expert for manual verification. There are two research topics involved in a fingerprint indexing algorithm: (1) extract features which are suitable for building index space; (2) build the index space and retrieve candidate identities. In this work, three feature extraction methods are developed based on the fingerprint template, and different index space creation methods are explored to build the index space and to retrieve candidates.

According to a law 'EU General Data Protection Regulation' published in 2016, biometric data is recognized as sensitive data which requires protection. Thus the security is important for the biometric system. The third research aspect in this dissertation is how to protect the user's fingerprint data. We studied this aspect by developing two approaches. The first one is a fingerprint template protection approach based on Bloom filters. We investigated applying Bloom filters on fingerprint data, while Bloom filters have been successfully used to protect face data and iris data. The experimental results proofed the feasibility of this attempt. The second one is that we designed a fingerprint indexing algorithm in the encrypted domain. The proposed approach extracts the binary features and builds index space by using encrypted minutia information, thus no plain fingerprint data needs to be stored in the database. The security of the proposed approach is enhanced by a standard encryption algorithm.

Acknowledgments

I would like to express my deepest appreciation to my principal supervisor, *Prof. Dr. Christoph Busch* for his time, advices and expert guidance throughout my PhD program. Without his incredible patience and encouragement, this dissertation would not have been completed.

I would like to gratefully thank my co-supervisor, *Prof. Dr. Bian Yang* for his invaluable advices, immense knowledge and insights. His guidance helped all the time of my research and publications. I would also like to express my gratitude to my co-supervisor *Prof. Dr. Patrick Bours* for his support during my study. I would like to express my appreciation to the members of the evaluation committee, *Prof. Dr. Paulo Lobato Correia*, *Prof. Dr. Matteo Ferrara*, *Prof. Dr. Sule Yildirim-Yayilgan* and the head of the committee *Prof. Dr. Stephen Wolthusen* for their suggestions and valuable comments.

I am very grateful for having the financial support from two European projects FIDELITY and PIDaaS, and it was a great pleasure and a fascinating experience working with our excellent project partners via countless discussions, meetings and writing. A special thanks to *Raghavendra Ramachandra* for his brilliant suggestions and support when we worked together for FIDELITY. I would also like to thank *Prof. Dr. Davide Maltoni* from University of Bologna for providing a large-scale synthetic fingerprint dataset used in this dissertation.

Further, thanks to my lab mates in Norwegian Biometrics Laboratory: *Anika Pflug*, *Ctirad Sousedik*, *Daniel Hartung*, *Edlira Martiri*, *Kiran Bylappa Raja*, *Marta Gomez-Barrero*, *Martin Olsen*, *Martin Stokkenes*, *Mohammed Derawi*, *Pankaj Wasnik*, *Soumik Mondal*. It was my great pleasure to spend the time with you and create so many enjoyable memories. In addition, I very much appreciate *Norwegian Information Security Laboratory* for hosting me as a PhD student.

Last but not the least, I would like to thank my friends for the great time and conversations in Gjovik, and I own a big thanks to my family for their selfless supports and understanding during this journey.

.

Contents

List of Figures

List of Tables

List of Algorithms

Part I

Introduction

This part gives an overview of this dissertation. In Chapter 1, we introduce the concept of a fingerprint identification system, and then we discuss our motivations and research questions which have been addressed in this dissertation. In Chapter 2, we investigate the state-of-the-art algorithms from three research aspects that were also selected as our studying topics. In Chapter 3, we describe our contributions corresponding to each research question.

Chapter 1

Fingerprint Identification System

1.1 Introduction

Some of humans' characteristics (such as face, voice and handwritten signature) have been used to recognize individuals in our daily lives. However, we have difficulty to recognize the individuals who are not close or familiar persons by using these characteristics. A solution to assist human with the purpose of identification is the *biometric system* which is an automated pattern recognition system to verify or recognize the identity of individuals based on their behavioural characteristics (such as gait, signature, keystroke dynamic, etc.) and biological characteristics (such as fingerprint, face, iris, finger vein, etc.) [143, 102, 108]. A variety of biometric systems have been deployed and even used for a long history for law enforcement and commercial uses.

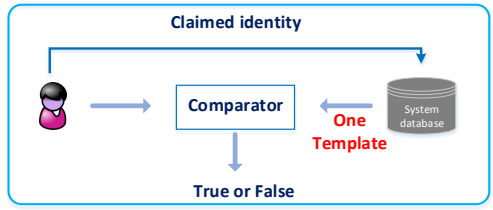
With the rapid improvement of the processing capability, memory and sensors in modern devices, more and more large-scale biometric systems have became operational or been developing around the world. In the U.S., the Federal Bureau of Investigation (FBI) IAFIS (Integrated Automated Fingerprint Identification System) major components became operational early in 1999 [26, 7], and it is hosting more than 70 million subjects with criminal background and 34 million civil prints according to the FBI's IAFIS website [7]. US-VISIT (United States Visitor and Immigrant Status Indication Technology) program became operational in 2004 [26, 31], and collected fingerprints and face images from more than 90 million subjects [178]. In Europe, The Visa Information System (VIS) was introduced in October 2011 to exchange visa data within Schengen States [34, 36]. Over 20 million applications have been processed by the end of 2015 [35], and every applicant has to provide his/her fingerprints and a digital face image since October, 2015. In Asia, India has been working on the world's largest biometric identity system (UIA of India) which aims to issue a 12-digit unique identity number for each resident in a country with 1.2 billion population [16]. This unique identity number is guaranteed by collecting the biometric data from each resident including fingerprint, iris and face. So far, over 1 billion have enrolled in this system. Table 1.1 lists these notable large-scale applications.

Based on the context of a application, a biometric system can be categorized into the

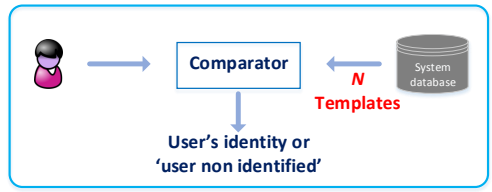
Table 1.1: A List of some notable large-scale systems. Note that 'Over 20 million applications processed' in EU VIS doesn't mean there are biometric data from 20 million subjects. But we believe it will reach this point soon, since every applicant has to provide his/her fingerprints and a digital face image since October, 2015.

Project name	Database size
FBI IAFIS	Over 104 million subjects [7]
US-VISIT program	Over 90 million subjects [178]
EU VIS	Over 20 million applications processed [35]
UIA of India	Over 1 billion people enrolled [16]

following two types of systems:



(a) Verification system



(b) Identification system

Figure 1.1: Block diagrams of verification system and identification system(adapted from [143]).

- **Verification system** compares the biometric data captured from a presented individual with a corresponding biometric data (for instance, links to the presented person by an username) pre-stored in the database, which indicates a 1:1 comparison as seen in 1.1a. The purpose of verification system is to confirm the identity of the presented individual by outputting a binary result "True or False" in order to answer a question "Is this person who he/she claims to be?".
- **Identification system** compares the captured biometric data from a presented individual with all biometric data stored in the database which indicates a 1:N comparisons, where N is the total number of subjects enrolled in the database as seen in 1.1b. The purpose of identification system is to establish the identity of the presented individual. Generally speaking, this system tries to answer the question "who is this person?", and will output the result: identity of the presented individual or "this person is not enrolled / identified".

From the technical perspective, the major difference between biometric verification system and identification system is the number of comparisons (1 comparison for verification versus N comparisons for identification). This N depends on the number of subjects enrolled in the database, thus the accuracy and response time of the identification system will be critical when the database contains millions of subjects, such as the system listed in Table. And we also can see that fingerprint is the common biometric modality which is selected by all these systems. Sometimes fingerprint-based biometric system becomes the synonym of the biometric system due to its popularity especially used in law enforcement community [71].

We focus on fingerprint identification system. A fingerprint is composed of a series of flow-like ridges and furrows on the surface of human fingers [125, 143], and it has been

used to verify the identity of humans for a long time. The first scientific paper on fingerprints was published in 1684 by N.Grew who gave the definition of ridge, furrow, and pore structure in a fingerprint [125]. The first research paper on fingerprint automatic comparison was published in 1963 [194]. A lot of fingerprint recognition algorithms have published and been applied to the practical systems. However, new challenges occur when we incorporate fingerprint recognition with new technology and devices, such as smartphone. And a shorter response time is still desirable while the database size of the existing fingerprint identification system is continuously growing. People are also getting concerned about the security and privacy of their biometric data when everything comes to online.

In this dissertation, we will research on several aspects related to large-scale fingerprint identification systems. In Section 1.2, we divide a fingerprint identification system into a set of components according to our understanding, and discuss the research aspects which are commonly involved in each component. In Section 1.3, we describe the motivations and the research objectives which we would like to address in this dissertation. The research questions are also formalized in Section 1.3. Section 1.4 describes the structure of this dissertation in order to provide an overview of the dissertation. The terminology used in this dissertation refers to ISO/IEC International Standard 2382:2012 [102].

1.2 Research aspects in a fingerprint identification system

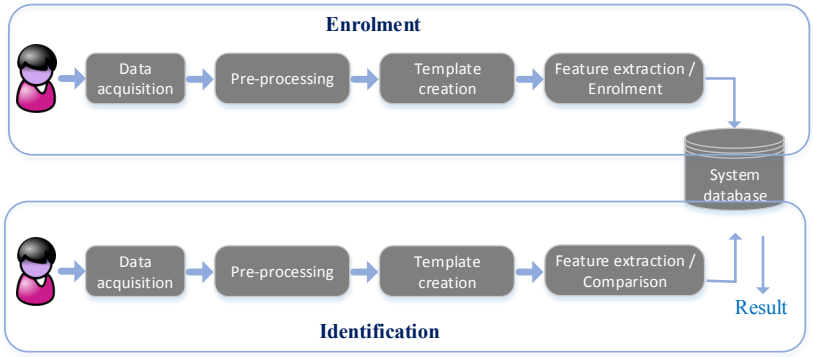


Figure 1.2: Block diagram of main modules for enrolment stage and identification stage in a fingerprint identification system.

A fingerprint identification system generally consists of two stages: enrollment and identification as shown in Figure 1.2. There are four modules in each stage including three common modules which are the data acquisition module, the pre-processing module and the template creation module.

The data acquisition module captures the fingerprint sample from a finger of an individual by using a dedicated sensor or any device which can obtain the ridge and furrow pattern of the finger. A research aspect used in data acquisition module is to assess the fingerprint sample quality before further processing, especially for enrollment stage. For instance, it would increase the system error rate if an image without a fingerprint is enrolled as a reference in the database. Therefore the fingerprint sample quality assessment is an essential component during the data acquisition in order not to negatively affect the system error rate. In a practical scenario, a re-capture action will be activated if the quality of

captured sample doesn't meet the requirement of a fingerprint sample quality assessment component.

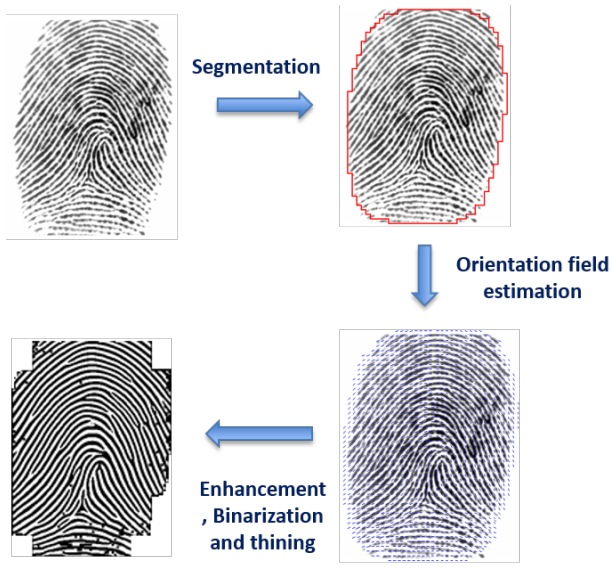


Figure 1.3: An example of fingerprint sample pre-processing. The sample is selected from FVC_2002 [12], and the right two pictures are adopted from [14].

The pre-processing module can involve several research aspects: segmentation, orientation field estimation, ridge enhancement, binarization and thinning [195]. The goal of fingerprint segmentation is to decompose the fingerprint sample image into two parts: the foreground, which is the region of interest (ridges and furrows), and the background. Fingerprint segmentation is important for an automated identification system, since it can prevent extracting spurious features from the background region. Following the fingerprint segmentation is the orientation field estimation which heavily affects the subsequent processes [226]. For instance, the directional filtering in ridge enhancement highly relies on the ridge direction flow [107], and a variety of singularity detection methods [52, 116] are based on Poincoré index which is calculated from orientation field. The other research aspects related to pre-processing module are ridge enhancement, binarization and thinning. Figure 1.3 illustrates an example how to pre-process a raw sample by applying segmentation, orientation field estimation, enhancement, binarization and thinning modules. After pre-processing, a fingerprint template can be generated by extracting the features from processed image (such as, the bottom-left image in Figure 1.3).

A commonly used fingerprint template is the minutia based template, since a minutia is considered the most robust feature and also has been standardized in ISO/IEC 19794-2 [99]. Some other features that may also be included in fingerprint template are singular points information, ridge information, or pore information [183]. A fingerprint-based biometric system generally stores fingerprint templates rather than raw fingerprint images. Storing fingerprint template has two advantages: (1) save storage space; (2) reduce response time during comparison, because the pre-processing step can be addressed offline at enrolment stage. However, some researchers have revealed that the original fingerprint information could be reconstructed from the fingerprint template [60, 78]. Thus storing fingerprint template in the plain domain may be risky, especially if the biometric data can be misused

in the rest of our life once it is compromised. This leads to another important topics: how to protect the users' fingerprint data in a large-scale fingerprint identification system, while fingerprint template protection has been studied more than a decade, but hasn't been well established yet.

During the identification stage, a crucial module is the comparison algorithm (or called verification algorithm) which commonly extracts robust features from a fingerprint template in order to produce a similarity score between a reference template and a probe template. Due to the sample translation and rotation commonly occurred at data acquisition, another research aspect involved in a comparison algorithm is the fingerprint alignment. The system performance would be significantly improved if the samples from the same source can be aligned properly, however, it is quite challenging to achieve. As we mentioned in previous section, a fingerprint identification system may perform N comparisons by using the probe fingerprint against all N enrolled fingerprints. However, this operation could be time consuming when this N comes to a large-scale level. In order to shorten the response time, a natural thought is to reduce the number of comparisons by dividing the whole database into several subsets. Then the probe sample is only needed to compare with the fingerprint in a single subset based on this probe's class. A famous classification method is called Henry classification system which was made by Edward Henry in 1896 [116]. The Henry classification system categorizes the fingerprints into five classes: arch, tented arch, left loop, right loop and whorl. Ideally, the number of comparisons can be reduced 80% after this classification if the fingerprints are evenly distributed to these five classes. But the distribution of fingerprints in these five classes is uneven: 3.7% arch, 2.9% tented arch, 33.8% left loop, 31.7% right loop and 27.9% whorl according to the result published in [143, 205]. In addition, the number of classes in Henry classification is limited. Even though the distribution is even, 20% searching space (if there are still 5 classes) is still large for a large-scale fingerprint identification system with millions of subjected enrolled.

Another solution to reduce the number of comparisons is the fingerprint indexing algorithm which retrieves a short list of potential candidates which will be further used by a verification algorithm to conduct a thorough comparison. In other word, the fingerprint indexing algorithm is a pre-selection algorithm, and a fingerprint identification system can be divided into two sequential steps: a fingerprint indexing step and a thorough comparison step, where the thorough comparison step is to compare the probe sample against each candidate by using a verification algorithm. With the help of fingerprint indexing, the number of comparisons can be significantly reduced. In general, a fingerprint indexing algorithm can be studied from two aspects: (1) extracting robust features which are suitable for creating index space; (2) building index space and retrieving candidates. The following section will discuss the research questions that we have chosen to answer in this dissertation.

1.3 Research objectives and questions

It is difficult to cover all research aspects involved in the fingerprint identification system in one dissertation, hence we selected several aspects which we consider are either rarely studied or crucial for a large-scale fingerprint identification system.

Controlling the quality of the fingerprint sample during data acquisition is a crucial step for any fingerprint recognition based biometric system. The first research topic selected for this dissertation is to assess the quality of a fingerprint sample captured from a smartphone's camera. This fingerprint sample captured from smartphone's camera or touchless device is also called fingerphoto, since it is technically the same as a normal image taken by a camera which could generally contain diverse backgrounds. There are two main reasons that why we decided to work on fingerphoto quality assessment: (1) the smartphone has become a part of our lives, and the embedded camera (high resolution, auto-focus) as well as other hardwares (such as powerful chipset, memory etc.) embedded in the smartphone

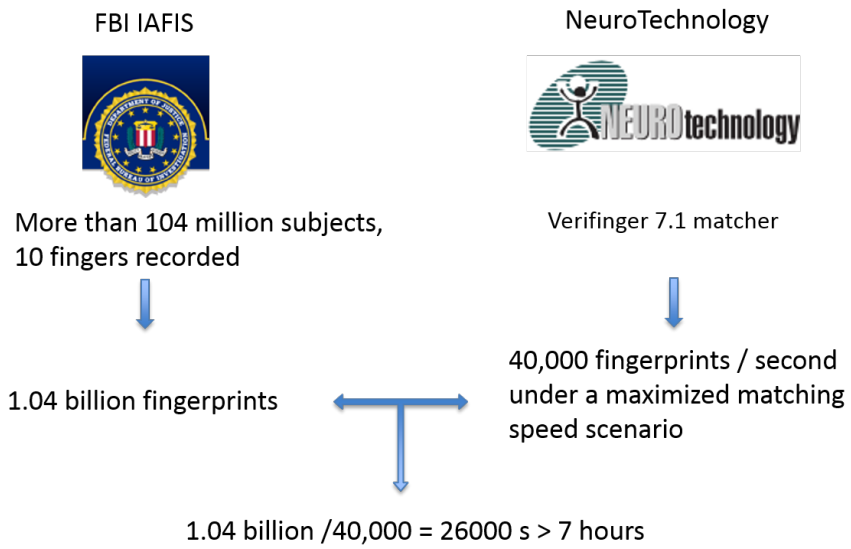


Figure 1.4: An example to illustrate why we need fingerprint indexing in a large-scale fingerprint identification system: it will take more than 7 hours to get the response from FBI IAFIS database [7] when we conduct an exhaustive searching (comparing the fingerprint to be identified with all the references enrolled in the database) by using a commercial product (NeuroTechnology matcher).

have the capabilities to make a smartphone's camera as an alternative of a dedicated fingerprint sensor. This also brings the convenience and low-cost on the sensor; (2) a lot of researchers have extensively worked on the quality assessment for the fingerprint captured by a dedicated sensor, however, fingerphoto quality assessment is rarely studied.

Every biometric identification system contains a database to store the biometric data. People are concerned about the security and privacy of their biometric data stored in the database. Last year, the Office of Personnel Management in the U.S. said that there are 5.6 million fingerprints stolen in a cyberattack [30]. Recently, a massive data breach put fingerprints from 55 million Philippine voters in danger [27]. On April 14, 2016, EU General Data Protection Regulation ("GDPR") became law, and biometric data is recognized as sensitive data which requires extra protection. Thus security is important for the biometric data. The Second research aspect selected for this dissertation is how to protect the user's fingerprint data when we need to store them in a database.

The third research aspect that we chose to work on is the fingerprint indexing which is very important for the large-scale fingerprint identification system. Figure 1.4 gives an example to explain why it is important. Let's assume a probe sample needs to be compared against all enrolled references in FBI IAFIS database storing fingerprints from 104 million subjects with 10 fingers recorded. Then the number of comparisons is 1.04 billion. On the other hand, NeuroTechnology Verifinger 7.1 matcher has the capability to compare 40,000 fingerprints per second under a maximized matching speed scenario [23]. In the end, the response time will be more than 7 hours if we perform an exhaustive searching (compare the fingerprint to be identified against all the references enrolled in the database). In order to shorten this response time, we decided to work on developing the fingerprint indexing algorithm which can facilitate searching in a large-scale fingerprint identification system.

Besides studying the fingerprint indexing algorithm without considering security measure, we also decided to investigate embedding the security mechanism into the fingerprint indexing algorithm.

Based on the above research objectives, this dissertation is focusing on answering a number of research questions as listed below:

RQ_1 **If we assume the smartphone's camera can facilitate the fingerprint sample capturing process, is it feasible to assess the quality of the fingerphoto taken by a general-purposed smartphone's camera in a real-life scenario?**

RQ_2 **Besides the existing feature extraction methods in the literature, what features can still be extracted from the fingerprint template and outperform the existing ones?**

RQ_3 **How to build the index space and retrieve candidate identities in a fingerprint indexing algorithm?**

RQ_4 **How to embed the privacy-preserving capability for the large-scale fingerprint identification system while still keeping the performance?**

1.4 Structure of the dissertation

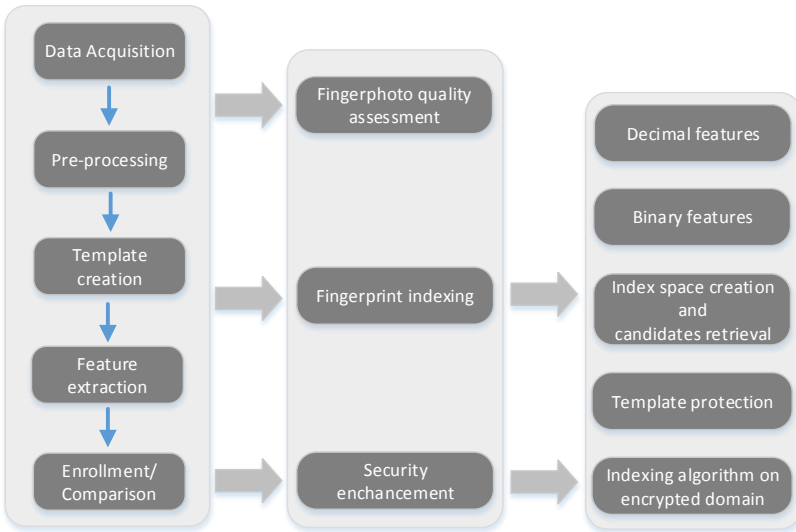


Figure 1.5: Block diagram of the main research aspects in this dissertation: fingerphoto quality assessment, fingerprint indexing and security enhancement.

We grouped the research questions into three main parts and structured this dissertation accordingly: fingerphoto quality assessment, fingerprint indexing and security enhancement. A brief introduction of each part is given as follows.

- Part I is an introduction. The remaining of this first part consists of two chapters: the following chapter gives an elaborate overview of the state of the art on the research

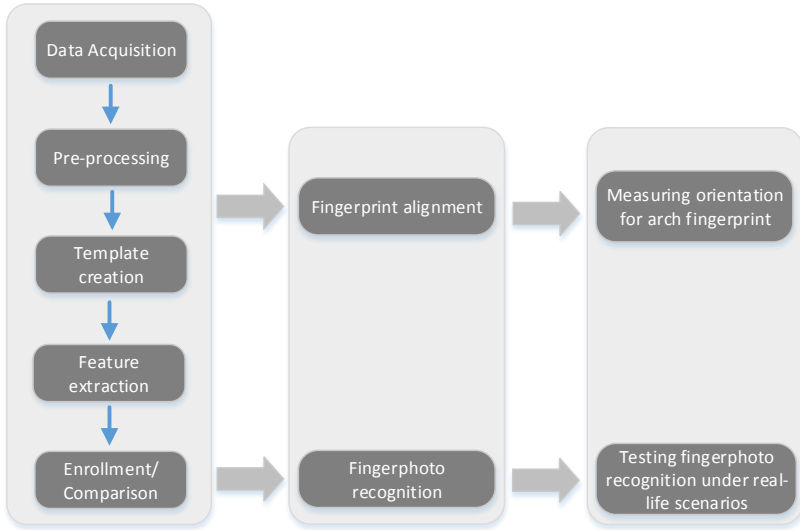


Figure 1.6: Block diagram of the research work in the appendix part: fingerprint alignment and fingerphoto recognition.

aspects selected for this dissertation; Chapter 3 describes the contributions of this dissertation as well as a list of publications.

- Part II, Part III and Part IV are composed of a number of research papers which can be categorised into three main research aspects: fingerphoto quality assessment, fingerprint indexing and security enhancement. Figure 1.5 gives a block diagram which illustrates the structure of the main part of this thesis. The first column of this figure gives some general components in a fingerprint identification system. The third column lists the main features that we have worked for each research aspect. This figure would be helpful to guide the reader, and will be displayed in the beginning of each part along with a description of the topics that are discussed in that part.
- Part V draws the conclusions and proposes the future work which can be further studied.
- Part VI is the appendix where we discussed two topics. Firstly, we proposed an approach to measure the orientation for the arch fingerprint by taking an input point with high curvature. This is semi-automated approach which is the reason that we don't include it in the main part. However, this approach can be further improved to an automated approach and integrated into the fingerprint identification system. Secondly, we investigated what would be the performance in terms of accuracy when we identify an individual by using the fingerphotos taken by a smartphone under real-life scenarios. The experimental results on this investigation is not desirable, which motivated to study on fingerphoto quality assessment. As seen in Figure 1.6, these two topics can be classified into two research aspects respectively: fingerprint alignment and fingerphoto recognition. We believe these two research aspects deserve to be studied in depth.

Chapter 2

The State of the Art

2.1 Introduction

This chapter gives an overview of the state-of-the-art algorithms from the following three perspectives which are also related to the research aspects studied in this dissertation.

- **Fingerprint quality assessment and fingerphoto recognition algorithms**: assessing the quality of the fingerprint captured by dedicated sensors (such as optical sensor) has been extensively studied by the researchers. At the beginning of this survey, a brief introduction of the fingerprint quality assessment will be given. The detailed information regarding this topic refers to a doctoral dissertations 'fingerprint image quality' published in November, 2015 [10]. Assessing the quality of the fingerphoto taken from a smartphone's camera is rarely studied by the researchers. Besides surveying on the fingerphoto quality assessment, we expand to investigate fingerphoto recognition algorithms on a smartphone's camera. This investigation will be introduced in the following section.

- **Fingerprint template protection**: this is one of the research aspects worked in this dissertation. A brief survey of fingerprint template protection is described in Section 2.3 which will mainly focus on the algorithms proposed in recent years.

- **Fingerprint indexing**: fingerprint indexing is a crucial component in a large-scale fingerprint identification system when we want to shorten the response time as we exemplified in Section 1.3. An extensive survey of fingerprint indexing approaches in the literature is given in Section 2.4.

2.2 Fingerprint quality assessment and fingerphoto recognition algorithms

It is easy to understand that the better quality of the fingerprint sample will lead to a higher recognition performance. This sense was also demonstrated in [82] by using NFIQ (NIST Fingerprint Image Quality) algorithm which was developed by Tabassi et al. in 2004 [74] and was the first open algorithm regarding fingerprint quality assessment. NFIQ algorithm analyzes a fingerprint image by extracting features from minutia counts and four quality maps: a direction map, a low contrast map, a low ridge flow map and a high curvature map [158]. The extracted feature vector will be fed into a neural network and generates a value of 1 (highest quality) to 5 (lowest quality) to indicate the quality of that fingerprint image. In addition, a variety of features have also been developed by researchers to assess the quality of the fingerprint image, such as frequency domain analysis (FDA), Gabor quality feature, Gabor-Shen quality feature, local clarity score (LCS), orientation flow (OFL), orientation certainty level (OCL), etc. The detail of these features refers to a survey paper [158] and a doctoral dissertation [10]. Some of these features have been included in ISO/IEC TR 29794-4:2010 [74], and some of these features are being incorporated into an updated NFIQ: NIST Fingerprint Image Quality 2 (NFIQ 2) [24].

Fingerphoto recognition based on a smartphone's camera is a recently emerged topic. After the initiative of introducing a fingerprint recognition based authentication mechanism in Apple iPhone 5S by using a touch-based sensor, the acceptance and visibility of mobile biometrics have been dramatically increased. Instead of using touch-based sensor, as we mentioned earlier, another alternative of obtaining a fingerprint sample is to use the embedded camera in a smartphone, since the camera's capabilities have been improved in terms of resolution, auto-focus, etc. Adopting camera for sample acquisition also has the advantages to reduce the cost and promote the user experience, when the smartphone became a part of our life and not every existing smartphone has a touch-based sensor. Recent results [192, 177, 186, 185] have shown that it is feasible to implement this touchless fingerprint recognition based authentication mechanism on a smartphone. Some details will be discussed as follows.

In 2011, a first attempt of using embedded cameras on smartphone for fingerprint recognition was studied by Derawi et al. [73]. The fingerprint samples were collected by a smartphone, the Nokia N95, which was placed on a fixed hanger under a laboratory environment. After the data collection using the camera of the Nokia N95, the processing and evaluation were carried out on a normal PC. The lowest Equal Error Rate (EER) they achieved is 4.66% which at least implies that the fingerprint captured from this general-purposed camera could be applicable for fingerprint recognition without requiring a dedicated sensor. In contrast to the laboratory setting, Stein et al. [186] proposed and implemented a fingerprint recognition based authentication method on two Android smartphones by using their embedded cameras. They applied Sobel filter to calculate the gradient magnitudes of the finger photo captured by the smartphone's camera in order to measure the quality of this photo, since this gradient magnitudes reflects the sharpness level of the captured finger photo. Several preprocessing methods were also proposed in their work including a finger area detection method which uses the value of the read channel in each pixel to detect the border of the finger area. The performance in terms of EER achieved by this proposed approach is less than 20% based on the experiments which was conducted under a real-life scenario. Later in another paper, authors significantly improved EER down to 1%$\sim$3% by integrating MorphoLite SDK for minutiae extraction and template comparison [185].

A recent approach was proposed by Sankaran et al. [177] in 2015. There are three contributions on their work. The first contribution is a segmentation and enhancement algorithm for the sample captured from smartphone. The basic idea of segmentation algorithm is to use an adaptive skin color threshold to extract finger area from the photo because of the distinguishable color difference between finger area and background which is similar to the segmentation method used in [186]. The enhancement algorithm used in this approach consists of three steps: applying median filtering after converting segmented image to gray scale, histogram equalization and subtracting the Gaussian blurred image ($\sigma = 2$) from the original image itself. The second contribution of this approach is a new feature extraction method based on Scattering Networks (ScatNet) [56] which is a filter bank of wavelets. The main reason the authors chose Scattering Networks is due to its good capability at extracting texture patterns from low resolution image where minutiae are difficult to be precisely extracted. The third contribution is a public fingerphoto database along with their corresponding touch-based scan fingerprints.

Another recent work was conducted by Tiwari et al. [192] who also proposed a fingephoto recognition approach for smartphone. The proposed approach consists of five components: fingerphoto acquisition, a region of interest (ROI) extraction, enhancement, feature extraction and comparison. The main contribution of the proposed approach is a feature extraction based on scale invariant robust features (SURF) [44] which detects highly distinctive and rotation invariant points from the fingerphoto. These SURF feature points are also considered to be robust against scale and illumination changes in the image. The proposed approach yield an EER of 3.33% according to the experiments on a dataset with 50 subjects.

An EU project called MobilePass also presented a fingephoto-based recognition solution in EAB Research Project Conference (RPC) in 2015 [21]. According to their report [21], this solution consists of several processes: sample capturing using a contactless camera; automated detection of visible finer in ROI; fingerprint segmentation (detection of fingertip); image enhancement (normalization and contrast enhancement); fingerprint quality assessment (sharpness-measurement and NFIQ); minutia extraction and comparison. Despite the performance was not reported, the demonstration showed a bright future for this solution.

The above fingerphoto based recognition approaches can be viewed as a complete fingerprint recognition pipeline which generally consists of quality assessment, preprocessing component, feature extraction component, comparison component, etc. Besides these complete fingerprint recognition pipelines, some approaches also have been proposed to focus on a specific topic. These approaches could be integrated into the previous pipelines or used to replace some components of previous pipelines in order to improve the overall performance. We list these approaches below, then followed by a brief description for each approach.

- Fingerphoto quality assessment: [222], [218];
- Reference point detection: [117];
- Core point detection: [121];
- Pre-processing: [124].

There are two fingerphoto quality assessment approaches in the literature. The first one proposed an approach to select the best quality image from a couple of images by calculating the number of pixels in finger area [222]. The second quality assessment approach applies Fast Fourier Transform to detect ridge-like blocks which are divided from the fingerphoto, and the total number of these ridge-like blocks is used to decide the quality of the finger photo [218]. Khalil proposed a reference point detection from the finger photo based on discrete wavelet transform [117], and achieved a detection rate of 78.21% in uncontrolled scenario. Another core-point detection approach was proposed by Kurniawan et al. [121] by analyzing ridge information after applying a discrete Fourier transformation. Lee et al. [124] proposed a preprocessing algorithm for fingerprint segmentation and orientation field estimation. The proposed segmentation approach uses the color information, frequency information and region growing. The orientation field estimation is based on an iterative robust regression method which can ignore the residuals associated with the outliers.

In addition, there are some researchers who have worked on using digital camera (such as Canon) or webcamera for obtaining fingerprint samples. The fingerphotos taken by these cameras show the same characteristics as the fingerprint samples taken from a smartphone's camera. The algorithms proposed in these articles could also be applied on smartphone for the fingerphoto recognition, thus it is worth to be aware of these approaches. Piuri et al. [160] proposed an approach using fingerprint samples captured from a webcam. The proposed approach is composed of a number of components including blur reduction, background subtraction, fingertip segmentation, orientation field estimation, feature extraction and matching. Mueller and Sanchez-Reillo [149] also adopted several webcams to obtain fingerphotos used for feature extraction and comparison afterwards. They collected 400 fingerphotos from 3 different webcams, and achieved FAR=0.18% at FRR=10.29%. Another approach was proposed to address the fingerphoto captured from a digital camera (Canon PowerShot Pro 1) [92]. This approach applies Gabor filter to extract features and PCA to reduce the dimensionality of the Gabor feature vectors. A recent work used an IDS camera to capture fingerprint samples and developed a machine learning based feature extraction method for fingerprint verification [113].

2.3 Fingerprint template protection

Unlike password, our biometric data is irreplaceable and cannot be updated once our enrolled biometric data is leaked. As we discussed in Section 1.3, two massive database breaches made millions of fingerprints at risk. Due to this security and privacy concerns about user's biometric data, developing template protection is a topic which has been studied for more than one decade but gains attention, because it is difficult to meet the security requirements without significantly sacrificing the recognition performance [155]. A comprehensive survey on biometric template protection approaches is given by Rathgeb et al. in 2011 [171]. They investigated the vast majority of template protection approaches which have been published before 2011. Since then, new fingerprint template protection approach continuously emerges. Our survey on this topic will focus on the approaches published in recent years.

Besides the performance requirement for any biometric algorithm, ISO/IEC 24745 [103] defines two major security requirements which a template protection approach needs to meet: irreversibility and unlinkability. Irreversibility requires a template protection has the capability to prevent the reconstruction of the original biometric template from the protected template, at least computationally infeasible. Unlinkability requires a template protection has the capability to generate different versions of s protected template for different applications (renewability or revocability) while preventing cross-matching. Based on the discussion in [171], template protection approaches are commonly classified two categories: (1) biometric cryptosystem, which is either a key-binding scheme or a key-generation scheme; (2) cancelable biometrics which can perform the comparison directly on encrypted/transformed templates without decryption. We follow this classification, and the rest of this section will discuss fingerprint template protections from these two categories respectively.

A couple of biometric cryptosystems were proposed in the literature in recent years. Some of them are the improvements based on the fuzzy vault originally proposed by Juels Ari and Sudan Madhu [114] which is a key-binding scheme. It is worth to mention that an error correcting code (ECC) is the cornerstone of a fuzzy vault based fingerprint template protection schemes, since ECC has the strong capability to tolerate the sample variations and it has also been applied to other modalities [167, 154, 179]. In 2014, Bringer et al. [54] applied the fuzzy vault on a binary fingerprint representation (adapted from their previous work in [53]) rather than minutiae-based representation (commonly decimal features). This approach also designed a multi-finger fusion method at template level before applying fuzzy vault by considering that a system has more than one finger used for enrolment. Another fuzzy vault based approach are designed to provide cancellability and diversity by using Hadamard transformation [43]. Another key-binding biometric cryptosystems is designed based on Delaunay quadrangle and a template protection technique called PinSketch [219]. The proposed approach generates two features vectors from each Delaunay quadrangle: one geometric feature vector which will be encrypted by PinSketch; another feature vector is called auxiliary feature which is extracted from a Delaunay quadrangle-centered polar coordinate space. This auxiliary feature is a topology code which is generated by two steps: (1) divided the polar coordinate space into as set of small blocks; (2) calculate a numerical value for each block. Only the geometric feature is used for key binding. The purpose of the auxiliary features is to enhance the discriminatory power of Delaunay quadrangles in order to improve the recognition accuracy of the proposed approach. Another recently published biometric cryptosystem approach is based on a modified Voronoi neighbor structures (VNS) which binds a secret key. [220], and it is an alignment-free approach.

Researchers also put a lot of effort on cancelable biometrics which directly compares transformed template rather than applying standard encryption algorithm. After reviewing recently published approaches, we observe that these approaches can be classified into two coarse groups: binary representation based cancelable biometrics and non-binary rep-

resentation based cancelable biometrics. Due to the efficiency requirement for the large-scale fingerprint identification system, we think the binary representation based cancelable biometrics has the advantage to be used for the large-scale fingerprint identification system. The following two paragraphs will give a brief description for these two groups respectively.

A recent cancelable fingerprint template protection approach was published in 2016 by Wong [206] who summarized and improved their previous work in [208, 207]. This approach is designed using kernel principal components analysis (kernel PCA) to convert a unordered and variable-size MLC (multi-line code) template into an ordered and fixed-length binary template, while MLC template is secured by a random projection before using kernel PCA and binarization. Another binary representation based approach was designed by Jin et al. [112] who performed a polar coordinate transform and a 3-tuple based quantization to generate the binary template. Subsequently a user-specific token is issued to protect this binary template. An irreversible fingerprint template protection approach is designed by using Bloom filters [39]. While Bloom filters requires a binary template as input, the authors proposed a binarization method by using minuitae relation code (MRC) [38] which doesn't drop border minutiae and isolated minutiae for feature generation. A Delaunay triangulation based approach is designed by Sandhya et al. [176] who proposed a feature extraction method to generate a fixed length binary vector. Then applying discrete Fourier transform on this binary vector outputs a complex vector which will be multiplied by user's key to yield a cancelable template. Another Delaunay triangles based approached was also proposed by Sandhya et al. who adapts convolution coding to encrypt feature vector and Viterbi algorithm to retrieve codeword [175]. Instead of extracting features from Delaunay triangles, they also proposed a similar approach but using features extracted from a k-nearest neighborhood structure in [174]. Jin et al. [111] generates a binary template by using random projection and features extracted from a minutiae vicinity, then applies Randomized Graph-based Hamming Embedding (RGHE) for protection. Mirmohamadsadeghi and Drygajlo proposed an approach to protect the minutia cylinder-code (MCC) by combining a transformation and a user key in order to provide irrevocability and irreversibility [146]. Ferrara et al. [79] proposed an approach called P-MCC based on a noninvertible transform and the well-known local minutiae representation MCC (Minutia Cylinder-Code). The authors proposed a noninvertible transform consisting of a K-L (Karhunen-Love) projection and a binarization step. The plain binary vector obtained from each cylinder will be used as input for this noninvertible transform to generate a protected binary vector. During the comparison stage, the similarity score will be directly generated on those protected binary vectors. Later on, the same authors improved the P-MCC in order to provide the diversity and unlinkability. This improved P-MCC is called Two-Factor Protected Minutiae Cylinder-Code (2P-MCC) [80], and the main idea of the 2P-MCC is to permute a subset of the original bits of each binary vector according to a secret key. This partial permutation also shortens the length of the protected binary vector. In addition, there is a fingerprint template protection approach designed by the researchers in our group [214]. This approach extracts a binary secure hash bit string from a minutiae vicinity (defined by a central minutia and a set of closest neighboring minutiae). The security of this approach is obtained by introducing the random offsets to the original minutiae information, while the proposed approach still achieved the desirable recognition performance. Some of our algorithms developed in this dissertation were inspired by the ideas from this fingerprint template protection approach.

Regarding non-binary representation based cancelable biometrics, Yang et al. [221] proposed an approach which adapts a many-to-one based non-invertible transformation 'polar transformation' to protect original template. The main idea of this approach is to map the features extracted from the Delaunay triangle-based local structure rather than mapping a single minutia. The authors think that this idea has the advantage to mitigate the negative influence of fingerprint sample non-linear distortion. Moujahdi et al. [148] de-

veloped a very interesting protection approach which generates a structure called special spiral curves. This structure is constructed by using three pieces of information: a reference, minutiae and a key. The reference point and minutiae decide the basic shape of special spiral curves. The key is used to change the scaling and rotation of this basic shape of special spiral curves. The same fingerprint sample can generate as many these structures as we want depending on the number of different keys, but these structures are not cross-matched because of different scaling and rotation. However, this approach shifts the security concern from protecting template to key management. Based on the idea of this structure, researchers Prasad et al. [163] explored projecting these spiral curves into a 4-D space to generate a binary string, and further uses this binary string as input for Discrete Fourier Transform to create a non-invertible template. Prasad et al. also proposed another alignment-free approach using Discrete Fourier Transform [164]. This proposed approach considers each minutia as a reference point, and builds a set of rectangles. A feature vector is generated for each reference minutia by calculating distance and orientation from rectangles. In the end, the cancelable template is created by feeding these feature vectors into Discrete Fourier Transform. Another alignment-free cancelable approach is designed by Wang et al. [200] who first extract feature from a set of pair-minutiae. A pair-minutiae is constructed by pairing up any two minutiae in a fingerprint sample. Then these features will be mapped into a number of bins to generate a binary string. Since this binary string is not secure enough, the authors further apply Discrete Fourier Transform for converting this binary string to a complex vector. Yang et al. [216] proposed a nonlinear dynamic random projection to protect the fingerprint template. Contrary to the conventional random projection based approaches which normally preserve a secret key, this proposed approach dynamically assembles a random projects matrix whose columns are selected from a set of public candidate projection vectors. This dynamically selection mechanism makes the adversary computationally impossible to reverse the original fingerprint features.

In addition, Kaizhi et al. [115] proposed an approach combining cancelable biometrics and biometric cryptosystem. The cancelable biometrics used in this approach is BioHashing. The input of BioHashing is a feature vector called FingerCode which is adopted from [107], and the output of BioHashing is a bit string which will be bound with a key by Fuzzy commitment. The key point of this approach is to detect an accurate reference point in order to generate a stable FingerCode, otherwise the performance might severely deteriorate. Another hybrid approach was proposed by A.Ghany et al [83] who extracted the feature from the ridge information in a fingerprint sample, which is different to the above methods that commonly extract features from minutiae or directly using minutia information. However, not all ridge will be considered for feature extraction, but only some smooth curves called principal curves (details refer to [224]) will be used to extract features. Then Kekre Transform (a key-binding scheme) is applied to take these features as input and to generate Kekre Transform coefficients. These coefficients are further secured by incorporating BioHashing function.

2.4 Fingerprint indexing

2.4.1 Overview of fingerprint indexing

In order to avoid an exhaustive searching in a large-scale fingerprint identification system, a fingerprint indexing approach can be applied to facilitate searching by outputting a short list of candidates. A fingerprint indexing generally consists of three components: feature extraction, index space creation and candidates retrieval. Figure 2.1 illustrates how a fingerprint indexing approach can incorporated into a fingerprint identification system. Feature extraction component is applied to the reference sample during enrolment stage and to the probe sample during identification stage respectively. During enrolment stage, the index space creation component takes the features generated from the feature extraction compo-

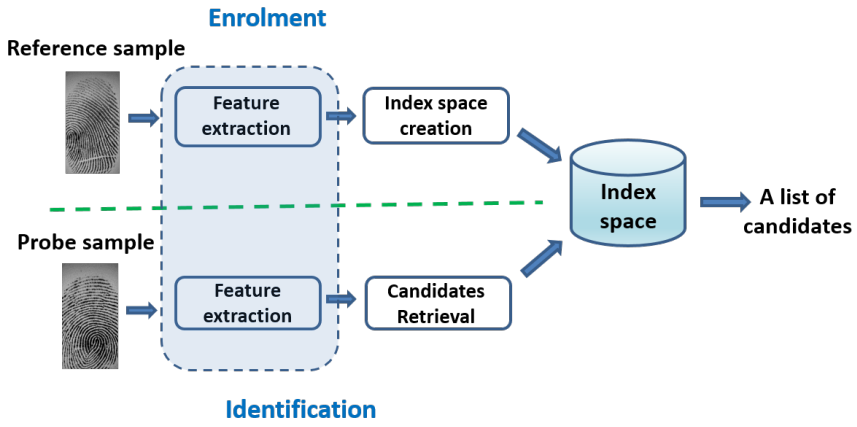


Figure 2.1: Illustration of how a fingerprint indexing approach can be applied into a fingerprint identification system.

nent to build up the index space which is a number of classes (or a list of tables storing the information of enrolled reference samples). Accordingly, the candidates retrieval component uses the features extracted from the probe sample to locate a small set of classes. Subsequently, the candidates will be determined from these targeted classes without considering the whole index space. This is why a fingerprint indexing approach can facilitate searching by reducing searching space. In the end, the short list of candidates is further used by a verification algorithm or even a human expert for manual verification. Since the candidates retrieval is highly reliant on the technique used in the index space creation component, we consider these two components as an united one 'index space creation and candidates retrieval component' in our following description. Section 2.4.2 gives an over of state of art feature extraction methods used for fingerprint indexing algorithms. Section 2.4.3 describes the index space creation and candidates retrieval methods developed by the researchers.

2.4.2 Feature extraction methods

Feature extraction method plays a critical role in a fingerprint indexing approach. A number of methods have been presented in the literature, and these methods can be classified in terms of the usage of different category of fingerprint pattern description. In general, there are three categories of fingerprint pattern description defined for fingerprint in biometrics community:

- **Category-1 feature (global features):** global ridge flow (as seen in 2.2 which is used for Henry Classification system), ridge frequency, orientation field (dash lines in Figure 2.3) and singular point (delta point marked as triangle and core point marked as circle in Figure 2.3, the definition of singular point refers to [99]), etc.;
- **Category-2 feature (local features):** minutia (as seen in Figure 2.4), local ridge pattern (ridge information around a minutia), etc.;
- **Category-3 feature (detailed features):** sweat pore, short ridge, island, etc. As seen in Figure 2.5.

Figure 2.2: Fingerprint global features: global ridge flow (figures are adopted from [4]).

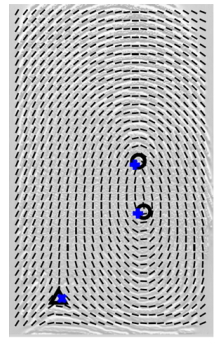


Figure 2.3: Fingerprint global features: orientation (dash lines) and singular point (delta point is marked as triangle, core point is marked as circle. Definition of singular point refers to [99]) (figure is adopted from [166]).

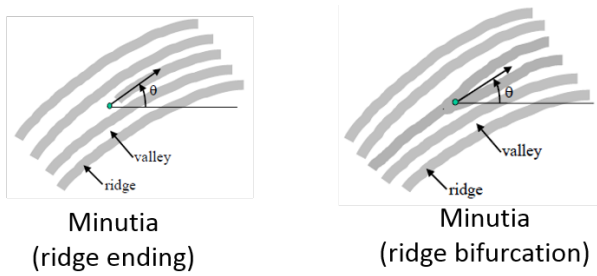


Figure 2.4: Typical fingerprint local feature: minutia (figures are adopted from [99]).

These three category features were investigated individually or collectively by researchers to extract robust features which will be used for building index space and searching. Thus, we can name three types of feature extraction methods: *Category-1 feature based methods*, *Category-2 feature based methods* and *Category-3 based methods*, while a feature extraction method based on 2 categories features will be classified into the higher level category. For instance, a feature extraction method will be assigned into local feature based algorithm if both Category-1 feature and Category-2 feature were used by this method. In addition, the

Figure 2.5: Category-3 feature: some detailed fingerprint features (figure is adopted from [41]).

rest of feature extraction methods are assigned into *other methods* if they are not using any of these three category features. We try to include as many existing fingerprint indexing approaches as possible. However, we think there are still some approaches missing due to access restriction and limited time.

2.4.2.1 Category-1 feature based methods

Global orientation field (OF) and singular points are two typical Category-1 features which have been investigated for fingerprint indexing approaches. Global orientation field (OF) is a topological feature which indicates the ridge flow by calculating an optimal dominant ridge direction for each block [170], and it has been studied by researchers to extract invariant features used for indexing fingerprint. One of OF-based fingerprint indexing approaches was based on a set of polar complex moments (PCMs) to develop a fixed-length feature vector representing a fingerprint sample [141]. Another approach uses an orientation feature vector by concatenating a set of local orientations within a circle of radius R [110]. Cappelli et al. proposed a fingerprint indexing approach based on orientation field and ridge frequency. [58]. An OF-based approach without relying on singular point was designed by Zheng et al. who adopted normalized histogram of orientation field as the feature vector [225]. Other researchers [212, 201, 142, 203] also applied OF for fingerprint indexing approach after proposing a novel fingerprint orientation estimation model.

Another typical Category-1 feature is singular point or a reference point if a singular point doesn't exist in the fingerprint. Singular point is considered as a fingerprint landmark which is also the basis of classifying fingerprints into five classes: right lop, left loop, whorl, arch and tented arch [116]. Researchers consider singular point has the advantage against local noise, two singular point fingerprint indexing approaches are proposed in [139] where the singular pint is generally detected after the orientation field estimation. However, detecting the singular point accurately is a challenging topic.

2.4.2.2 Category-2 feature based methods

Majority of researchers have been focusing on developing invariant features using Category-2 feature (minutia and local ridge information) to extract invariant features against fingerprint sample rotation and translation which commonly occur during sample acquisition. Category-2 feature based methods can be classed into three small groups: (1) methods only rely on minutia information; (2) methods consider minutia and local ridge information together; (3) methods combine the Category-1 feature and Category-2 feature.

Minutiae based template has been standardized in ISO/IEC 19794-2:2011 [99] and widely accepted by researchers to develop fingerprint indexing algorithms. In general, there are

around 30-100 minutiae can be extracted from a fingerprint sample. Because of the fingerprint sample translation and rotation, the location and direction of minutiae vary even though two samples were captured from the same source. This variation drives researchers to extract robust features instead of directly adopting minutiae information. Using Delaunay triangulation has been extensively studied to calculate a set of geometrical features which are considered to be invariant against sample translation and rotation. Delaunay triangulation was first proposed by Bebis et al. [48]. The idea of Delaunay triangulation is to form a smaller group of minutiae triangles ($O(N^3)$) triangles, where N is the number of minutia) instead of considering all possible triangles which can be formed by N minutiae. Ideally, N minutiae can form $O(N^3)$ triangles. The reduction of the minutiae triangles can benefit the processing time and tolerate the local noise and distortions. An improved work based on Delaunay triangulation was done by Gago-Alonso et al. in [81, 151] where the authors expanded the triangle set and introduced new features. Some similar methods based on minutiae triplet features were proposed in [223, 49, 68, 50] where several geometric constraints were set to eliminate some outliers during geometrical feature calculation. Features extracted from minutiae quadruplets also can be applied for indexing fingerprints as seen from articles [95, 96]. Instead of forming any geometric shape, researchers also investigated geometrical features which can be obtained from a set of minutiae, for instance, $m(m > 4)$ minutiae in a local area. One of these m minutiae is used as a central minutia and the rest of minutiae are aligned to this central minutia in order to minimize the effect of spurious and missing minutiae. After the local alignment, invariant features can be extracted to represent this local area. These methods are presented in [199, 42, 131]. Another local alignment based method is fingerprint indexing based on minutia cylinder-code [62] which constructs a 3-D cylinder based on a central its surrounding minutiae. All of these feature extraction methods only rely on minutiae location and direction information.

The second type of Category-2 feature based methods is using minutiae information together with local ridge information. Generally speaking, the robustness of extracted features would raise if more information from the sample image is considered during feature extraction. Ross et al. proposed a feature extraction method to generate ridge curve features and geometric features from the triplets based on Delaunay triangulation [172]. In Ross's method, only one ridge associated with each minutia is considered for ridge curve features. Liang et al. proposed a method to consider three ridges associated with bifurcation minutia [138, 137] instead of one ridge in Ross's method. Delaunay triangulation was also adopted in Liang's methods. Another ridge feature around a minutia is the ridge curvature which has also been used by Biswas et al. [51] to extract robust features combing the geometric features from the minutiae triplets. A feature extraction method proposed in [129] explored to use ridge curvature, ridge density around a minutia as well as geometrical information from the minutiae triplets.

A few researchers have been worked on combining Category-1 and Category-2 features to extract invariant features used for indexing fingerprints. Raffaele Cappelli and Matteo Ferrara proposed a method to use orientation field and ridge frequency information from Category-1 feature together with binary vector from minutiae cylinder-code [59], and they applied score level and rank-level fusion based on the features from these two levels respectively. Another method based on the combination of Category-1 and Category-2 features fuses three types of features: orientation filed, FingerCode and minutiae triplets, where FingerCode is a 384-dimensional vector indicating the ridge frequency derived from a region of interest around a reference point in the fingerprint sample. Munoz-Briseno et al. [152] proposed a method using a reference point and geometric features extracted from minutiae triplets based on Delaunay triangulation.

2.4.2.3 Category-3 feature based methods

Category-3 features contains swear pore, short ridge, island, etc while requiring high resolution sensor to obtain these details, for instance 1000 ppi sensor [105]. In the literature,

Table 2.1: List of fingerprint indexing approaches based on Category-1 features in the literature

Fingerprint indexing approach	Feature extraction	Index space creation
Zheng et al. (2011) [225]	orientation field	k-means
Liu et al. (2006) [142]	orientation field, reference point	continuous classification
Cappelli et al. [58]	orientation field, ridge frequency	continuous classification
Liu et al. (2005) [139]	singular point	unclear
Wang et al. (2007) [201]	orientation field	continuous classification
Jinag et al. (2006) [110]	orientation field	continuous classification
Liu et al. (2012) [141]	orientation field	k-means
Wang et al. (2011) [203]	orientation field	continuous classification
Liu et al. (2006) [140]	complex filter responses on orientation field	continuous classification

there are two fingerprint indexing approaches which explored using swear pore as a feature to create fingerprint index space. The first approach incorporated the swear pores with geometric features based on minutiae triplets which is from Category-2 feature. The second approach combines the features from three levels: ridge orientation, ridge frequency, ridge count and ridge length from Category-1; minutiae from Category-2; swear pores from Category-3 [161]. Considering these detailed features from Category-3 may benefit the performance of fingerprint indexing algorithm, while it may also compromise the computational complexity because of the combination of minutiae and swear pores.

2.4.2.4 Other methods

Shuai et al. proposed a feature extraction method based on scale invariant feature transformation (SIFT) which not only detects minutiae but also a lot of SIFT points [182]. A feature vector is composed of minutiae and reduced SIFT points. Similar as SIFT which has been extensively used in generic image retrieval, speeded-up robust features (SURF) has also adapted to extract features used for fingerprint identification system by He et al. [89]. Another interesting fingerprint indexing approach generates an index code for a fingerprint sample based on the similarity scores between this sample with a set of reference samples [86].

There is a secure fingerprint indexing approach that was published in 2013 by Hartloff et al. [88] who proposed to calculate distances and certain angles from minutiae template. These distances and certain angles are further used for generating a set of paths, then the proposed approach applies fuzzy vault on these paths for creating the index space. The security capability of proposed approach relies on fuzzy vault scheme which is vulnerable to brute force attack [145]. Recently, researchers also proposed solutions to address this drawback [189, 57], however, these improved fuzzy vault schemes haven't been adapted for fingerprint indexing.

Table 2.2: List of fingerprint indexing approaches based on Category-2 features in the literature, where LSH stands for Locality Sensitive Hashing.

Fingerprint indexing approach	Feature extraction	Index space creation
Gago-Alonso et al. (2013) [81]	minutiae triplets	invariant-based indexing
Munoz-Briseno et al. (2013) [151]	minutiae triplets	invariant-based indexing
Bhanu et al. (2003) [50]	minutiae triplets	invariant-based indexing
Yuan et al. (2012) [223]	minutiae triplets	LSH
Choi et al. (2003) [68]	minutiae triplets	continuous classification
Bhanu et al. (2001) [49]	minutiae triplets	continuous classification
Cappelli et al. (2011) [62]	minutiae cylinder-code	LSH
Vig et al. (2012) [199]	minutiae	invariant-based indexing
Bebis et al. (1999) [48]	minutiae triplets	invariant-based indexing
Feng et al. (2006) [76]	minutiae, local ridge	invariant-based indexing
Liang et al. (2007) [138, 137]	minutiae triplets, local ridge	invariant-based indexing
Biswas et al. (2008) [51]	minutiae triplets, ridge curvature	k-means
Ross et al. (2007) [172]	minutiae triplets, local ridge	k-means
Ogechukwu N. Iloanusi (2014) [96]	minutiae quadruplets	k-means
Iloanusi et al. (2011) [95]	minutiae quadruplets	k-means
Cappelli et al. (2012) [59]	orientation field, ridge frequency, minutiae cylinder-code	continuous classification, LSH
Boer et al. (2001) [72]	orientation field, singular point, minutiae triplets	continuous classification
Munoz-Briseno et al. (2014) [152]	singular point, minutiae triplets	invariant-based indexing

Table 2.3: List of fingerprint indexing approaches based on Category-3 features and other approaches in the literature

Fingerprint indexing approach	Feature extraction	Index space creation
N.Poonguzhali et al. (2013) [161]	global ridge info., minutia, swear pore	LSH
R.Singh et al. (2009) [183]	minutiae triplets, swear pore	invariant-based indexing
Shuai et al. [182]	reduced scale invariant feature transformation (SIFT)	LSH
He et al. (2010) [89]	speeded-up robust features (SURF)	PCA cluster + LSH
Li et al. (2006) [135]	indexing code based on symmetrical filters	
Gyaourova et al. (2008) [86]	index code based on similarity scores	

2.4.3 Index space creation and candidates retrieval methods

Depending on the features extracted from the fingerprint sample or fingerprint template, there are mainly three index space creation methods which can be named as invariant-based indexing, continuous classification and approximate nearest neighbor (ANN) indexing. Invariant-based indexing calculates an invariant index value for each feature vector which will be subsequently assigned into the hash table associated with that unique index value. During candidates retrieval stage, searching will be only conducted on those hash tables which are targeted according to the features extracted from the query sample. Fingerprint indexing approaches [138, 76, 48, 199] adopt this invariant-based indexing technique for index space creation.

Continuous classification is generally applied on original feature space which is normally decimal value. An unsupervised learning algorithm *K-means* has been commonly chosen by researchers to partition feature space into a number of clusters. Each cluster can be considered as a hash table used in invariant-based indexing method. Fingerprint indexing approaches [95, 72, 172, 51, 49, 129, 131] adopt continuous classification for index space creation.

A typical ANN indexing method used for fingerprint indexing approach is Locality Sensitive Hashing (LSH) which is an approximate searching method. LSH algorithm assigns the feature vector into a number of hash tables using a set of hash functions. Each of these hash functions randomly selects components from the feature vector to calculate an index value which is used to determine which hash table will be used to store that feature vector. LSH mainly operates on binary feature vector, thus a binary transformation is generally required from the original feature space. LSH was selected by fingerprint indexing approaches in [62, 223, 161, 130]. An improved LSH called spherical LSH (S-LSH) was proposed by Wang et al. [202] in order to directly operate on original feature space.

Besides these three index space creation methods, some researcher proposed to build a tree structure hash table for indexing feature vectors. The idea of building a tree structure hash table was inspired by the indexing technique used in conventional files and relational databases [120, 42]. Table 2.1 and Table 2.2 list the fingerprint indexing approaches based on Category-1 and Category-2 respectively. Table 2.3 lists all fingerprint indexing approaches which have been classified into Category-3 based approaches and other approaches. It is hard to include all fingerprint indexing methods in these tables due to access restriction, thus some methods may be missing.

2.4.4 Performance metrics

The performance of fingerprint indexing algorithms is commonly reported as using a trade-off between *pre-selection error rate* and *penetration rate*. Some researchers also used *hit rate* [87] [96] (*correct index rate* [138] or *correct index power* (CIP) [86]) rather than the pre-selection error rate, while *pre-selection error rate* = 1 - *hit rate*. Without loss of generality, we refer to the definition of *pre-selection error rate*. In accordance with ISO/IEC 19795-1:2006 [101], a pre-selection error "occurs when the corresponding enrolment template is not in the preselected subset of candidates when a sample from the same biometric characteristic on the same user is given". Let's assume there are N_R reference samples enrolled in the database. N_P is the number of probe samples used for searching. The *pre-selection error rate* is calculated in Equation 2.1.

$$\textit{Pre-selection error rate} = \frac{N_{error}}{N_P} \tag{2.1}$$

where N_{error} is the number of the pre-selection errors. Obviously, $N_{error} \leq \mathrm{N}_P$.

As defined in ISO/IEC 19795-1:2006 [101] as well, the *penetration rate* "measures of the average number of pre-selected templates as a fraction of the total number of templates". In order to calculate this *penetration rate*, we make two assumptions: (1) H_i is the minimum number of retrieved candidates meanwhile the correct identifier for the probe sample P_i is included in these candidates; (2) $H_1 \leq H_2 \leq ... \leq H_{N_P}$.

Based on the investigation of calculating *penetration rate* in the literature, researchers interpret this concept from two perspectives and developed two types of formulations accordingly. The first interpretation is to calculate the *penetration rate* on average at a certain *Pre-selection error rate* [172, 87]. Equation 2.2 calculates a *penetration rate* corresponding to a *Pre-selection error rate* at $(N_{error})/N_P$.

$$\textit{Penetration rate} = (\frac{\sum_{i=1}^{(N_P - N_{error})} H_i}{N_R}) * \frac{1}{(N_P - N_{error})} \tag{2.2}$$

The second interpretation is to calculate the *Pre-selection error rate* at a certain *penetration rates* which is an opposite way comparing to the first one. Generally speaking, the *penetration rate* under this interpretation is the proportion of the total database that the system needs to search, and it is simply determined by the number of the retrieved candidates as seen in Equation 2.3. This interpretation was adopted by a number of researchers [86, 201, 91, 162, 138, 137, 126, 76, 227].

$$\textit{Penetration rate} = \frac{N_{max}}{N_R} \tag{2.3}$$

Where N_{max} is the number of retrieved candidates by a fingerprint indexing approach.

We think both interpretations of the *penetration rate* are capable of reporting the performance for the fingerprint indexing algorithm, while the second one is easier to understand and widely adopted in the published approaches. However, we observe that some researchers confused these two interpretations according to their experiment reports, as they chose only one of these two interpretation to calculate their results but comparing to the results calculated from both interpretations in different articles. In this dissertation, we reported the performance under second interpretation which corresponds to Equation 2.3 in order to be comparable with most of existing approaches.

Chapter 3

Contributions

3.1 Contributions

The research questions listed in Section 1.3 are the beacons to guide and conduct all research activities. The contributions of this dissertation can be formulated by answering these research questions. A list of publications follows this section.

RQ_1 **If we assume the smartphone's camera can facilitate the fingerprint sample capturing process, is it feasible to assess the quality of the fingerphoto taken by a general-purposed smartphone's camera in a real-life scenario?**

- While the vast majority of fingerprint quality assessment methods in the literature are designed for a fingerprint sample captured from the dedicated fingerprint sensors, we investigated the feasibility of the quality assessment approach to analyze the fingerphoto taken by the smartphones' cameras. Unlike the fingerprint sample captured from the dedicated fingerprint sensors has relatively clear background, the fingerphoto generally has a complicated background as well as suffering the defocusing issue. In addition, the hand motion and the large distance between the finger and the camera may lead to a very low quality sample. These challenges don't exist for the fingerprint sample captured from the dedicated fingerprint sensors. This is the reason that the conventional fingerprint quality assessment methods have the difficulty to qualify the fingerphoto, and motivated us to work on this topic.

  The conventional fingerprint quality assessment normally has a segmentation step before evaluating the quality. By considering the limited computation capability, we proposed a one-stop-shop fingerphoto quality assessment approach which doesn't require a segmentation step. The proposed approach divides the original fingerphoto into a set of non-overlapping blocks and classify each block either a high-quality block or a non high-quality block (background block or low-quality block). The quality of the fingerphoto is indicated by a final quality score which relies on the number of high-quality blocks. The performance of the proposed approach was reported in several forms including graphical demonstration, Spearman's rank correlation coefficient and false detection case. According to the experimental results, the proposed approach shows the capability to differentiate the high quality block from the background block and low quality block in a fingerphoto. In response to the research question, we would say that it is feasible to assess the quality of the fingerphoto taken by a general-purposed smartphone's camera in a real-life scenario

  In addition, a fingerphoto dataset was created using three popular smart phones under three different real-life scenarios. We also believe that the proposed approach can be applied to analyze the fingerphoto taken by other contactless cameras (such as webcam, digital camera), since the fingerphoto taken by these cameras (including smartphone's camera) has the same characteristics in terms of complex background, focusing issue and illumination condition. The details of these contributions have been presented in [217, 132, 128] (see Chapter 4, 5 and 6).

RQ_2 **Besides the existing feature extraction methods in literature, what features can still be extracted from the fingerprint template and outperform the existing ones?**

Three feature extraction methods were developed and used for different fingerprint indexing approaches. The first features extract method generates the feature vector consisting of 9 decimal values based on minutia and local ridge information. The second method extracts the feature vector with 24 decimal values only using minutia information. Since the minutia is considered as the most robust feature in the fingerprint template and standardised in ISO/IEC 19794-2 [99], a minutia based algorithm may have the advantage in terms of interoperability. The third method generates the binary string used for a fingerprint indexing approach, since operating on the binary string is potentially faster than operating on the decimal values when the efficiency requirement is critical for the large-scale fingerprint identification system. A brief introduction of these three feature extraction methods is described as follows:

- The first feature vector is composed of 9 components extracted from a triangle in a minutiae vicinity, where a minutiae vicinity is formed by a central minutia and its three closest minutiae. Four features are geometric traits calculated from the triangle. Three features represent orientation differences among three minutiae who form the triangle. Another two features are derived using ridge curvature and ridge density around the location of the minutia neighbors. This contribution has been presented in [129] (see Chapter 7).

- The second feature extraction method extracts 24 features from a minutiae vicinity only using minutia location and direction information. A self-alignment scheme is designed in each vicinity where 12 newly aligned minutiae will be generated based on this scheme. The first 12 feature of the proposed feature vector are calculated by using the location information of these 12 minutiae, and the rest of features are derived from these minutiae' directions. The details of this feature extraction method refers to Chapter 8 (as well seen in [131]).

- The third feature extraction method generates a fixed-length binary string from a minutiae-disk which consists of a central minutia and a number of surrounding minutiae. The number of the binary strings generated for a fingerprint relied on the number of minutiae in this fingerprint. Besides a offline training step, this method is composed of the following sequential steps: local alignment, quantization and binary string generation. The details of these steps have been presented in [130] (see Chapter 9).

  Three fingerprint indexing approaches have been developed based on these feature extraction methods. The performance of these approaches have been evaluated on a number of public datasets. Based on the experimental results, the proposed approaches show the improvement by comparing to a state-of-the-art fingerprint indexing approach.

RQ_3 **How to build the index space and retrieve candidate identities in a fingerprint indexing algorithm?**

- Classification-based index space creation approach is commonly used in the fingerprint indexing algorithm to enrol decimal feature vectors. We followed this technique, but improved it to better suit the features generated by our own approaches. The improvement is that each subject is only labelled to one cluster even if there are multiple feature vectors from the same subject assigned to the same cluster. We think this improvement has the ability to tolerate spurious minutiae. The details of this improvement have been presented in [129] (see Chapter 7). The second improvement is that we propose to divide the index space (or clusters) into four parts according to the minutia direction. We think this improvement can benefit the classification process, and the experimental results demonstrate the feasibility of our thoughts. This contribution has been presented in [131] (see Chapter 8).

RQ_4 **How to embed the privacy-preserving capability for the large-scale fingerprint identification system while still keeping the performance?**

Using the fingerprint template protection approach to achieve the privacy-preserving capability is a common thought as the researchers have proposed a variety of approaches on this topic. However, the fingerprint template protection approach normally encounters the performance degradation after satisfying the security requirements. This challenge motivated us to investigate the fingerprint template protection approach. Meanwhile, we are also aware that the fingerprint indexing algorithm is crucial for the large-scale fingerprint identification system, thus our main effort to answer this research question is to develop the fingerprint indexing algorithm which can protect the biometric data which still keeping the good performance. In the end, we developed a new fingerprint template protection approach and a fingerprint indexing algorithm with a security mechanism. A brief introduction of these two approaches is given as follows.

- A fingerprint template protection approach has been developed based on Bloom filters. When Bloom filters has been successfully adopted to protect iris and face biometrics, we explored the possibility of applying Bloom filters to fingerprint template. According to the experimental results, we conclude that it is feasible to apply Bloom filters on fingerprint template. We think this is the first attempt to apply Bloom filters on fingerprint. In addition, a pre-alignment mechanism was also designed in the proposed approach in order to generate a robust binary template. The details of this contribution has been described in [134] (see Chapter 10).

- A fingerprint indexing algorithm with a security mechanism is designed. The proposed approach is based on the fingerprint indexing algorithm developed in Chapter 9, but an encrypted module is proposed to protect the fingerprint template. This encryption module is embedded between local alignment module and binary feature generation module, thus the subsequent modules (binary feature generation, index space creation and candidate retrieval) are processing the data in an encrypted domain. No plain fingerprint data needs be stored in the database. The proposed encryption module is based on a block cipher encryption scheme which indicates that the security of the proposed fingerprint indexing algorithm is protected by the standard encryption algorithm. The performance was evaluated on both public datasets and a large-scale synthetic dataset, the results show that the proposed approach achieved the similar performance as the approach introduced in Chapter 9 which doesn't consider a security mechanism. Since the proposed approach only relies on the minutia information which has been standardised in ISO/IEC 19794-2 [99], it has the advantage in terms of interoperability. This contribution has been presented in Chapter 11.

3.2 List of publications

The following research articles are part of this dissertation.

1. Li G., Yang B., Olsen, M. A., Busch, C. (2013, June). Quality assessment for fingerprints collected by smartphone cameras. In Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on (pp. 146-153). IEEE.

2. Yang B., Li G., AND BUSCH C. "Qualifying fingerprint samples captured by smartphone cameras" Image Processing (ICIP), 2013 20th IEEE International Conference on. IEEE, 2013.

3. Li G., Yang B., Rathgeb, C. and Busch C., 2015, March. Towards generating protected fingerprint templates based on bloom filters. In Biometrics and Forensics (IWBF), 2015 International Workshop on (pp. 1-6). IEEE.

4. Li G., Yang B. and Busch C., 2014, March. A score-level fusion fingerprint indexing approach based on minutiae vicinity and minutia cylinder-code. In Biometrics and Forensics (IWBF), 2014 International Workshop on (pp. 1-6). IEEE.

5. Li G., Yang B. and Busch C., 2015, March. A Novel Fingerprint Indexing Approach Focusing on Minutia Location and direction. In Identity, Security and Behavior Analysis (ISBA), 2015 IEEE International Conference on (pp. 1-6). IEEE.

6. Li G., Yang B. and Busch C., 2015, September. A Fingerprint Indexing Scheme with Robustness against Sample Translation and Rotation. In Biometrics Special Interest Group (BIOSIG), 2015 International Conference of the (pp. 1-8). IEEE.

7. Li G., Yang, B., Raghavendra R. and Busch C., 2012. Testing Mobile Phone Camera Based Fingerprint Recognition under RealLife Scenarios. Norsk informasjonssikkerhetskonferanse (NISK), 2012.

8. Li G., Busch C. and Yang B., 2014, May. A novel approach used for measuring fingerprint orientation of arch fingerprint. In Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2014 37th International Convention on (pp. 1309-1314). IEEE.

9. Li G., Busch C. and Yang B., 2016, May. A fingerprint indexing algorithm on encrypted domain. Accepted by the 15th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (IEEE TrustCom-16).

Additional research articles are not part of this dissertation as listed below.

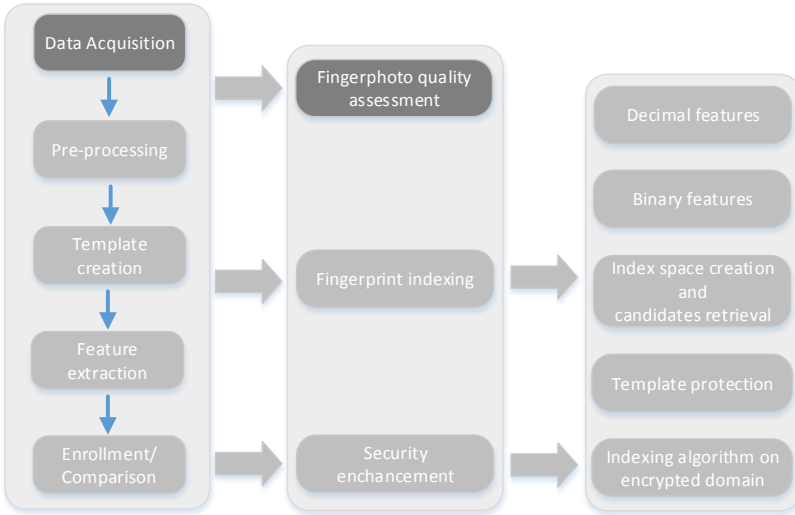
10. Li G., Yang B. and Busch C., 2013, July. Autocorrelation and dct based quality metrics for fingerprint samples generated by smartphones. In Digital Signal Processing (DSP), 2013 18th International Conference on (pp. 1-5). IEEE.

11. Li G., Yang B. and Busch C., 2013, October. Lightweight Quality Metrics for Smartphone Camera Based Fingerprint Samples. In Intelligent Information Hiding and Multimedia Signal Processing, 2013 Ninth International Conference on (pp. 342-345). IEEE.

12. Yang B., Chu H., Li G., Petrovic, S. and Busch C., 2014, March. Cloud Password Manager Using Privacy-Preserved Biometrics. In Cloud Engineering (IC2E), 2014 IEEE International Conference on (pp. 505-509). IEEE.

Part II

Fingerphoto Quality Assessment



This part is dedicated to the first research aspect 'fingerphoto quality assessment'. The work of this part intended to answer the first research question: RQ_1: **If we assume the smartphone's camera can facilitate the fingerprint sample capturing process, is it feasible to assess the quality of the fingerphoto taken by a general-purposed smartphone's camera in a real-life scenario?**

This part is composed of three chapters based on three research papers. The first paper in Chapter 4 developed a fingerphoto quality assessment approach which extracts 7 features from for each non-overlapping block divided from a fingerphoto. Based on these features, a binary decision is outputted by the proposed quality assessment approach where '1' indicates the block is a high-quality block and '0' implies either a low quality block or a background block. The total number of the high-quality blocks will qualify the quality of the captured sample.

Chapter 5 expanded the previous work to extract 12 features from a block divided from a fingerphoto, and a more extensive evaluation of the proposed approach is also given. Chapter 6 gives a thorough discussion and analysis of our proposed fingerphoto quality assessment approach. We elaborated the motivation, methodology and observations of the proposed approach. These detailed information reveals the logic behind the proposed approach to answer two general questions: why and how we extracts those 12 features. In addition, a quality score normalization method is designed, and a comprehensive experiment was conducted with the comparison to two existing quality metrics used for fingerprint image.

The work in Chapter 4 was published in [217]: GUOQIANG LI, BIAN YANG, CHRISTOPH BUSCH. "Qualifying fingerprint samples captured by smartphone cameras" Image Processing (ICIP), 2013 20th IEEE International Conference on. IEEE, 2013.

The work in Chapter 5 was published in [132]: GUOQIANG LI, BIAN YANG, MARTIN A. OLSEN, CHRISTOPH BUSCH. "Quality assessment for fingerprints collected by smartphone cameras". In Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on (pp. 146-153). IEEE.

The work in Chapter 6 was published online: GUOQIANG LI, BIAN YANG, CHRISTOPH BUSCH. "Qualifying fingerprint samples captured by smartphone cameras in real-life sce-

narios". Published on https://brage.bibsys.no/xmlui/handle/11250/2388306, ISBN: 978-82-8340-040-3.

Chapter 4

Qualifying Fingerprint Samples Captured by Smartphone Cameras

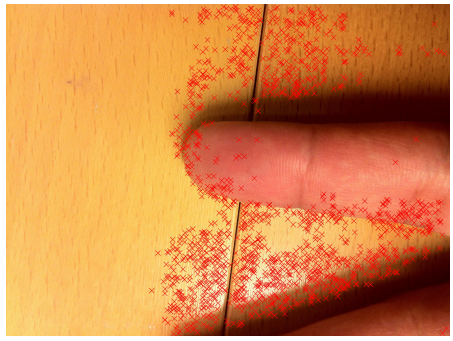
Abstract

This paper proposes an approach to qualify fingerprint samples captured by smartphones' cameras under real-life scenarios, foreseeing the future application using such general purposed cameras as fingerprint sensors. In this approach, a sample image is first divided into non-overlapping blocks. Then a 7-dimensional feature vector will be formed from the proposed 7 quality features. We use a trained support vector machine to produce a binary indication for each image block on its quality. Finally a quality score is generated to indicate the whole fingerprint sample's quality by counting the number of qualified blocks in a sample. Experiments demonstrate the proposed approach's capability of qualifying such quality-challenging fingerprint samples - the Spearman's rank correlation coefficient ρ ($-1 \leq \rho \leq 1$) between the proposed quality metric and samples' normalized comparison scores reaches as high as 0.53 in our experiment.

4.1 Introduction

Fingerprint recognition is widely used in commercial and forensic areas, for which professional sensors are usually chosen to acquire data under a controlled environment compliant to existing standards [97, 100]. Recently some researches [73, 160, 218, 186] have shown it is possible to use smartphones' cameras as an alternative to implement the fingerprint recognition functionality. Such smartphone cameras have advantages in deployment cost, users' convenience, and even privacy protection ($i.e.$, samples are captured by a user's own device). However, the quality of samples captured by such general-purposed cameras is unstable under real-life scenarios [133] due to de-focusing, poor illumination, complicate background noise, and camera motion during the picture-taking process. Therefore, it is essential to conduct sample quality control before enrolling or verifying a sample taken by a smartphone camera.

We propose an approach to assess the quality of fingerprint samples captured by smartphones' cameras. The approach produces a quality score for a fingerprint sample to predict the sample's utility in terms of recognition performance, and then the camera can decide either to adopt the current sample or to automatically adjust the camera settings for the next sample capturing. Most of existing fingerprint quality assessment methods, including the NIST Fingerprint Image Quality (NFIQ) [74], mechanisms [93, 98], are designed for samples generated by professional fingerprint sensors, and thus hard to accurately assess the quality of samples captured by smartphone cameras [218]. Figure 4.1 gives two samples which are considered as high quality ('score 1' by NFIQ) but with a lot of spurious minutiae detected on background. The rest of the paper is organized as follows: Section 4.2 describes the proposed approach; Section 4.3 gives experimental results; and Section 4.4 concludes this paper.



(a) Minutiae extracted by NIST's NBIS function mindtct [25]

(b) Minutiae extracted by NeuroTechnology VeriFinger6.0

Figure 4.1: Two samples measured as high quality by NFIQ with red cross indicating the minutiae extracted by NIST's NBIS function mindtct and NeuroTechnology VeriFinger 6.0 respectively.

4.2 The proposed approach

4.2.1 The general process

To assess the quality of a sample, the sample image I is divided into N non-overlapped image blocks. A set of quality metrics are employed to indicate for each image block a binary quality decision d_i (qualified, labelled by "1"; or not qualified, labelled by "0") and then use the counts of the qualified blocks as the quality score for the sample. The whole process is illustrated in Figure 4.2, where a support vector machine (SVM) is trained over manually-cropped image blocks (as ground truth) labelled as high-quality ones and non-high-quality ones, and used during assessment as a classifier to output the binary decision on the quality of the investigated block. The sum over d_i in a sample shall be used as the global sample's quality score:

$$S_I = \sum_{i=1}^{N} d_i \tag{4.1}$$

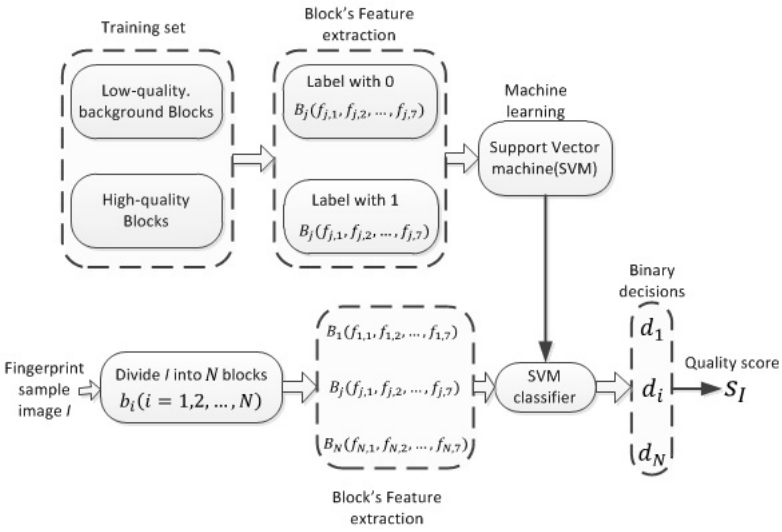


Figure 4.2: Processes of the proposed approach

4.2.2 Proposed quality features

4.2.2.1 Image block alignment along ridge orientation

As mentioned in the above, a sample image I is divided into non-overlapped image blocks $\mathbf{b}_i^0(i = 1, 2, ..., N)$ sized $R \times C$ in pixel. Before extracting quality features for quality assessment, we align such image blocks along the ridge orientation, assuming a high-quality block is a fingerprint-ridge-like pattern. The PCA based gradient orientation estimation method [45] is used to find a block's principle orientation. That is, inside each block $\mathbf{b}_i^0$ neighboring pixels differences d_v and d_h (in vertical and horizontal directions respectively) are obtained to form a gradient vector with orientation $tan^{-1}(d_v/d_h)$. Then the block principle gradient orientation θ_i is calculated by exploiting PCA to find the principle one among all orientations of the $(R-1) \times (C-1)$ calculated gradient vectors. Now by clock-wisely rotating the $\sqrt{2}(R-1) \times \sqrt{2}(C-1)$ size area concentric to $\mathbf{b}_i^0$ by angle θ_i we can crop a block $\mathbf{b}_i$ sized $R \times C$ concentric to $\mathbf{b}_i^0$. In this way we assume $\mathbf{b}_i$ has the maximum gradient value in the horizontal direction.

4.2.2.2 Quality features for block quality assessment

We propose 7 quality features $f_i(i = 1, 2, ..., 7)$ in four categories to assess an image block $\mathbf{b}_i$'s quality: (1) Gray-scale values statistics; (2) fingerprint minutiae feature (3) autocorrelation based features; and (4) frequency features from the autocorrelation result. The details are as follows.

1. Gray-scale values statistics

(1) f_1: Exposure (a block's gray scale, which should not be dim nor very bright). Denote the average pixel value of $\mathbf{b}_i$, i.e.

$$f_1 = \frac{1}{R \times C} \sum_{r=1}^{R} \sum_{c=1}^{C} b_i(r, c) \tag{4.2}$$

where $b_i(r, c)$ is the pixel value at the r-th row and c-th column inside the block $\mathbf{b}_i$.

(2) f_2: Certainty of the block principle gradient orientation. We use a modified definition of *ocl* (orientation certainty level) in [104] as follows:

$$f_2 = \begin{cases} 1 - \frac{\lambda_2}{\lambda_1} & \text{if } \lambda_1 \neq 0 \\ 0 & \text{if } \lambda_1 = 0 \end{cases} \tag{4.3}$$

where λ_2 is the second eigenvalue of the covariance matrix of all gradient vectors in the PCA calculation.

2. Autocorrelation based features

Considering the fact that $\mathbf{b}_i$ has the principle gradient orientation aligned to the horizontal direction, autocorrelation calculation along the horizontal direction of $\mathbf{b}_i$ could be useful to enhance the dominant spatial frequencies and thus suitable for quality feature extraction. Instead of calculating autocorrelation directly, we do the autocorrelation on the horizontally-differential vectors $\mathbf{d}_i(r), 1 \leq r \leq R-1$. The details are as follows:

$$\mathbf{acr}_i = \sum_{r=1}^{R-1} autocorr(\mathbf{d}_i(r)) \tag{4.4}$$

where $\mathbf{d}_i(r) = (b_i(r,2) - b_i(r,1), b_i(r,3) - b_i(r,2), ..., b_i(r,C) - b_i(r,C-1))$.

The resultant $\mathbf{acr}_i$ is the $(C-1)$-dimensional sum-up vector with each row's autocorrelation calculated as follows:

$$autocorr(\mathbf{d}_i(r))(j) = \sum_{c=1}^{C-1} d_i(r,c) d_i(r,c+j) \tag{4.5}$$

where $(0 \leq j \leq C-2)$, with all $(C-1)$ amplitudes divided by the highest amplitude of $autocorr(\mathbf{d}_i(r))$. Before feature extraction, low-pass filtering by setting zero the higher half of DCT-transform frequencies is used to smoothen the autocorrelation resultant vector. We denote the $(C-1)$ dimensional vector after the low-pass filtering as $\mathbf{ACR}_i$.

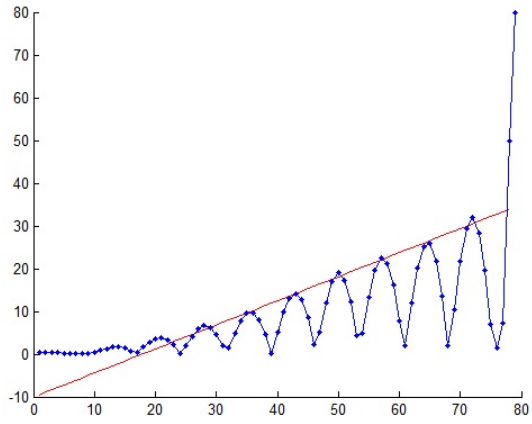


Figure 4.3: $\mathbf{ACR}_i$ curve (red straight line to fit the peak points)

(3) f_3: $\mathbf{ACR}_i$'s peak activity rate. From the observations in the experiments, we find the peaks of the $\mathbf{ACR}_i$ curve have a stable increasing rate if the sample quality is good enough. We use the 1-order polynomial (i.e. a straight line) to fit the M detected peak points with the x-coordinates $P_1(x), P_2(x), ..., P_M(x) (M) \ll C-1$ in the $\mathbf{ACR}_i$ curve (as seen from Figure 4.3) and obtain a straight line with slope S and M amplitudes $A(P_n(x))(n = 1, 2, ..., M)$.

Then the $\mathbf{ACR}_i$'s peak activity rate is defined as

$$f_3 = \frac{1}{M}\sum\nolimits_{n=1}^{M} A(P_n(x))(A(P_n(x)) > 0). \tag{4.6}$$

(4) f_4: $\mathbf{ACR}_i$'s peak pick-up rate. We denote it as the slope S directly:

$$f_4 = S \tag{4.7}$$

3. Frequency features from the autocorrelation result. This category of features are derived from the frequency characteristics of the FFT coefficients of $\mathbf{ACR}_i$, which we denote as $f\mathbf{ACR}_i$. Frequency features can be useful to represent the ridge spatial frequency characteristics.

(5) f_5: Principle frequency's dominance rate.

$$f_5 = \frac{4 \times \sum_{n=2}^{C/4} Q_i(n)}{(C-4) \times Q_i(1)} \tag{4.8}$$

where we denote $Q_i(1), Q_i(2), ..., Q_i(C/4)$ as the first quarter of $f\mathbf{ACR}_i$'s components sorted by descending amplitude.

(6) f_6: Principle frequency's prominence rate.

$$f_6 = \frac{\sum\limits_{n=-X}^{X} f\mathbf{ACR}_i(L+n) - \sum\limits_{n=-H}^{H} f\mathbf{ACR}_i(L+n)}{2(X-H) \times P_i(1)} \tag{4.9}$$

where L is the principle frequency's index in the vector $f\mathbf{ACR}_i$, $0 < H < X < C$. And $L - X > 0$, otherwise $f_6 = 0$.

4. Local features

Besides the above global features, which involve all pixels in a block for calculation, we may use some local features for quality assessment as well. The standard fingerprint minutiae feature [99] can be exploited for this purpose.

f_7: Number of minutiae detected from an image block. This feature could be useful to distinguish those real fingerprint ridge areas from those high frequency noise background, assuming a real fingerprint ridge block contains only limited number of minutiae.

4.2.2.3 Feature dynamic range normalization

All the 7 features $f_i (i = 1, 2, ..., 7)$ are normalized in their dynamic range by Z-score before being fed to the SVM:

$$f_i^{'} = \frac{f_i - E(f_i)}{\sigma(f_i)} \tag{4.10}$$

where $E(f_i)$ and $\sigma(f_i)$ are the expectation and standard deviation values of the feature f_i.

4.3 Experimental design and results

4.3.1 Experimental set up

We selected three mobile phones - iPhone 4, Samsung Galaxy S, and Nokia N8 - to capture fingerprint samples from 100 different fingers from 25 subjects. Three typical scenarios are defined: in-door scenario in a good illumination but with a challenging desk textural surface (as seen in Figure 4.1 and Figure 4.4a), darkness scenario with illumination only from the smartphone's flash (as seen in Figure 4.4b), and the out-door scenario with complicate background (as seen in Figure 4.4c). All three phones were used to capture three samples for each finger in the in-door and the out-door scenarios, but only Nokia N8 was used in

(a) In-door

(b) Darkness

(c) Out-door

Figure 4.4: Fingerprint samples under different scenarios.

Table 4.1: EER value of different cameras using Verifinger 6.0 for template generation and comparisons.

Computing scenario	All	Nokia	iPhone	Samsung
Number of reference images	73	63	31	23
Number of probe images	351	193	44	44
EER	16.9%	4.3%	2.39%	5.37%

the darkness scenario (the other two failed to take picture in the darkness). In total, there are 2100 fingerprint images captured in the experiment.

In order to generate the normalized comparison scores defined by NIST to measure the sample quality [75] as a ground truth for sample quality (which is however not suitable for quality prediction because its calculation involves all the samples offline), cropping the foreground (finger area) from the raw images has been conducted first to obtain the ground-truth comparison performance of the data set. There are 424 samples from 73 fingers that can successfully generate templates by NeuroTechnology VeriFinger 6.0, and for each finger we captured 2 samples in order to compute comparison score. Equal Error Rates (EERs) are computed across all of the cameras and intra-cameras respectively as shown in Table 4.1. Since some of fingers have only one sample for a specific cell phone's camera, the sum of 63+31+23+193+44+44 is less than the sample number 424 in Table 4.1. These EERs indicate it's feasible to implement fingerprint recognition on mobile phone. All of our experiments are based on these 424 samples. Our quality assessment approach addresses the full images captured from cameras. The samples generated by Samsung Galaxy S camera have been enlarged 1.5 times before quality assessment to achieve largely the same resolution as the samples from the other two. The parameters used in our experiments were set as: $R = C = 80$, $H = 2$, $X = 4$.

For the fingerprint minutiae feature, NIST's NBIS function mindtct [25] is used to detect minutiae in an image block, in light of the fact that NeuroTechnology VeriFinger 6.0 failed to extract minutiae on the vast majority of training blocks in our experiment, probably due to its own functionality of sample quality checking while doing minutiae detection.

4.3.2 Quality prediction performance

We analyze the correlation between the normalized comparison score c_i and the proposed quality score q_i for each sample x_i. Comparison scores are produced by NeuroTechnology VeriFinger 6.0 comparator and we use the samples with maximum sum of intra-finger sample comparison scores as references for enrolment in recognition performance testing. In order to include these reference samples into the correlation calculation, we assign the largest comparison score found in the calculation to all the reference samples. After calculating the normalized comparison score c_i as a ground-truth quality indicator (due to its

Table 4.2: Spearman's rank correlation coefficient ρ of block features with the block quality decision.

Features	1^{st}	2^{nd}	3^{rd}	4^{th}	5^{th}	6^{th}	7^{th}
ρ	0.05	0.1	0.3	0.30	0.28	0.36	0.06

representation of error rates) for each sample, we can use it to calibrate the quality predicting performance of the proposed quality features.

At last, the 424 pairs (c_i, q_i) are obtained. We quantize the quality scores into 8 bins and calculate the average value of the normalized comparison scores in each quality score bin, as shown in Figure 4.5. The graph shows good correlation between the quality score (thus the proposed quality metrics) and the normalized comparison scores.

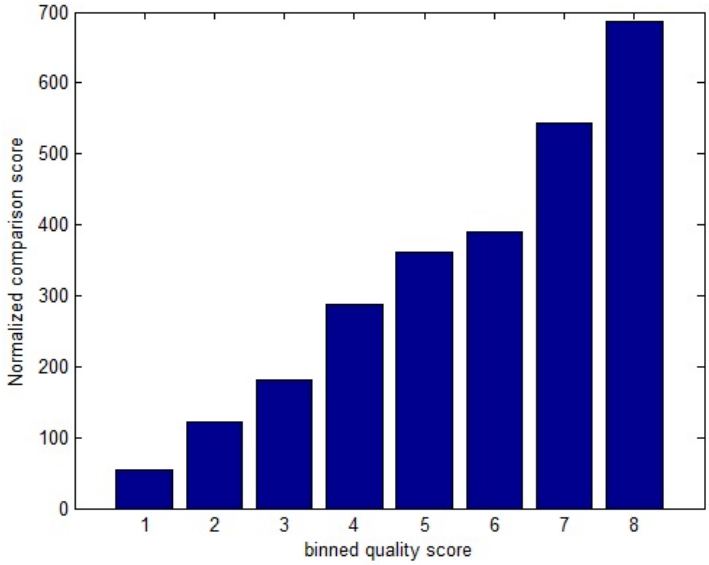


Figure 4.5: Normalized comparison scores in 8 quality bins.

4.3.2.1 Spearman's rank correlation coefficient

Spearman's rank correlation coefficient $\rho(-1 \leq \rho \leq 1)$ can measure how well two variables correlate, $i.e.$, in our case how well our propose quality metric correlates to the observed biometric performance ($i.e.$ utility of the sample). It is computed as 0.5346 on the 424 pairs (c_i, q_i) we tested, which demonstrates our approach's effectiveness to distinguish the high-quality fingerprint samples from the poor quality or background ones. We also computed this coefficient between each feature and the block quality decision, given in Table 4.2.

4.3.3 False detection statistics

Since the desk's texture in the in-door scenario looks similar to fingerprint ridges in our experiments, there are still some samples that cause a false detection (background blocks

Table 4.3: EER under different levels of quality score

	Low	Medium	High
EER	21.95%	9.44%	3.86%

qualified). Especially, the number of falsely qualified blocks is 278 without using the feature f_7 which is the count of the minutiae in a block. This number can be reduced to 187 using the feature f_7, and it in total accounts for only 2.7% of all 6874 qualified blocks from these 424 samples.

4.3.4 Equal-Error-Rate (EER) under different levels of quality score

We divided the 424 samples into 3 groups by quality score assigned to each sample: low quality group (0-10), medium quality group (10-20) and high quality group (>20). We select the sample with maximum quality score as the reference sample for each finger. The EERs for the three groups are shown in Table 4.3. The results shows that the EER decreases significantly when the quality score increases. This is a further proof that our approach is useful in identifying high-quality samples to achieve better recognition performance.

4.4 Conclusion

A effective approach is proposed in this paper to predict the quality of samples captured by the smartphones' cameras under uncontrolled real-life scenarios. Experimental results demonstrated the approach's accuracy in assessing the quality of such samples which are considered challenging by traditional fingerprint quality assessment methods.

Chapter 5

Quality Assessment for Fingerprints Collected by Smartphone Cameras

Abstract

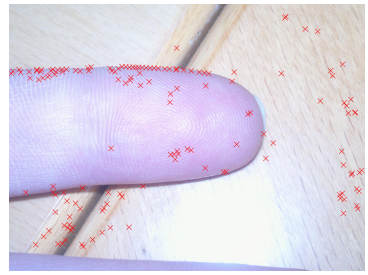
We propose an approach to assess the quality of fingerprint samples captured by smartphone cameras under real-life scenarios. Our approach extracts a set of quality features for image blocks. Without needing segmentation, the approach determines a sample's quality by checking all image blocks divided from the sample and for each block a trained support vector machine gives a binary indication - "high-quality" or "non-high-quality" (including the low quality case and the background block case). A quality score is then generated for the whole sample. Experiments show this approach performs well in identifying the high quality blocks - the Spearman correlation coefficient between the proposed quality scores and samples' normalized comparison scores (ground truth) reaches 0.53 while the rate of false detection (background blocks judged as high-quality ones) is still low as 4.63 percent over a challenging dataset collected under various real-life scenarios.

5.1 Introduction

Fingerprint recognition has been widely used in industry and forensic area. It is quite common to select the dedicated sensors to acquire biometric samples in a controlled environment compliant to existing standards [97, 100]. However, smartphones are being found in almost everyone's pocket nowadays and normally embedded with a 5-mega-pixels (or above) camera, it becomes feasible to use these general-purposed cameras for capturing fingerprint samples. Previous research [73, 160, 218, 186] has shown that it is feasible to implement the fingerprint recognition functionality using smartphones' cameras as an alternative to dedicated fingerprint sensors. Compared to the quality of the fingerprint samples captured under the ideal laboratory environment, the sample quality is quite unstable while data acquisition takes place under a real-life scenario [133] due to camera motion, de-focusing, poor illumination and complicated backgrounds. Thus it is essential to assess the sample quality before implementing practically useful biometrics-enabled applications on these smartphone cameras.

Several quality assessment methods and mechanisms have been proposed in the literature, such as [74, 93, 98], but they are designed for samples generated by the dedicated fingerprint sensors. There are two fingerprint samples shown in Figure 5.1 which are considered as high quality ('score 1' by NIST Fingerprint Image Quality (NFIQ) [74]) but with a lot of spurious minutiae detected on the background. These methods are not designed to cope with fingerprint samples captured by smartphone cameras [218] with so complicated environments requiring accurate segmentation and noise (variance in lighting and color) suppression of the foreground (finger area). Consequently, their simple pre-processing mechanisms ($e.g.$, the quality map used in NFIQ to identify foreground blocks) which were accustomed to contact-based fingerprint patterns are not capable towards such contactless-based samples any more.

We propose a segmentation-free approach in this paper to assess the quality of fingerprint samples captured by the smartphones' cameras under real-life scenarios. A critical



(a) by NIST MINDTCT

(b) by VeriFinger 6.0

Figure 5.1: High quality samples detected by NFIQ (score 1) with red cross indicating the detected minutae.

challenge during taking photo under real-life scenarios is the unpredictable background which may cause false detection of the finger area. Instead of using pixel-level foreground (finger area) segmentation, which could be both inaccurate and high in computational complexity for a mobile device, the approach checks each image block's quality status - high quality or non-high quality ($i.e.$, the low quality and the background cases) - and combines all blocks' quality decisions to produce the final quality score for the sample. This quality score can be adopted to predict the sample's utility in terms of recognition performance, and then the camera can decide either to store the samples (if the quality score is large enough) or to automatically adjust the camera settings (such as the focusing distance or flash) for the next sample capture. The remaining sections are organized as follows: Section 5.2 presents the proposed approach; Section 5.3 shows experimental results; and Section 5.4 concludes this paper.

5.2 The proposed approach

5.2.1 Processes of the proposed approach

A conventional quality assessment usually include two steps: fingerprint area segmentation and quality prediction of the fingerprint area. Instead, we propose a one-step quality assessment approach which will not differentiate the foreground (fingerprint area) from the background in view of the computational efficiency and low-memory consumption requirement to mobile phones.

Figure 5.2 illustrates the processes of the proposed approach which uses support vector machine (SVM) to generate a quality binary decision d_i (1 = high quality; 0 = non-high quality) for each block $b_i (i = 1, 2, ..., n)$ divided from a sample image I. The SVM classifier

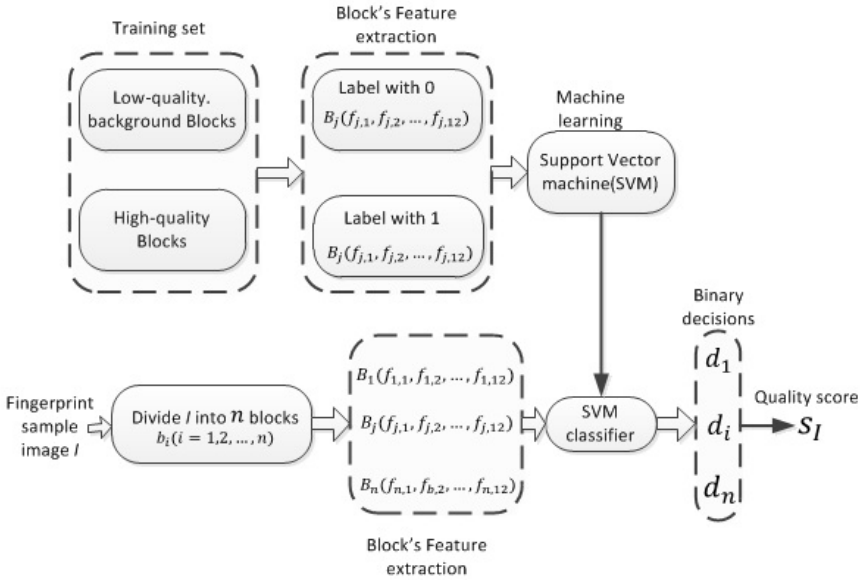


Figure 5.2: Processes of the proposed approach.

is trained from a ground truth data set composed of high-quality block features with label one and non-high-quality block features with label zero to obtain a binary classifier. During quality assessment, we use the trained classifier to predict if each input block b_i divided from sample image I should be classified as high-quality or not. A global quality score S_I is generated to indicate the whole fingerprint sample's quality by counting the number of blocks labelled as high quality. To make this quality indicator more accurate, sample images can be resized to offset the variance in the finger-to-camera distance, as we did in Section 5.3.1.2.

5.2.2 Proposed quality features

5.2.2.1 Image block alignment in ridge orientation

A sample image is divided into non-overlapping blocks $\mathbf{b}_i^0 (i = 1, 2, ..., N)$ sized $R \times C$ in pixel (R and $C = 2^k, k = 1, 2, 3, ...$) on which the quality features are computed. Before computing the features, the image blocks are aligned according to their ridge orientation using the PCA based gradient orientation estimation method [45]. That is, inside each block $\mathbf{b}_i^0$ neighbouring pixels' differences d_v and d_h (in both vertical and horizontal directions respectively) are obtained to form a gradient vector with orientation $tan^{-1}(d_v/d_h)$. Then the principal component analysis θ_i is calculated by exploiting PCA to find the principal one among all orientations of the $(R-1) \times (C-1)$ calculated gradient vectors. Now by clock-wise rotating the $\sqrt{2}(R-1) \times \sqrt{2}(C-1)$ size area concentric to $\mathbf{b}_i^0$ by angle θ_i we can crop a block $\mathbf{b}_i$ sized $R \times C$ concentric to $\mathbf{b}_i^0$. In this way we assume $\mathbf{b}_i$ has the maximum gradient value in the horizontal direction.

5.2.2.2 Quality features for block quality assessment

We propose 12 quality features $f_i (i = 1, 2, ..., 12)$ in three categories to assess an image block's quality: (1) pixel based features; (2) autocorrelation based features; and (3) frequency features from autocorrelation result. And the details are as follows.

1. Pixel based features

(1) f_1: Exposure (a block's gray level). Denote the average pixel value of $\mathbf{b}_i$,

$$f_1 = \frac{1}{R \times C} \sum\nolimits_{r=1}^{R} \sum\nolimits_{c=1}^{C} b_i(r, c) \tag{5.1}$$

where $b_i(r, c)$ is the pixel value at the r-th row and c-th column inside the block $\mathbf{b}_i$.

(2) f_2: Significance of the principal component analysis. We represent it using the first eigenvalue λ_1 of the covariance matrix of all gradient vectors in the PCA calculation.

(3) f_3: Certainty of the block principal gradient orientation. We use a modified definition of *ocl* (orientation certainty level) in [104] as follows:

$$f_3 = \begin{cases} 1 - \frac{\lambda_2}{\lambda_1} & \text{if } \lambda_1 \neq 0 \\ 0 & \text{if } \lambda_1 = 0 \end{cases} \tag{5.2}$$

where λ_2 is the second eigenvalue of the covariance matrix of all gradient vectors in the PCA calculation.

2. Autocorrelation based features

Considering the fact that $\mathbf{b}_i$ has the principal gradient orientation aligned to the horizontal direction, autocorrelation calculation along the horizontal direction of $\mathbf{b}_i$ could be useful to enhance the dominant spatial frequencies and thus the autocorrelation result can be used for quality feature extraction. Instead of calculating autocorrelation directly, we do the autocorrelation on the horizontally-differential vectors $\mathbf{d}_i(r), 1 \leq r \leq R-1$. The details are as follows:

$$\mathbf{acr}_i = \sum\nolimits_{r=1}^{R-1} autocorr(\mathbf{d}_i(r)) \tag{5.3}$$

where $\mathbf{d}_i(r) = (b_i(r, 2) - b_i(r, 1), b_i(r, 3) - b_i(r, 2), ..., b_i(r, C) - b_i(r, C-1))$.

The resultant $\mathbf{acr}_i$ is the $(C-1)$-dimensional sum-up vector with each row's autocorrelation calculated as follows:

$$autocorr(\mathbf{d}_i(r))(j) = \sum_{c=1}^{C-1} d_i(r, c) d_i(r, c+j) \tag{5.4}$$

where $(0 \leq j \leq C-2)$, with all $(C-1)$ amplitudes divided by the highest amplitude of $autocorr(\mathbf{d}_i(r))$. Before the follow-up feature extraction, low-pass filtering by setting zero the higher half of DCT-transform frequencies is used to smoothen the autocorrelation resultant vector. We denote the final $(C-1)$ dimensional vector as $\mathbf{ACR}_i$.

(4) f_4: $\mathbf{ACR}_i$'s peak active rate. From the observations in the experiments, we find the peaks of the $\mathbf{ACR}_i$ curve have a stable increasing rate if the sample quality is good enough. We use the 1-order polynomial (a straight line) to fit the M detected peak points with the x-coordinates $P_1(x), P_2(x), ..., P_M(x) (M) \ll C-1$ in the $\mathbf{ACR}_i$ curve (shown in Figure 5.3) and obtain a straight line with slope S on which M amplitudes $A(P_n(x))(n = 1, 2, ..., M)$ can be found. Then the $\mathbf{ACR}_i$'s peak active rate is defined as

$$f_4 = \frac{1}{M} \sum\nolimits_{n=1}^{M} A(P_n(x))(A(P_n(x)) > 0). \tag{5.5}$$

(5) f_5: $\mathbf{ACR}_i$'s peak pick-up rate. We denote it as the slope S directly:

$$f_5 = S \tag{5.6}$$

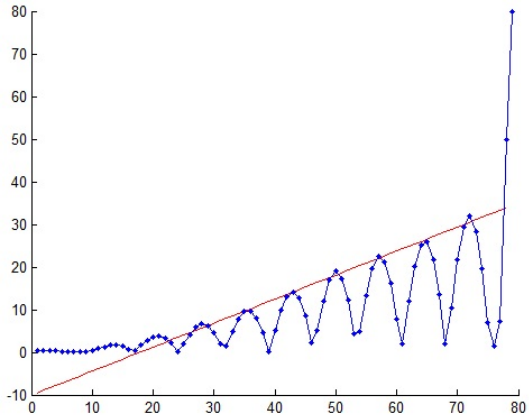


Figure 5.3: $\mathbf{ACR}_i$ curve, C=80 (the straight line is the linear best fit of the M peak points).

(6) f_6: $\mathbf{ACR}_i$'s peak variance rate. We use this rate to represent the degree the M amplitudes $A(x)$ on the fitted line diverge from the actual M peak amplitudes $P_1(y), P_2(y), ..., P_M(y)$

$$f_6 = UP/DOWN. \tag{5.7}$$

where, $UP = (\frac{1}{M}\sum_{n=1}^{M}|P_n(y) - A(P_n(x))|)$
$DOWN = (max(A(P_n(x))) - min(A(P_n(x))))$.

(7) f_7: $\mathbf{ACR}_i$'s peak drop rate. We use this rate to represent the degree of the amplitude drop $AD_j = P_{j+1}(y) - P_j(y)(j = 1, 2, ..., M-1)$ of one peak compared to its previous counterpart. From the observation in the experiments, large drops in amplitude seldom happen to high quality blocks.

$$f_7 = UP/DOWN \tag{5.8}$$

where, $UP = 1 - (\sum_{j=1}^{M-1}|AD_j(AD_j < 0)|/(M-1))$
$DOWN = (max(A(P_n(x))) - min(A(P_n(x))))$.

3. Frequency features from the autocorrelation result

This category of features is derived from the frequency characteristics of the FFT coefficients of $\mathbf{ACR}_i$, which we denote as $f\mathbf{ACR}_i$ Frequency features are useful to represent the ridge spatial frequency characteristics.

(8) f_8: Principal frequency's amplitude:

$$f_8 = max(abs(f\mathbf{ACR}_i)) \tag{5.9}$$

(9) f_9: Principal frequency's index in vector $f\mathbf{ACR}_i$.

(10) f_{10}: Principal frequency's dominance rate.

$$f_{10} = \frac{4 \times \sum_{n=2}^{C/4} Q_i(n)}{(C-4) \times Q_i(1)} \tag{5.10}$$

where we denote $Q_i(1), Q_i(2), ..., Q_i(C/4)$ as the first quarter of $f\mathbf{ACR}_i$'s components sorted by descending amplitude.

(11) f_{11}: Principal frequency's prominence rate – close neighbours.

$$f_{11} = \frac{(\sum_{n=-H}^{H} f\mathbf{ACR}_i(L+n)) - f\mathbf{ACR}_i(L)}{2H \times f\mathbf{ACR}_i(L)} \tag{5.11}$$

where L is denoted as the feature f_9 that is the principal frequency's index in the vector $f\mathbf{ACR}_i$. We consider $2H$ neighbours around the principal frequency.

(12) f_{12}: Principal frequency's prominence rate – second close neighbours:

$$f_{12} = \frac{\sum_{n=-X}^{X} f\mathbf{ACR}_i(L+n) - \sum_{n=-H}^{H} f\mathbf{ACR}_i(L+n)}{2(X-H) \times f\mathbf{ACR}_i(L)} \tag{5.12}$$

where L is the principal frequency's index in the vector $f\mathbf{ACR}_i$, $0 < H < X < C$. And $L - X > 0$, otherwise $f_{12} = 0$.

5.2.2.3 Feature dynamic range normalization

All the 12 features $f_i (i = 1, 2, ..., 12)$ are z-score normalized prior to being used by the SVM as:

$$f_i^{'} = \frac{f_i - E(f_i)}{\sigma(f_i)} \tag{5.13}$$

where $E(\bullet)$ and $\sigma(\bullet)$ are expectation and standard deviation values of the feature f_i.

5.3 Experimental design and results

Good sample quality can be represented by its high normalized comparison scores [74]. We evaluate in this section how well the normalized comparison score, as the ground truth, and the quality score generated from the proposed approach correlate. Spearman's rank correlation coefficient [153] is computed between the two for each sample.

5.3.1 Experimental setup

5.3.1.1 Data collection and experimental settings

Three smart phones - iPhone 4, Samsung Galaxy S, and Nokia N8 - were selected to capture fingerprint samples from 100 different fingers of 25 groups (corresponding to 25 subjects) of right index finger, right middle finger, left index finger and left middle finger. Table 5.1 lists the specification of selected mobile phone cameras. Three scenarios are tested: indoor scenario with good illumination but challenging background with similar color and texture as fingers (shown in Figure 5.1); dark scenario with illumination only from the smartphone automatic flash; outdoor scenario with complex background such as buildings, lawns, lakes and trees. Figure 5.4 shows the finger examples generated in the three scenarios respectively. We used each phone to capture three samples for each finger in the first and third scenarios, but only Nokia N8 in the second scenario (the other two failed to take photos in darkness). In total, there are 2100 fingerprint samples captured. For quality assessment, the parameters used in our experiments were set as: $R = C = 80$, $H = 2$, $X = 4$ (refer to Section 2.2). 100 high-quality blocks and 200 non-high-quality ones (visually judged as ground truth) with size $R \times C$ were randomly cropped from samples for SVM training.

5.3.1.2 Sample pre-processing

Although our proposed approach does not need to segment the foreground (finger area) for quality assessment, in practice for recognition purpose pre-processing are usually needed over the samples directly output from the cameras. Such pre-processing steps could include (1) segmentation of the fingerprint area; (2) sample resizing (to offset the distance variance of fingers from the camera); and (3) fingerprint area enhancement. Instead of performing such pre-processing steps for quality assessment, we need to do them in this paper to calculate the normalized comparison score [74] of each sample to obtain the ground truth

Table 5.1: Specification of the 3 smartphones' cameras.

Mobile phone	Nokia N8	iPhone 4	Samsung Galaxy S
Mega pixel	12.0	5.0	5.0
Resolution	1536×1936	2592×1936	1600×960
Auto-focus	Yes	Yes	Yes
Image format	JPEG	JPEG	JPEG
ISO control	automatic	automatic	automatic
Flash source	Xenon	LED	no flash
Flash setting	automatic	automatic	no flash
Aperture	f/2.8	f/2.8	f/2.6
Sensor size	1/1.83″	1/3.2″	1/3.6″

(a) (b) (c)

Figure 5.4: Fingerprint samples under 3 different scenarios: (a) In-door, (b) Dark(auto flash) and (c) Out-door.

of the sample's quality. We tested two types of pre-processed samples in our experiments as follows.

Pre-processing Type 1. Segmentation only, in which only manually segmentation is performed to crop the foreground, without applying sample resizing and enhancement. This type provides the baseline condition for normalized comparison score calculation.

Pre-processing Type 2. Manual segmentation of the foreground, foreground resizing, and foreground enhancement. Resizing is realized by the following steps: (1) fitting the finger-tip shape as a half-circle, detect this finger-tip circle using Hough transform over the boundary of the foreground; (2) align the radius of the detected finger-tip half-circle to a fixed value (20 pixels in our experiments); and (3) resize the whole cropped sample according to the new aligned radius value. In this way, all the resized samples contain finger-tips with almost the same radius value. After the resizing, the fingerprint enhancement implementation from [9] is applied to generate ridge orientation and frequency enhanced images.

Note that for both types, segmentation is done in a manual way which is necessary because the segmented foreground is deemed as ground truth for normalized comparison score calculation. At the recognition phase, segmentation algorithm such as the pre-processing in [186] can be applied to realize segmentation in real time. How to improve the pre-processing steps is out of the scope of this paper.

Also note that all the pre-processing steps mentioned above are only for recognition performance and normalized comparison scores calculation and they are not at all used by our proposed quality assessment approach. In this paper, all the quality scores are generated from the full size original samples with full backgrounds. We assume such a pre-processing-free quality estimation step is desirable for smartphones in terms of efficiency and power saving, considering the accurate segmentation and enhancement of foreground

Table 5.2: EER value of intra-cameras using VeriFinger 6.0 based on 424 templates from original cropped samples.

Camera type	Nokia	iPhone	Samsung
Number of reference images	63	31	23
Number of probe images	193	44	44
Number of imposter scores	11929	1239	968
EER	4.3%	2.3%	5.3%

Table 5.3: EER value of intra-scenario using VeriFinger 6.0 based on 424 templates from original cropped samples.

Computing scenario	Indoor	Darkness	Outdoor
Number of reference images	50	53	26
Number of probe images	117	97	55
Number of imposter scores	5268	4378	1008
EER	19.6%	0.01%	1.8%

shall involve high computational complexity.

5.3.1.3 Accuracy performance evaluation

To evaluate the recognition accuracy performance we generate two datasets called 'original cropped samples' and 'enhanced cropped samples' corresponding to the Type 1 and Type 2 processed data in Section 5.3.1.2.

We used two software - NIST MINDTCT and the NeuroTechnology VeriFinger 6.0 to generate the templates from original cropped samples and enhanced cropped samples. By NIST MINDTCT there are 2100 templates generated as expected. By VeriFinger 6.0 there are only 424 templates belonging to 73 fingers generated from the original cropped samples and 906 templates belonging to 97 fingers generated from the enhanced cropped samples due to the sample quality checking functionality inherent in the software. We see by sample enhancement the number of samples that can generate templates by VeriFinger 6.0 doubles (from 424 to 906). We give in Table 5.2 and Table 5.3 an example of accuracy performance using VeriFinger 6.0 over the 424 templates from the original cropped samples. Note that in both tables the sums of references and probes are less than 424 - this is because some fingers have only one sample for a specific camera or scenario and thus not selected for performance calculation.

5.3.2 Distribution of the quality scores

Figure 5.5(a) gives the distribution of the quality scores of all 2100 samples, with the minimum score 0 and the maximum score 47. As a reference to the performance examples in Table 5.2 and Table 5.3, the distribution of the quality scores of the 424 samples from which

templates can be generated by VeriFinger 6.0 is depicted in Figure 5.5(b). We can observe some correlation between the proposed quality scores and the binary quality decision made by VeriFinger 6.0 ($i.e.$, most of the samples that generate templates have the quality score larger than 4).

5.3.3 Evaluation of the proposed quality assessment approach

In this section we analyse the correlation between the normalized comparison score c_i and the quality score q_i generated by our approach for each sample x_i. The normalized comparison score is defined as follows according to the NIST definition [74].

$$c(x_i) = \frac{s_m(x_i) - E[s_n(x_{ji})]}{\sigma(s_n(x_{ji}))} \tag{5.14}$$

where $E[\cdot]$ is mathematical expectation, and $\sigma(\cdot)$ is standard deviation, $s_m(x_i)$ is the genuine comparison score generated by comparing the samples from the same finger and $s_n(x_{ji})$ are the imposter scores of sample x_i generated by comparing the samples from different fingers, $\forall j, i \neq j$.

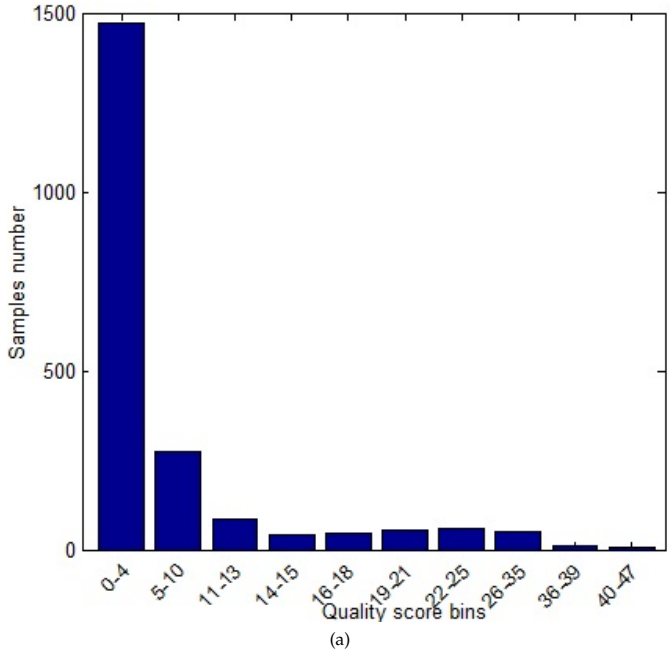
As we mentioned in the section 5.3.1.3, there are two types of datasets (original cropped samples and enhanced cropped samples) that are used to generate the normalized comparison scores. The comparison scores $s_m(x_i)$ and $s_n(x_{ji})$ are produced by both NIST BOZORTH3 and NeuroTechnology VeriFinger 6.0 comparator. In the VeriFinger 6.0 case, we assign the comparison score 0 to those samples that cannot successfully generate templates. We use the samples with maximum value of intra-finger sample comparison scores as references for enrolment in recognition performance testing. In order to include these reference samples into the correlation calculation, we assign the largest comparison score found in the testing to the corresponding reference sample during normalized comparison score calculation. At last, we obtain a group of score pairs $(c_i, q_i), i = 1, 2, ..., 2100$. To illustrate the correlation of the two types of scores, we can quantize the quality scores q_i into 10 bins and calculate the average value of the normalized comparison scores c_i in each quality score bin. An example of such correlation is shown in Figure 5.6 where the comparison scores are generated by VeriFinger 6.0 from the two datasets (original and enhanced cropped samples). The graphs indicate very high correlation between the two types of scores.

We compute the Spearman's rank correlation coefficient ρ as a quantitative method to analyze how well two variables c_i and q_i correlate. The results are given in Table 5.4 with different experimental settings for generating the normalized comparison scores (note that for generating the proposed quality scores we use the same 2100 full size original samples with full background for all settings). The results show that the proposed quality metrics are accurate to assess the samples' quality in all settings assuming the normalized comparison score for each sample as the ground truth of sample quality. The NFIQ related results in Table 5.4 provide a reference to demonstrating the effectiveness and advantage of the proposed quality assessment approach.

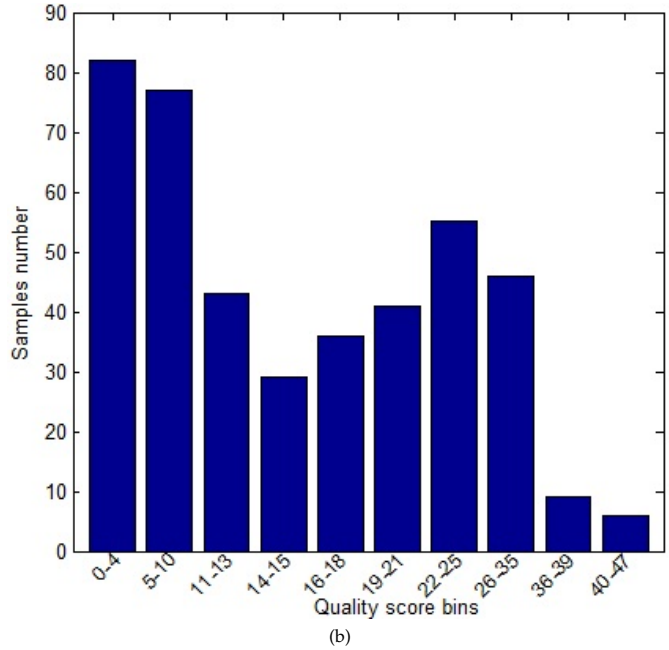
Generally speaking, the quality of samples that can successfully produce template via NeuroTechnology VeriFinger 6.0 extractor should be better than that of those samples that fail to generate the templates. If we assign a score 1 to those samples with template generated and a score 0 to those without, a pair of (t_i, q_i) can be constructed where t_i equals 0 or 1. The Spearman's rank correlation coefficients for these 2100 pairs (t_i, q_i) are shown in Table 5.5, which also indicates a high correlation.

5.3.4 The false detection case

In our experiments, there are a few samples with false detection (background blocks labelled as high-quality ones) mostly in the in-door scenario of the challenging background - the office desk surface (seen in Figure 5.1 and Figure 5.4a) - has the texture and the color

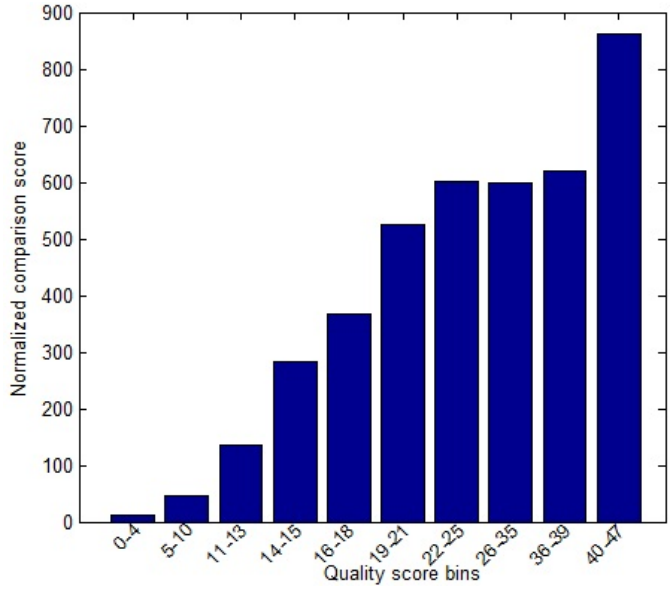


(a)

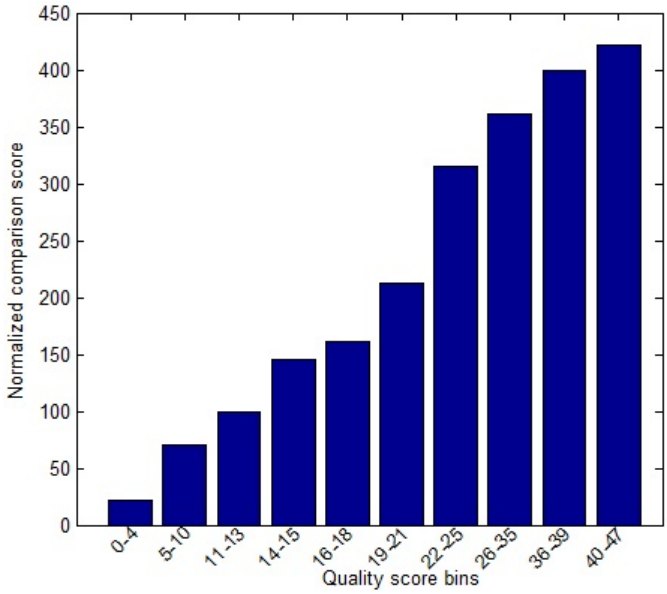


(b)

Figure 5.5: Quality scores distribution: (a) 2100 samples and (b) 424 samples that can successfully generate minutiae templates by VeriFinger 6.0 extractor.



(a)



(b)

Figure 5.6: Normalized comparison scores v.s. proposed quality scores under 10 quality score bins. (a): Normalized comparison scores generated from original cropped samples (424 samples with templates + 1676 zero comparison score samples). (b): Normalized comparison scores generated from enhanced cropped samples ((906 samples with templates + 1194 zero comparison score samples)).

Table 5.4: Spearman's rank correlation coefficients ρ under different experimental settings using the normalized comparison scores as ground truth for quality (for quality score generating full size original samples with full background are used for all settings).

Experimental settings using normalized comparison scores as ground truth	ρ
424 Original cropped samples using NeuroTechnology VeriFinger 6.0 comparator	0.59
906 Enhanced cropped samples using NeuroTechnology VeriFinger 6.0 comparator	0.42
2100 Original cropped samples (424 template-generated samples + 1676 samples with manually set normalized comparison score zero) using VeriFinger 6.0 comparator	0.47
2100 Enhanced cropped samples (906 template-generated samples + 1194 samples with manually set normalized comparison score zero) using VeriFinger 6.0 comparator	0.53
2100 Original cropped samples using BOZORTH3	0.20
2100 Enhanced cropped samples using BOZORTH3	0.49
2100 NFIQ on original full samples	-0.06
2100 NFIQ on original cropped samples	0.07

Table 5.5: Spearman's rank correlation coefficients ρ under two experimental settings using VeriFinger 6.0 sample quality checking binary decision as ground truth for quality (for quality score generating full size original samples with full background are used for all settings).

Experimental settings using VeriFinger 6.0's template generation decision as ground truth	ρ
Original cropped samples	0.45
Enhanced cropped samples	0.57

Table 5.6: Rate of false detection (background blocks identified as high-quality ones).

	False detected blocks	Total detect high quality blocks	Rate
424 samples	112	6312	1.77%
906 samples	258	8369	3.08%
2100 samples	471	10155	4.63%

looking similar to finger areas. The total number of false detected high-quality blocks is 471 blocks accounting for 4.63% of all 10155 high quality blocks on these 2100 samples, listed in Table 5.6.

Table 5.7: Spearman's rank correlation coefficient ρ between individual block features and the block quality decision.

Features	1^{st}	2^{nd}	3^{rd}	4^{th}
ρ	0.05	0.04	0.08	0.30
Features	5^{th}	6^{th}	7^{th}	8^{th}
ρ	0.30	0.26	0.27	0.27
Feature	9^{th}	10^{th}	11^{th}	12^{th}
ρ	0.14	0.27	0.26	0.35

Table 5.8: EER under different levels of quality score from the 2100 samples using NIST BOZORTH3.

Type	Group 1	Group 2	Group 3
Quality score	0-3	4-11	12-47
Samples number	1401	376	323
EER from original images	48.6%	46.6%	45.2%
EER from enhanced images	49.0%	35.0%	24.1%

5.3.5 Correlation between individual features and the block quality decision

We also evaluated the correlation between each of 12 block features and the binary quality decision for each block by computing Spearman's rank correlation coefficient. Table 5.7 shows the correlation coefficient for each feature.

5.3.6 Purpose verification of quality assessment: EER under different levels of quality scores

Recall that the purpose of sample quality assessment is to select high quality samples for recognition use. To verify if this purpose is achieved by the proposed approach or not, we calculate EERs under three levels of quality scores using NIST BOZORTH3 and NeuroTechnology VeriFinger 6.0 on different datasets. The sample with maximum quality score is always selected as the reference sample for each finger in all experiments. There are four types of combinations to compute EERs as follows:

(1). We divide the 2100 original cropped samples into 3 groups in terms of quality score: Group 1 with quality score 0-3 (more than 50% samples are with low quality), Group 2 with quality score 4-11, and Group 3 with quality score larger than 11. NIST MINDTCT and BOZORTH3 are used to extract and compare the templates. The experimental results are shown at the row "EER from original samples" in Table 5.8.

(2). Using the same settings as (1) but on 2100 cropped enhanced samples. The experimental results are shown at the row "EER from enhanced samples" in Table 5.8.

(3). We only used the 424 original cropped samples with templates generated by NeuroTechnology VeriFinger 6.0. And the three groups are $[0-9], [10-19], [20, 47]$. NeuroTechnology VeriFinger 6.0 is used to generate the comparison scores. The results are shown in Table 5.9.

(4). We only use 906 enhanced cropped samples with templates generated by NeuroTechnology VeriFinger 6.0. And the three groups are $[0-3], [4-14], [15, 47]$. NeuroTech-

Table 5.9: EER under different levels of quality score from 424 original samples using VeriFinger 6.0.

Type	Group 1	Group 2	Group 3
Quality score	0-9	10-19	20-47
Samples number	147	136	141
Number of genuine scores	90	74	90
Number of imposter scores	3757	2746	3266
EER	22.2%	12.8%	3.9%

Table 5.10: EER under different levels of quality score from 906 enhanced samples using VeriFinger 6.0.

Type	Group 1	Group 2	Group 3
Quality score	0-3	4-14	15-47
Samples number	362	324	220
Number of genuine scores	275	242	147
Number of imposter scores	19187	16027	8968
EER	35.3%	22.5%	2.7%

nology VeriFinger 6.0 is used to generate the comparison scores. The results are shown in Table 5.10.

Note that in Table 5.8 - 5.10 we try to group the samples in even distribution of sample amount. We observe that EERs are significantly reduced along the increase of sample quality except the case NIST BOZORTH3 operating on original samples which is however not very likely to be adopted for practical use. The experimental results demonstrate the effectiveness of our proposed quality assessment approach in predicting the quality of fingerprint samples generated by smartphone cameras. Note that for Group 1 and 2, the EERs in Table 5.10 is higher than those in Table 5.9, which could be due to the fact that sample enhancement increases the number of samples that can generate templates but decreases the average sample quality in the meanwhile.

5.4 Conclusion and future work

This paper proposes an effective fingerprint sample quality assessment approach for the samples captured by the smartphone cameras using a set of block based quality features. Our approach is pre-processing-free (without needing segmentation and enhancement) and block-based (memory saving and parallelizable in computation) thus potentially efficient in computation on mobile devices. The correlation between the quality score generated by the proposed approach and the normalized comparison score (as ground truth of quality) of each sample has been evaluated by computing Spearman's rank correlation coefficient of the two scores. Experimental results demonstrate that the proposed quality assessment approach is capable of identifying the high-quality fingerprint area from both those low-quality ones and those complicated background ones and thus capable of pre-

dicting the sample quality. Our future work will focus on reducing the false detection rate and improve the block size normalization across different cameras.

Chapter 6

Qualifying Fingerprint Samples Captured by Smartphone Cameras in Real-Life Scenarios

Abstract

While biometrics has been extensively adopted by industry and governments for identification and forensics purposes relying on dedicated biometric sensors and systems, the consumer market driven by innovations in consumer electronics (smartphones, tablets, etc.) is believed to be the next sector that biometric technologies can find wider applications. Compared to dedicated biometric sensors, the sensors embedded in such general-purposed devices may suffer from sample quality instability, which has significant impact on biometric performance. The concern on sample quality may jeopardize the market confidence in consumer devices for biometric applications. In this paper, we propose an approach to assessing the quality of fingerprint samples captured by smartphone cameras under real-life uncontrolled environments. Our approach consists of a sample processing pipeline during which a sample is divided into blocks and a set of local quality features are extracted from each block, including 3 pixel-based features, 4 autocorrelation based features, and 5 frequency features from the autocorrelation result. Afterwards, a global sample quality score is calculated by fusing all image blocks' qualification status. Thanks to the extracted features' capability in discriminating high-quality foreground (fingerprint area) blocks from low-quality foreground ones and background ones, the proposed approach does not require foreground segmentation in advance and thus we call it a one-stop-shop approach. Experiments compare the proposed approach with NFIQ and the proposed pipeline using standardized quality features, and demonstrate our approach's better performance in qualifying smartphone-camera fingerprint samples.

6.1 Introduction

Biometrics [106] has been widely adopted for identification purpose (to verify or to search for the identity of an individual) and forensics purpose (to collect and compare biometric traits as legal evidence). As the most widely-adopted ($e.g.$, by ICAO [94] for ePassport) biometric modality for governmental and industrial applications, fingerprint recognition has been standardized by ISO [100] and nowadays deployed in many identity management solutions. While fingerprint has been extensively used and enabled by dedicated biometric sensors and systems, the consumer market driven by innovations in consumer-oriented mobile devices (smartphone, tablet, smart-watch, Google Glass, $etc.$) in recent years is opening an even wider market for fingerprint technologies enabled by such general purposed mobile devices. These general-purposed mobile devices, when adopted for different biometric applications, $e.g.$, device access control [186] [17], remote identity authentication [20] [8], or simply a biometric reader, may have advantages in portability, costs, state-of-the-art sensor integration, multi-functional integration, interface compatibility, convenience to use, and even privacy for personal use both technically and psychologically, since the device as a biometric reader is always under the owner's control. These integrated general purposed sensors (camera, microphone, accelerometer, $etc.$) show potential to be exploited as biometric sensors. However, the sample quality, which has significant impact on biometric performance [187], rendered by such embedded or plug-in

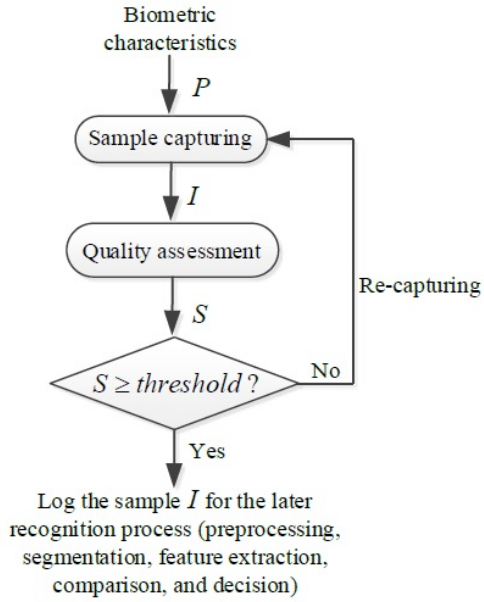
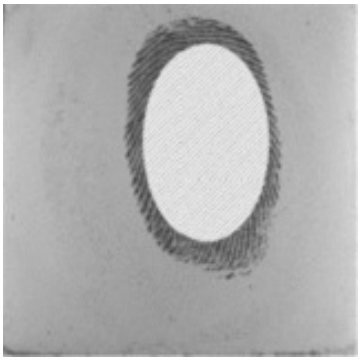


Figure 6.1: A general process of biometric sample quality control (P: probe; I: captured sample; S: quality score)

sensors ($e.g.$, the fingerprint samples captured by a smartphone built-in camera in an uncontrolled environment) is usually less stable compared to the case using dedicated biometric sensors. This concern on sample quality may jeopardize the market confidence in such general-purposed mobile devices, especially those popular consumer electronics, for biometric use.

To ensure that a biometric system is operated with high accuracy performance ($i.e.$, low error rates), sample must be carefully controlled in quality during the capturing phase. Figure 6.1 gives a generic workflow for sample quality control in a biometric system, where the quality assessment function is the key part to the whole process. Sample quality control ensures that a captured sample has enough quality for the following recognition process in the sense that both the FTA (fail-to-acquire) rate [101] for features generation and the FTE (fail-to-enroll) rate [101] for reference generation can be minimized as well as biometric recognition accuracy being maximized. Since the quality control process takes iterations before logging a qualified sample, a computationally-efficient quality assessment approach is always desired, especially for mobile devices.

For fingerprint samples, various quality assessment approaches have been studied [187] [40] [210] [66] [85] [211] and standardized [104] but all these approaches are limited in scope to samples generated from dedicated fingerprint sensors, $i.e.$, touch-based sensors or environment-controlled touchless sensors, which generates a fairly clean background and a high-contrast foreground ($i.e.$, ridge patterns) such as the example in Figure 6.2a. However, samples captured by a general purposed smartphone camera look so different, such as the example in Figure 6.2b, that existing sample quality assessment approaches may not work in this case. This drove us to investigate the feasibility of the existing approaches on smartphone-camera samples and, or to propose new approaches to address this new challenge brought by such smartphone cameras.



(a) Captured by optical sensor L-1 DFR2100

(b) Captured by camera embedded in Samsung Galaxy S

Figure 6.2: Two samples from the same finger (the fingerprint in (a) is cropped for privacy protection purpose in this example)

To test the quality of smartphone-camera fingerprint samples, we created a database with samples collected from 3 widely-used smartphones under various real-life scenarios. Both biometric performance testing and sample quality assessment were done on this database. A new one-stop-shop approach is proposed and compared to some traditional approaches in sample quality assessment. As a pilot study on this topic, our work described in this paper has the following merits:

(1) A real-life scenario smartphone-camera fingerprint database was established containing samples in large quality variance, which could be, as far as we know, the first database of this type in the biometric research society;

(2) A one-stop-shop pipeline was proposed for sample quality assessment without needing computationally-intensive foreground (fingerprint area) segmentation for quality-challenged (complicate background or ill-illuminated) samples;

(3) Differential-autocorrelative-integration (DAI), an efficient block ridge pattern descriptor, was proposed to extract quality features with high discriminability;

(4) Metrics were suggested for evaluating the performance of sample quality assessment methods suitable for smartphone-camera fingerprint samples.

Section 6.2 gives background on the fingerprint sample quality assessment, unique characteristics of smartphone-camera samples, and the challenges to existing fingerprint quality assessment approaches. Section 6.3 proposes our one-stop-shop pipeline, which

is designed to cope with the said challenges, tailors some quality metrics used for samples generated from dedicated sensors, and proposes new quality metrics to better suit the smartphone-camera fingerprint samples. Section 6.4 introduces the real-life smartphone camera fingerprint database this paper established and the experimental settings for performance testing of the proposed quality assessment approach. Section 6.5 presents testing results with comparison to some typical quality features designed for traditional fingerprint samples. Section 6.6 concludes this paper.

6.2 Background information

6.2.1 Fingerprint sample quality: concept and methodology

Biometric sample quality has significant impact on a biometric system's recognition performance [187]. This is because the performance evaluation process involves cross comparisons among subjects' templates, as both probes and references, generated from biometric samples. Low-quality samples, even few in amount, can play a major role [97] in contribution to error rates, $e.g.$, the false match rate (FMR) and the false non-match rate ($FNMR$). The purpose of sample quality control, $i.e.$, trying to discern low-quality probe samples, is indispensable for a biometric system expected to operate in high accuracy.

To define the concept of biometric sample quality in a standard way, the international standard ISO/IEC 297941:2009 [101] considers it from three different perspectives:

(1) Character, based on the inherent features of the source, $e.g.$ poor character due to scars in a fingerprint;

(2) Fidelity, reflecting the degree of a sample's similarity to its source;

(3) Utility, indicating how (positively or negatively) a sample, by its quality status, contributes to the accuracy performance of a biometric system. Obviously the utility has dependency on both the character and the fidelity of a sample.

For a biometric recognition system, the utility of a sample is of most interest because it is directly contributing to the recognition accuracy. To describe fingerprint sample quality, normalized comparison score (Equation 6.1), expressed by NIST was defined in [187], which we believe can be generalized to all biometric modalities to characterize the utility of a biometric sample in the recognition accuracy sense expressed by error rates. Suppose x_i is a sample to be assessed in quality, its normalized comparison score $c(x_i)$ is

$$c(x_i) = \frac{s_m(x_i) - E[s_n(s_{ji})]}{\sigma(s_n(s_{ji}))} \tag{6.1}$$

where $E[\bullet]$ is a mathematical expectation, $\sigma[\bullet]$ is a standard deviation, $s_m(x_i)$ is a genuine comparison score generated by comparing the probe x_i to its reference originated from the same finger, and $s_n(x_{ji})$ are the imposter scores of sample x_i generated by comparing the probe x_i to the references originated from non-mated fingerprint samples, $\forall j, i \neq j$. Characterizing the distinguishability of the genuine comparison score from all imposter scores obtained from the studied probe sample, the quality metric goes coherently with the recognition performance in the sense of error rates. However, calculating a normalized comparison score implies comparisons between a probe and all references in the database. This process is unrealistic to launch as an online operation due to a high computational complexity, let alone when a sample is used for enrolment there does not exists any reference at all. These facts negate the feasibility using the normalized comparison score directly to assess a sample's quality. However, the normalized comparison score can be reasonably deemed as ground truth of the quality of a sample in the sense of utility, and thus provides a reference to correlation calculations ($e.g.$, Spearman's rank correlation [187]) with

any quality metrics that can be operated in an online mode without requiring information provision from biometric references in the database. Such a recognition performance predictive approach suitable for online operation is what we call sample quality assessment approach in this paper.

In the case of fingerprint samples, poor-quality samples generally produce spurious minutia or lose genuine minutiae. For instance, a sample with partial fingerprint area can have only a small portion of minutiae recorded and even lose singular points (the core point and the delta point), which are important global reference points for sample alignment. For dedicated fingerprint sensors, in addition to partial fingerprint recording, low quality can be attributed to varying temperature / humidity conditions of the finger skin, low physical pressure, too less presentation time, incorrect finger positioning angles, $etc.$ Such low-quality samples should be rejected after the sample quality assessment process, and a re-capturing action under improved environmental conditions should be initiated.

Clarity of ridges and valleys is a commonly recognized criterion to measure the quality of a fingerprint sample [66]. Several factors can influence the clarity of ridges and valleys, such as the acquisition sensor itself, the capturing environment, skin disease, skin humidity and specifically for touch-based fingerprint sensors also the pressure [210]. For example, a wet finger placed on an optical fingerprint sensor or high pressure exerted during a capture process will generate a sample image with connected dark area inside which ridges and valleys are difficult to discriminate. In some scenarios, fingerprint samples can be captured in a controlled environment compliant to standards [100] [101] [104] [97] to maximize the sample quality. For example, automatic fingerprint identification systems (AFIS) are widely deployed for border control and other national and international identity management purposes, such as the Visa Information System (VIS) in Europe, US-VISIT / IDENT system in US, and the Aadhaar project in India. In such scenarios, professional sensors distinguished from massive performance tests are usually chosen to acquire fingerprint samples under an ideal environment (fair and stable illumination, comfortable indoor climate, assistance and guidance from attendants, $etc.$).

There are numerous fingerprint sample quality assessment methods [187] [40] [210] [66] [85] [211] [181] [159] [157] have been proposed using various quality features for sample quality assessment. Some of the quality features have been incorporated into the ISO/IEC:29794-4 technical report [104]. As a holistic approach employing multiple features (including minutiae) and artificial neural network for fingerprint sample quality assessment, the NFIQ function [187] [188] was released by NIST in 2004 and widely adopted since. The NFIQ function can label a sample in 5 quality levels among which level 1 indicates the best quality. Since the year 2011, the NFIQ 2.0 project [24], as an improved version of NFIQ, has been initialized and is currently under progress. However, all these methods mentioned above focused on samples captured by traditional sensors and did not consider the characteristics of smartphone-camera fingerprint samples.

6.2.2 What makes smartphone camera based fingerprint capturing different

In consumer markets, Apple has released iPhone 5s with a fingerprint sensor built into the phone's home button [17]. Although such integrated dedicated fingerprint sensors can better ensure the sample quality, they incur additional cost and space occupation in a smartphone. The cameras embedded in smartphones, however, are promising to provide us an alternative option to sense fingerprint at almost no hardware cost. Such touchless sensors can in theory generate samples in higher utility compared to touch-based sensors because they can capture a larger finger area, which translates to more distinguishable features ($e.g.$, more minutiae) [123]. Nowadays a typical smartphone is equipped with a high-resolution 5$\sim$20 mega-pixel camera, which enables them to capture fingerprint samples equivalent to very high DPI (dots per inch). Previous research [186] [73] [160] have shown this possibility.

Compared to the case fingerprints are captured by a general-purposed camera in an ideal laboratory environment [186] [73] [160], the samples captured in real life scenarios de-

(a) Failed to focus

(b) Camera far away from the finger

Figure 6.3: Two samples captured by a smartphone camera.

fined in our previous work [218] [133] [132] [217] show quite unstable quality due to camera motion, de-focusing, unfavored illumination, incorrect finger positioning, and complicated backgrounds. Figure 6.2 illustrates two fingerprint samples captured from the same finger: Figure 6.2a is a sample captured by a touch-based optical sensor L-1 DFR2100 and Figure 6.2b is a sample captured from the same finger by the camera embedded in Samsung Galaxy S. Figure 6.3 shows two smartphone camera fingerprint samples which are not qualified for the recognition purpose: (a) fails to focus on the finger area; and in (b) the camera was placed too far resulting in low resolution in the fingerprint area. In both samples, the ridges and valleys are not able to record and thus impossible for feature extraction required for recognition. Such samples should be precisely detected by a sample quality assessment function and then discarded.

Observed from Figure 6.2, we can see the difference between the two types of samples: samples captured from traditional fingerprint sensors (including those professional touchless fingerprint sensors such as TST BiRD [29]) exhibit relatively stable quality characterized by clean and homogeneous background, evenly distributed illumination, fair focusing and positioning, but limited fingerprint area; while samples captured from smartphone cameras exhibit unstable quality characterized by unpredictable background, sometimes biased illumination and de-focusing, but in general larger fingerprint area.

As mentioned in Section 6.2.1, there exist many quality assessment approaches for samples captured like the type in Figure 6.2a. But it is doubtful such methods can be directly applied to those smartphone camera fingerprint samples like the type in Figure 6.2b, assuming that complicate foreground segmentation and illumination adjustment required by such smartphone-camera samples have never been incorporated into the design of traditional quality assessment approaches. To verify this assumption, we tested the NFIQ

(a) Minutiae detected by NIST function MINDTCT

(b) Minutiae detected by VeriFinger 6.0 Extractor

Figure 6.4: Samples with high quality (level 1) labelled by NFIQ: blue cross marking the detected minutiae

function on some smartphone camera samples and only found that a significant percentage of samples labelled with high quality (level 1) are in fact low-quality ones. As minutiae count and quality information are used in NFIQ, these challenging samples might fool the NFIQ function with too many spurious minutiae detected from both the background and foreground. Figure 6.4 illustrates two examples in this case, where 6.4a and 6.4b show minutiae detection results by the NIST function MINDTCT and the widely-used commercial minutiae detector Neurotechnology VeriFinger 6.0 Extractor [22], indicating both these two popular minutiae detectors were not good at coping with such smartphone-camera fingerprint samples. From these observations, we can reasonably infer that simple preprocessing mechanisms (*e.g.*, the quality map used in NFIQ to identify foreground blocks) are not capable towards such samples.

6.2.3 One-stop-shop quality assessment

The spurious minutiae detected in the examples in Figure 6.4 is due to lack (or incapability) of accurate foreground segmentation. An accurate segmentation algorithm usually requires intensive computation. Such intensive resource consumption could be unsuitable for the iterative process of quality control shown in Figure 6.1, especially for mobile devices. In addition, such an accurate segmentation algorithm itself is not easy to achieve dealing with unpredictable backgrounds. Furthermore, unlike traditional fingerprints, which are evenly illuminated under a controlled environment, a lot of samples captured by smartphone cameras are biasedly illuminated causing shade areas within a finger area, such as

the typical case in Figure 6.3a. Such shades are easy to detect as foreground but actually provides no useful information for recognition. Considering all these facts, we envision a segmentation-free approach that discriminates high-quality fingerprint patterns from those low-quality ones and the background ones in one operation. We propose in this paper such a one-stop-shop quality assessment approach for smartphone-camera fingerprint samples in real-life scenarios. Details are given in Section 6.3.

6.3 Proposed quality metrics

6.3.1 Pipeline of the proposed approach

The proposed one-stop-shop approach, as shown in Figure 6.5, divides a sample image I into N non-overlapping blocks $B_i(i = 1, 2, \dots, N)$, and checks each image block's quality status - qualified or non-qualified (including the low-quality case and the background case) - before fusing all blocks' quality decisions d_i to produce the final quality score S_I for the sample. From each block, a 12-dimensional quality feature vector $B_i(f_1, f_2, \dots, f_{12})$ is formed. During enrolment, such quality feature vectors together with their ground-truth quality labels are used to train a SVM classifier; and during quality assessment, a probe sample is labelled by the trained SVM classifier as "qualified" or "non-qualified". The ground-truth blocks are selected and labelled manually according to their sources, $i.e.$, samples with low and high normalized comparison scores. Summation is selected as the decision fusion rule. In this way, $\#QB$, the number of "qualified blocks" in a sample can be output as the quality score S_I after being normalized by the number of blocks in the sample:

$$S_I = \frac{\#QB}{N} \tag{6.2}$$

During the sample capturing process, the subject can be required to place his/her finger in an appropriate finger-to-camera distance. A simple rule, used in photography for sharpness evaluation [180], is to evenly divide the whole image into 3×3 rectangular regions and require the foreground (fingerprint area) to approximately cover this central region. In addition, to offset the variability in digital resolution of different camera settings, we define the block size in a way that in average around 4∼10 ridges can be identified in one block. Heuristically, the block size (in pixel amount) can be determined against the size of the central region.

6.3.2 Block orientation alignment

Before feature extraction, all blocks need be aligned in orientation, assuming a high-quality block contains homogeneously-oriented ridges. If the block size is large ($e.g.$, ridge count > 10), this assumption may not apply to those extremely-high curvature ridge areas, $e.g.$, the core or delta points. Fortunately, such areas normally cover only a small percentage of an entire fingerprint. Besides, the typical block size of 4∼10 ridges limits the inhomogeneity in orientation. We tested a subset of blocks from our test database and found only $<$ 4% blocks have challenge in orientation alignment, judged by human eyes, among which except those inherently high-curvature blocks, most inaccurate orientation alignment have only distortions of 5∼10 degrees.

Suppose an image block B_i is sized $R \times C$ in pixel (R and $C = 2^k, k = 1, 2, 3, \dots$). After low-pass Gaussian filtering to suppress random noises, the Principal Component Analysis (PCA) based gradient orientation estimation method [46] is used to find a block's principal orientation. That is, inside each block neighboring pixels' differences d_v and d_h (in vertical and horizontal directions respectively) are obtained to form a gradient vector with orientation $tan^{-1}(d_v/d_h)$. Then the block principal orientation θ_i is calculated by PCA to identify the principal one among all orientations of the $(R-1) \times (C-1)$ calculated

Figure 6.5: The proposed one-stop-shop sample quality assessment approach

gradient vectors. By clock-wisely rotating the $\sqrt{2}(R-1) \times \sqrt{2}(C-1)$ size area concentric to B_i by angle θ_i, we can crop a block $B_i^{'}$ sized $R \times C$ concentric to B_i. In this way we assume $B_i^{'}$ has the maximum gradient in the horizontal direction. Note that this block principal orientation derived from gradients is perpendicular to the principle orientation of the block ridges.

6.3.3 An efficient ridge pattern descriptor: Differential-Autocorrelative-Integration (DAI)

After block orientation alignment, quality features can be extracted from the ridge pattern. Assuming that a fingerprint ridge block exhibits a periodic characteristic that can be approximated by sinusoidal-wave-like ridge and valley repetition, we expect to represent this periodic characteristic by spatial frequency (called principal frequency in Section 6.3.4) while suppressing noises in other frequency bands. Driven by this intuition, we propose the following procedure to describe a block ridge pattern.

Step 1 Differential operation along rows. With low-pass filtering done before orientation alignment, we consider using differential operation, effecting as high-pass filtering to capture the ridge-valley variations, on neighboring pixels along each row in the orientation aligned block $B_i^{'}$. It is an operation same as we performed on B_i to calculate d_h during orientation alignment;

Step 2 Autocorrelation along rows. Autocorrelation [2], as a commonly-used signal processing method to detect periodic patterns polluted by noises, is used on the $(C-1)$ pixel residues in each row. After the autocorrelation calculation, we keep the former $(C-1)$ dimensions and remove the latter $(C-2)$ redundant dimensions of the autocorrelation result vector;

Step 3 Integration along columns. Sum up all R autocorrelation results to obtain a $(C-1)$ dimensional vector. Here summation is used to increase the robustness of the descriptor by suppressing, if any, local minor inhomogeneity in ridge pattern ($e.g.$, caused by ridge endings and bifurcations).

We name the above three-step operation as Differential-Autocorrelative-Integration (DAI), as a new ridge pattern descriptor for quality feature extraction. Both spatial domain and Fast Fourier Transformation (FFT) frequency domain features can be extracted from this DAI descriptor, as we show in Section 6.3.4.

As we can observe from Figure 6.6 -6.8, for a high-quality block, the absolute amplitudes of local peaks and valleys take on a stable increase in Figure 6.6. However this cannot be observed for a low-quality block and a background block as shown in Figure 6.7 and Figure 6.8. Moreover, in the Fourier transform domain we can observe the highest peak has distinctly higher prominence in a high-quality block as seen in Figure 6.6.

6.3.4 Proposed quality features

We summarize the description of Section 6.3.1 - Section 6.3.3 and illustrate the proposed sample quality assessment pipeline in Figure 6.9.

We propose 12 quality features $f_i (i = 1, 2, \ldots, 12)$ of three types to assess an image block's quality: (a) 3 pixel based features; (b) 4 DAI descriptor based features; c) 5 spectrum features of the DAI descriptor. A quality feature vector can be formed by these 12 features for an image block. The detail of each feature is described as follows. Note that these 12 features are not necessarily in practice the best ones for smartphone camera fingerprint sample quality assessment but included in order to characterize the different dimensions of fingerprint patterns.

i). Pixel based features

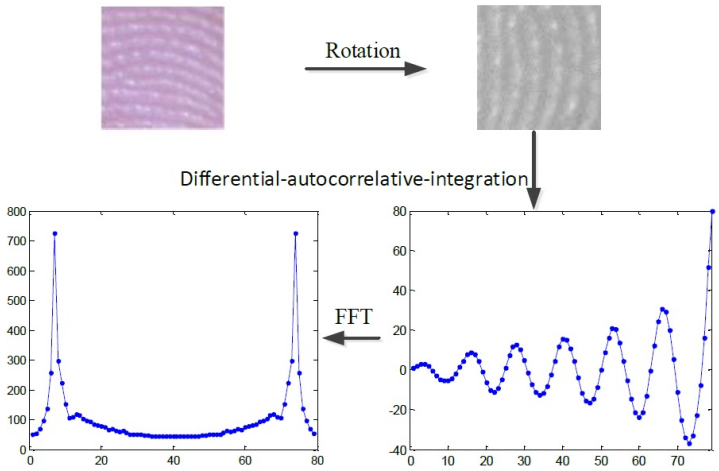


Figure 6.6: Processing a high-quality block

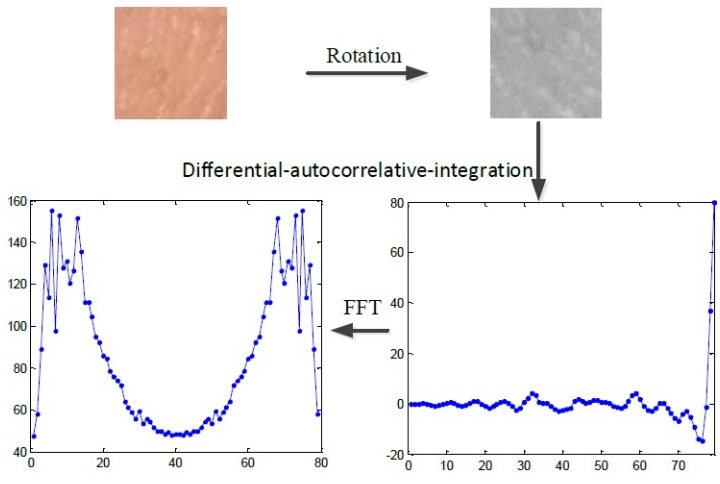


Figure 6.7: Processing a low-quality block

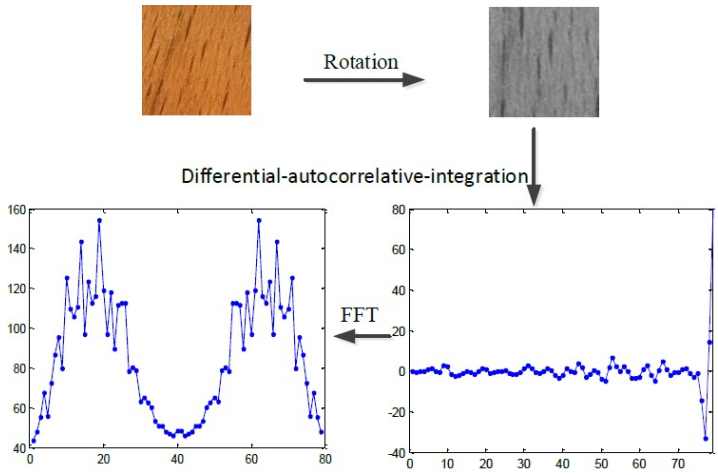


Figure 6.8: Processing a low-quality block

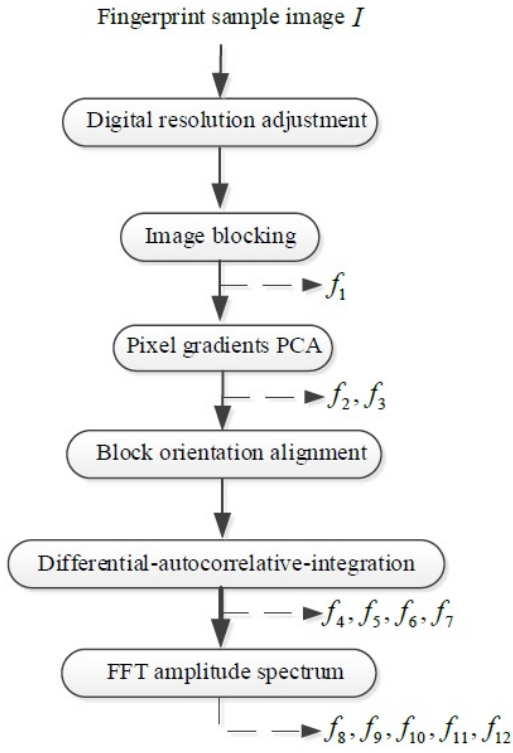


Figure 6.9: Features extraction from different steps of the proposed pipeline

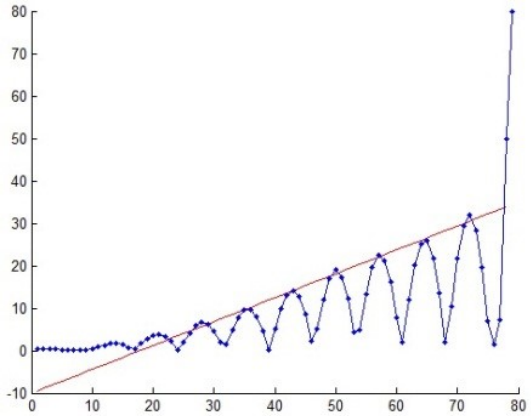


Figure 6.10: $|\boldsymbol{ACR}_i|$ curve ($i.e.$, the absolute amplitudes of first half of $|\boldsymbol{ACR}_i|$'s autocorrelation), C = 80 (the straight line is the linear best fit of the M peak points)

(1) f_1 : Exposure, calculated from the average pixel value of B_i

$$f_1 = \frac{1}{R \times C} \sum_{r=1}^{R} \sum_{c=1}^{C} B_i(r, c) \tag{6.3}$$

where $B_i(r, c)$ is the pixel at the $r - th$ row and $c - th$ column inside the block B_i. Both too-bright and too-dark fingerprint areas are unfavored for feature extraction.

(2) f_2 : Significance of the principal orientation

$$f_2 = \lambda_1 \tag{6.4}$$

where λ_1 is the first eigenvalue of the covariance matrix of all gradient vectors obtained from the PCA calculation, indicating the significance of a block's principal orientation calculated in Section 6.3.2.

(3) f_3 Certainty of the principal orientation. We use a modified definition of ocl (orientation certainty level) in [104] as follows:

$$f_3 = \begin{cases} 1 - \frac{\lambda_2}{\lambda_1} & \text{if } \lambda_1 \neq 0 \\ 0 & \text{if } \lambda_1 = 0 \end{cases} \tag{6.5}$$

where λ_2 is the second eigenvalue of the covariance matrix of all gradient vectors.

ii). DAI descriptor based features

As described in Section 6.3.3, we calculate autocorrelation on the horizontally-differential vectors $\boldsymbol{d}_i(r), 1 \leq r \leq R - 1$, and obtain the DAI descriptor as

$$\mathbf{acr}_i = \sum\nolimits_{r=1}^{R-1} autocorr(\boldsymbol{d}_i(r)) \tag{6.6}$$

where $\boldsymbol{d}_i(r) = (b_i(r, 2) - b_i(r, 1), b_i(r, 3) - b_i(r, 2), ..., b_i(r, C) - b_i(r, C - 1))$. And $autocorr$ $(\boldsymbol{d}_i(r))(j) = \sum_{c=1}^{C-1} \boldsymbol{d}_i(r, c)\boldsymbol{d}_i(r, c + j)$, $(0 \leq j \leq C - 2)$, with all $C - 1$ amplitudes divided by the highest amplitude of $autocorr(\boldsymbol{d}_i(r))$. Before the subsequent feature extraction steps, a low-pass filtering is applied by setting the upper half of $\boldsymbol{DAI}_i$'s DCT-transform frequency coefficients to zero. Thus a smoothening of the $\boldsymbol{DAI}_i$ vector is reached, denoted as $\boldsymbol{ACR}_i$.

(4) $|\boldsymbol{ACR}_i|$'s peak activity rate.

From the observation in the experiments, we find the local peaks (excluding the maximum peak *i.e.* the $(C-1)th-$ dimension of $|\boldsymbol{ACR}_i|$) of the $|\boldsymbol{ACR}_i|$ curve (i.e., the absolute amplitude curve of $|\boldsymbol{ACR}_i|$) have a stable increasing rate in those ground-truth good quality blocks. We use a 1st-order polynomial (i.e. a straight line) to fit the M detected peaks in their x-coordinates $x_{p1}, x_{p2}, \ldots, x_{pM}$ in the $|\boldsymbol{ACR}_i|$ curve (shown in Figure 6.10) and obtain a fitted straight line with slope value S. M amplitudes $A(x_{pn})(n = 1, 2, \ldots, M)$ on the fitted line can be found. Then the $|\boldsymbol{ACR}_i|$'s peak activity rate is defined as:

$$f_4 = \frac{1}{M}\sum_{n=1}^{M} A(P_n(x))(A(P_n(x)) > 0). \tag{6.7}$$

As we can observe in Figure 6.6 and Figure 6.7, the peaks in the autocorrelation result are closer to the x-axis for low-quality blocks and background block than the high-quality block case. Thus the value of f_4 is expected to be significantly higher for a high-quality block.

(5) $|\boldsymbol{ACR}_i|$'s peak pick-up rate.

We denote it using the slope of the straight line in Figure 6.10. A denotes the amplitudes (y-coordinates) in x-coordinates x_{pn} on the line:

$$f_5 = S = \frac{A(x_{p(n+1)}) - A(x_{pn})}{x_{p(n+1)} - x_{pn}} \tag{6.8}$$

where $n \in 1, 2, \ldots, M$. This feature may take on a high value if the ridges in the block are not uniformly illuminated or influenced by external noises like dirt spots.

(6) f_6 : $|\boldsymbol{ACR}_i|$'s peak variance rate.

We use this rate to represent the degree that the actual M peak amplitudes $y_{p1}, y_{p2}, , y_{pM}$ deviate from the fitted line.

$$f_6 = \frac{\frac{1}{M}\sum_{n=1}^{M} |P_{pn} - A(x_{pn})|}{max(A(x_{pn})) - min(A(x_{pn})} \tag{6.9}$$

where $n \in 1, 2, \ldots, M$. This feature may take on a high value if the ridges in the block are not uniformly illuminated or influenced by external noises like dirt spots.

(7) f_7 : $|\boldsymbol{ACR}_i|$'s peak drop rate.

We use this rate to represent the degree of the amplitude drop $AD_n = y_{p(n+1)} - y_{pn} (n = 1, 2, \ldots, M-1)$ of one peak compared to its neighboring peak on the left side. From the observation in the experiments, large drops in amplitude seldom happen to high quality blocks.

$$f_7 = 1 - \frac{\sum_{j=1}^{M-1} |AD_j(AD_j < 0)|/(M-1)}{max(A(P_n(x))) - min(A(P_n(x)))} \tag{6.10}$$

iii). Spectrum feature of the DAI descriptor

This type of features is derived from the FFT amplitude spectrum characteristics of $\boldsymbol{ACR}_i$, which we denote as $|f\boldsymbol{ACR}_i|$, characterizing ridges' spatial frequency properties.

(8) f_8 : Principal frequency's amplitude:

$$f_8 = max(|f\boldsymbol{ACR}_i|) \tag{6.11}$$

Due to the periodicity of ridge structures, a high-quality block may have a principal frequency with high amplitude in its FFT amplitude spectrum.

(9) f_9 : Principal frequency, *i.e.*, f_8's frequency index in the amplitude spectrum.

As observed in Figure 6.6 to Figure 6.8, most of the energy concentrates on the principal amplitude and its neighbors in the high-quality blocks, forming a sharper peak. The features f_{10}, f_{11}, f_{12} are thus extracted to describe the degree of energy concentration. The feature f_{10} depicts the energy distribution among a quarter of $|f\boldsymbol{ACR}_i|$'s components with

highest amplitudes. The feature f_{12} and f_{11} depict the energy distribution among a close adjacent and a second-close adjacent frequency ranges centering the principal one.

(10) f_{10} : Principal frequency's dominance rate:

$$f_{10} = 1 - \frac{4 \times \sum_{n=2}^{C/4} Q_i(n)}{(C-4) \times Q_i(1)} \tag{6.12}$$

where we denote $Q_i(1), Q_i(2), , Q_i(\lfloor((C-1))/4\rfloor)$ as the quarter of $f\boldsymbol{ACR}_i$'s highest amplitudes, i.e., the amplitudes of former $\lfloor((C-1))/4\rfloor$ frequencies of $f\boldsymbol{ACR}_i$ after being sorted in a descending order by amplitude. Obviously, $Q_i(1) = f_8$. A high value of feature f_{10} indicates good quality for a block, in the sense that the amplitude spectrum has a dominant principal frequency compared to its peer amplitude peaks, if any.

(11) f_{11} : Principal frequency's prominence rate close adjacent frequency range:

$$f_{11} = 1 - \frac{(\sum_{n=-H}^{H} f\boldsymbol{ACR}_i(L+n)) - f\boldsymbol{ACR}_i(L)}{2H \times f\boldsymbol{ACR}_i(L)} \tag{6.13}$$

(12) f_{12} : Principal frequency's prominence rate (second-close adjacent frequency range):

$$f_{12} = \frac{\sum_{n=-X}^{X} f\boldsymbol{ACR}_i(L+n) - \sum_{n=-H}^{H} f\boldsymbol{ACR}_i(L+n)}{2(X-H) \times f\boldsymbol{ACR}_i(L)} \tag{6.14}$$

where L is the principal frequency's index in the amplitude spectrum vector $F = |f\boldsymbol{ACR}_i|$, $0 < H < X$, and $L - X > 0$, otherwise $f_{12} = 1$. A high value of feature f_{11} and f_{12} indicate good quality for a block, in the sense that the principal frequency's amplitude takes on a prominent peak outstanding from neighboring frequencies.

6.3.5 Feature dynamic range normalization

The features $f_i (i = 1, 2, \ldots, 12)$ are z-score normalized prior to being used by the SVM:

$$f_i^{'} = \frac{f_i - E(f_i)}{\sigma(f_i)} \tag{6.15}$$

where $E(\bullet)$ and $\sigma(\bullet)$ are expectation and standard deviation respectively.

6.4 Experimental settings

6.4.1 Experiments design and dataset collection

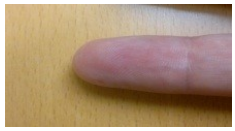
For evaluating a quality assessment approach, we assume that higher quality samples result in lower error rates in recognition performance testing. We can thus use the normalized comparison scores [187] as the ground truth to calibrate samples' quality by correlating the quality scores calculated from the proposed one-stop-shop approach with their normalized comparison scores. Three evaluation metrics - Spearman's rank correlation coefficient, Error Reject Curves (ERC) [85], and false detection rate - were adopted to evaluate the performance of the proposed quality assessment approach, with results given in Section 6.5.

Three smartphones: iPhone 4, Samsung Galaxy S, and Nokia N8 were used to capture fingerprint samples from 100 different finger instances from 25 subjects. From each subject, four fingers - left index, left middle, right index, and right middle - were required to generate 3 samples from each. Table 6.1 specifies the three smartphone cameras. We considered three real-life scenarios: (1) the indoor scenario with ideal illumination but a challenging background (desk surface) (Figure 6.11a); (2) the dark scenario with only illumination from the smartphone's automatic flash (Figure 6.11b); and (3) the outdoor scenario with a complicate background (Figure 6.11c). All three smartphones were used in the indoor and the

Table 6.1: Specification of the three smartphones' cameras.

Mobile phone	Nokia N8	iPhone 4	Samsung Galaxy S
Mega pixel	12.0	5.0	5.0
Resolution	1536×1936	2592×1936	1600×960
Auto-focus	Yes	Yes	Yes
Image format	JPEG	JPEG	JPEG
ISO control	automatic	automatic	automatic
Flash source	Xenon	LED	no flash
Flash setting	automatic	automatic	no flash
Aperture	f/2.8	f/2.8	f/2.6
Sensor size	1/1.83″	1/3.2″	1/3.6″

outdoor scenarios but only Nokia N8 was used in the dark scenario (the other two failed to capture samples in darkness). In total there are 2100 fingerprint samples captured.



(a) in-door

(b) dark(automatic flashing)

(c) outdoor

Figure 6.11: Fingerprint samples captured in three scenarios.

6.4.2 Pre-processing for ground-truth quality calculation

The proposed approach does not need to segment the foreground (finger area) for quality assessment. However, to obtain the normalized comparison scores as sample quality's ground truth, pre-processing is required to the captured samples to generate fingerprint templates. Such pre-processing steps could include (1) segmentation of the fingerprint area; (2) sample resizing (to offset the distance variance of fingers from the camera); and (3) fingerprint area enhancement.

Pre-processing step 1: manual segmentation is performed to crop the foreground as a ground-truth fingerprint area. In practical fingerprint recognition systems, a segmentation algorithm such as the pre-processing in [186] can be applied to for segmentation in real time. How to improve accuracy and efficiency of the pre-processing is the key to recognition performance but out of scope of this paper.

Pre-processing step 2: sample resizing is implemented by the following sub-steps: (1) fit the fingertip shape as a half-circle, and detect this circle using the Hough transform over the boundary of the foreground; (2) align the radius of the detected fingertip half-circle to a constant value; and (3) resize the whole cropped sample according to the new aligned radius. In this way, all the resized samples contain fingertips with almost the same radius. After the resizing, the fingerprint enhancement implementation from [9] is applied to enhance the ridge orientation and frequency.

Pre-processing step 3: Histogram equalization will be performed to enhance the sample outputted from the pre-processing step 2.

Note that all the pre-processing steps mentioned above are only for normalized comparison scores calculation instead of quality assessment in our proposed approach. In this

Table 6.2: Experimental parameter settings

Parameter	Values
Scaling factor of training function	1
$R = C$	80
H	2
X	4
Training function	svmtrain in Matlab
'kernal_function' of 'svmtrain'	rbf

paper, all the quality scores are generated from the full-size original samples with full backgrounds. We assume such a segmentation-free quality estimation step is efficient in computation and thus suitable for smartphones since an accurate segmentation algorithm usually requires intensive computations. Nevertheless, some suboptimal-but-efficient segmentation [193] can be used prior to the proposed approach to further reduce the computational complexity.

6.4.3 Dataset preparation and parameter setting

We applied the aforementioned pre-processing steps to the 2100 samples and obtained a foreground-cropped dataset in order to calculate the normalized comparison scores as ground-truth sample quality. The VeriFinger 6.0 Extractor was used to generate the templates from this foreground-cropped dataset. There are only 906 foreground-cropped samples successful in generating templates, which should be attributed to VeriFinger's own sample quality control functionality. In order to create a training set, which covers sufficient high-quality blocks and non-high-quality ones, we selected 29 samples (high-quality ones by visual check) out of those original full samples that generated the 906 templates, and selected 21 samples (low-quality ones by visual check) from the rest 1194 (= 2100 − 906) original full samples. The two groups of selected samples were taken as SVM's training sets. The original captured 2100 fingerprint samples were thus divided into three datasets in our experiments:

Dataset_50 (training set): those original full fingerprint samples used for selecting blocks for SVM training, consisting of two sub-sets (the 29 high-quality samples and the 21 low-quality samples). Figure 6.12 shows some examples of high-quality sample blocks, low quality ones, and background ones respectively.

Dataset_877 (testing set I): there are 877 (= 906 − 29) fingerprint samples used for testing. The corresponding 877 foreground-cropped samples are able to generate templates by VeriFinger 6.0 Extractor. Thus we can calculate a normalized comparison score for each sample in this dataset.

Dataset_1173 (testing set II): there are 1173 (= 1194 − 21) fingerprint samples used for testing. The corresponding 1173 foreground-cropped samples are unable to generate templates by VeriFinger 6.0 Extractor. Thus we set their normalized comparison scores to zero in experiments.

To align the digital resolution roughly equivalent to that of the other two cameras, we enlarged the samples generated by Samsung Galaxy S camera 1.5 times. Other parameters used in our experiments are listed in Table 6.2. The block size $R = C = 80$ was heuristically set in order to meet the ridge density requirement of 4∼10 ridges per block. Table 6.3 gives the statistics of image blocks used in the experiments.

(a) High-quality blocks

(b) Low-quality blocks

(c) Background blocks

Figure 6.12: Examples of high-quality, low-quality, and background blocks from the training set Dataset_50. (the left and right blocks in (c) were from the background of the authors wood-texture office desktop)

Table 6.3: Statistics of sample blocks used in the experiments

Dataset		Amount of block used
Training set	Dataset_50: sub-set_29	77
	Dataset_50: sub-set_21	797
Testing set	Dataset_877	418,430
	Dataset_1173	693,253

6.4.4 The distribution of quality features

It would be interesting to see the distribution of the proposed quality features calculated from the training set. The training set consists of 77 qualified blocks and 797 non-qualified blocks. We compute the feature vectors from the two sets of blocks respectively and give the result in Figure 6.13. The light blue box indicates the qualified case and the dark blue box for the non-qualified case. In general, a good quality feature would be desired to maximize the separability of the two sets of blocks.

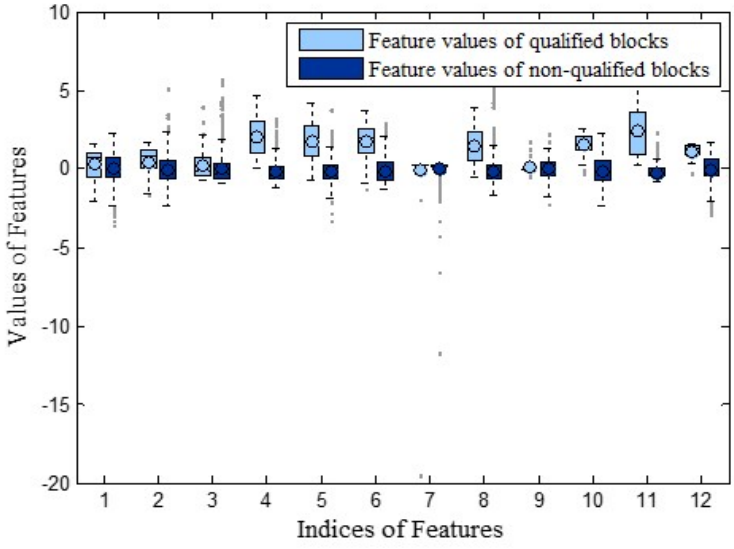
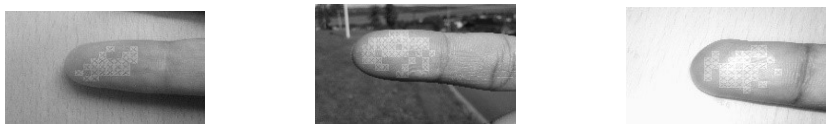


Figure 6.13: Quality feature value distribution

6.4.5 Quality scores generation

Our quality assessment approach addresses the full image without needing segmentation since it regards both low-quality blocks and background ones as non-qualified. We generate a quality score for each sample in Dataset_877 and Dataset_1173. Figure 6.14 gives examples of qualified samples with qualified (high-quality foreground blocks) marked by white cross ('X') blocks.



(a) (b) (c)

Figure 6.14: Samples with high-quality foreground blocks detected in three scenarios: (a) indoor; (b) outdoor; (c) dark.

The quality score is calculated in this way: first we divide a samples quality score (the amount of qualified blocks detected from the sample) by the number of the samples blocks as mentioned in Section 6.2, and then normalize the division result to the dynamic range [0, 100]. The samples quality score is expressed as

$$q_i = \frac{S_i - min(S)}{max(S) - min(S)} \times 100 \tag{6.16}$$

where S_i is the quality score of the i-th sample image in Dataset_877 or Dataset_1173, calculated by Equation 6.2. S is the set of all sample quality scores.

6.4.6 Normalized comparison scores generation

In order to evaluate the performance of the proposed approach, we need to calculate the normalized comparison score $c_i (i = 1, 2, \ldots, 877)$ for the i-th sample in Dataset_877 as its ground-truth quality.

We use the samples with the maximum quality scores calculated from the above subsection as references in normalized comparison scores calculation by Equation 6.1. VeriFinger 6.0 comparator was used to generate comparison score between two templates. To include these reference samples themselves into quality assessment, we need to generate normalized comparison scores for them as well. Namely this requires a reference sample be compared to itself to obtain a genuine comparison score $(i.e., s_m(x_i)$ in Equation 6.1). We assign the globally highest genuine comparison score calculated from two different samples in the experiments to these reference samples as their genuine comparison scores. Figure 6.15 presents the normalized comparison scores distributions over 8 quality score bins in Dataset_877. We can see good correlation between the quality scores and the normalized comparison scores from Figure 6.15b. The next section will quantitatively measure this observed correlation.

6.5 Performance evaluation

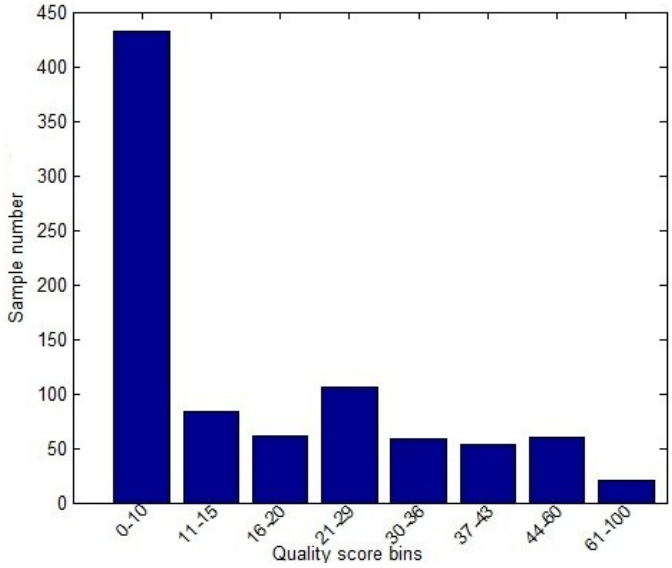
In this section, we evaluate the performance of the proposed approach on two levels: the quality feature level and the holistic approach level. We suggest using three metrics to evaluate a quality assessment approach designed for smartphone camera fingerprint samples: Spearman's rank correlation, Error Reject Curves (ERC) and false detection rate. The three metrics can work in a complementary way focusing on different aspects of the evaluation.

On the quality feature level, we compare the proposed 12-dimensional feature vector with two standardized and widely-used local quality features, namely Local Clarity Score (LCS) [104] and Frequency Domain Analysis (FDA) [104]. For a fair comparison, when a sample quality score is calculated using LCS or FDA, the same pipeline procedures from "Digital resolution adjustment" to "Block orientation alignment" in Figure 6.9 and the same scoring rule as in Equation 6.16 are employed but to replace the proposed 12-dimensional feature vector based SVM decision by a threshold LCS or FDA score decision for each block. For comparison, we also calculated the correlation coefficient by only using the f_{12} in the proposed pipeline by a thresholded f_{12} score decision.

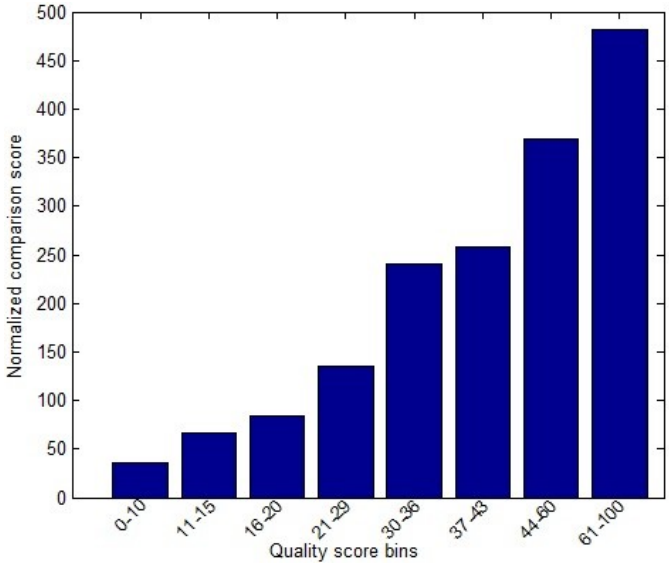
On the holistic approach level, as it is difficult to find in publicized literature such a holistic approach targeted at smartphone-camera fingerprint samples, we can only compare the proposed approach as a whole to the NIST fingerprint sample quality assessment function NFIQ (described in Section 2), as it has been most widely used since being proposed.

6.5.1 Spearman's rank correlation

Computing the Spearman's rank correlation coefficient $\rho(-1 \leq \rho \leq 1)$ is a quantitative method to analyze how well two variables correlate. A value of 1 or -1 indicates being perfectly monotonically correlated, while 0 indicates being uncorrelated. We compute the Spearman's rank correlation coefficient between the normalized comparison score c_i and the quality score q_i generated by the proposed approach, over the two datasets Dataset_877 (testing set that can generate fingerprint templates by VeriFinger 6.0 Extractor) and Dataset_877 + Dataset_1173 (all samples for testing). The results are given in Table 6.4. Note that for all quality scores generated we used original full samples without any segmentation. The results show that the proposed quality assessment approach can accurately predict a sample's quality in terms of higher correlation coefficients compared to NFIQ and the other two features (LCS and FDA) based approaches. We can see NFIQ



(a) Sample number distribution



(b) Normalized comparison score distribution

Figure 6.15: Sample number and normalized comparison score distributions over quality score bins: Dataset_877 (which can generate templates by VeriFinger 6.0 Matcher)

Quality assessment method	Spearman's rank correlation coefficient ρ	
	Dataset_877	Dataset_877 + Dataset_1173
NFIQ	-0.0926	-0.0459
LCS in the proposed pipeline	0.4557	0.4172
FDA in the proposed pipeline	0.5490	0.4266
Only using 12^{th} feature in the proposed pipeline	0.4538	0.4412
Proposed approach	0.6086	0.5851

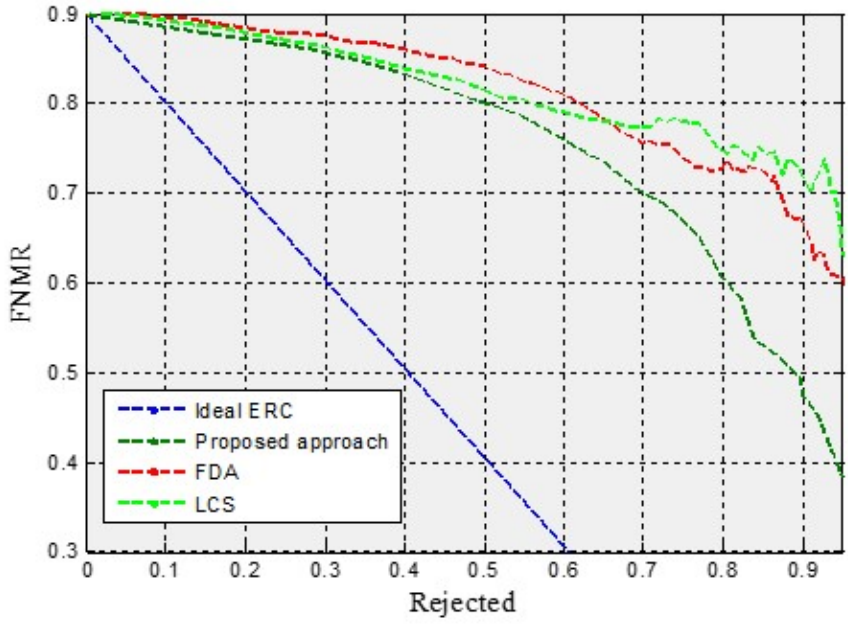
Table 6.4: Sample quality assessment methods comparison by Spearman's rank correlation coefficient

missed the point completely when being used to assess such smartphone-camera fingerprint samples with complicate illumination and background. The result goes coherently with the analysis in Section 6.2.2, $i.e.$, as an effective quality indicator to traditional fingerprint samples, NFIQ was not intended for such smartphone camera fingerprint samples. Compared to LCS and FDA in the same block-based quality assessment pipeline, the proposed 12-dimensional feature vector with SVM classification shows better performance too. To compare individual features, we achieved the correlation performance from f_{12} equivalent to LCS and FDA.

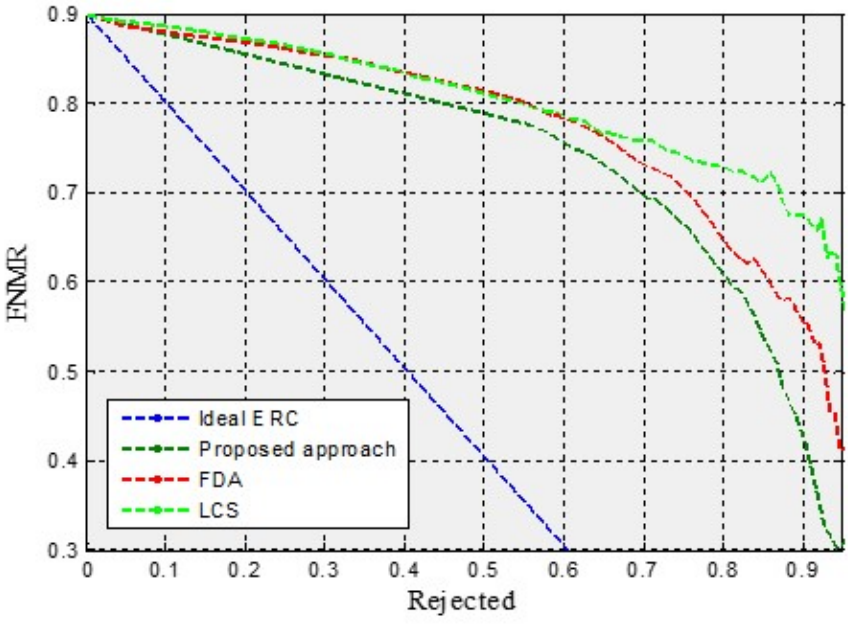
6.5.2 Reject Curves (ERC Error)

Spearman's rank correlation is an efficient way to evaluate the correlation of two variables in a global sense. However, it does not give information how one variable can influence the other in a scalable way. For sample quality control in an operational mode, people may be interested in knowing how to find a suitable threshold for quality control to filter out some low-utility samples in order to achieve better system recognition performance, as a system's error rates (false match rate and false non-match rate) are usually contributed by a few low-utility samples [187]. Error reject curves (ERC) was proposed [85] to address this need to show how quality score threshold tuning (rejecting genuine comparison cases below the quality threshold) can influence the system's false non-match rate ($FNMR$). Each genuine comparison is assigned a quality score by Equation 6.7 in the paper [85], which in our experiment equals to the lower one of the two samples' quality scores. An easy-to-understand way to the ERC metric is - suppose at a certain genuine comparison score threshold we have the a $FNMR$ value, we can expect to reduce this $FNMR$ by rejecting some percentage of the genuine comparison cases (both two samples associated with each comparison) with the lowest quality scores among all. In this sense, the correlation between the $FNMR$ and the quality rejecting percentage can be measured in fine granularity. Figure 6.16 compares the ERC performance of LCS, FDA, and the proposed approach over the two datasets Dataset_877 and Dataset_877 + Dataset_1173. We can see over both datasets the proposed approach excels the other two features based approaches. Here the same pipeline as the proposed approach was used for the two features. The $FNMR$ was initialized at 90% just because this 2100-sample database is very challenging in sample quality in overall.

NFIQ performance was not illustrated in the ERC charts because of its apparently bad performance and sparse quality levels that are difficult to generate a curve.



(a) ERC performance: Dataset_87



(b) ERC performance: Dataset_877 + Dataset_1173

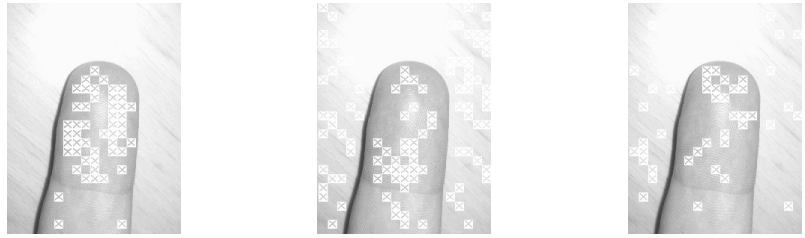
Figure 6.16: Quality assessment methods comparison by error reject curves (ERC).

Table 6.5: Comparison of false detection rate (# Falsely detected block /# all detected blocks)

Quality metrics	False detection rate		
	Dataset_877	Dataset_1173	Dataset_877 + Dataset_1173
LCS in the proposed pipeline	69.85%	90.64%	78.4%
FDA in the proposed pipeline	80.56%	94.9%	86.74%
Only using the proposed 12^{th} feature in the proposed pipeline	12.89%	20.26%	16.16%
Proposed approach	2.67%	11%	4.02%

6.5.3 False detection rate

In our experiments, some samples are susceptible to false detection problem (background blocks labelled as high-quality ones), which mostly occurs in the indoor scenario with the challenging background - the authors' wood-texture office desk surface (shown in Figure 6.3 and Figure 6.4. Such false detection problem may not pose a direct threat to quality assessment if both the foreground and the background are well focused like the examples in Figure 6.17. Fortunately, in our database, most of such samples with challenging background have fair focus on the foreground at the same time, which leads to high correlation between the amount of false detected blocks and the amount of the qualified blocks on the foreground. This fact to some degree suppresses the influence of false detection to Spearman's rank correlation and the ERC performance in this testing database. However, we can envision for some untypical cases, such as no finger is captured in a sample with such challenging background, or the case the background instead of the foreground is focused, false detection will severely impact the performance of a quality assessment approach.



(a) (b) (c)

Figure 6.17: False detection sample for there quality metrics: (a) proposed approach, (b) FDA, and (c) LCS. Qualified blocks marked by cross.

Table 6.5 lists the statistics of false detection under the four block-based quality assessment approaches. We can see the proposed holistic approach performs distinctly better than the same quality assessment pipeline adopting the other two features. On the individual quality feature level, f_{12} exhibits much lower false detection rate than LCA and FDA as well.

6.5.4 Correlation between individual features and the block quality decision

We also evaluated the correlation between each of 12 features and the binary quality decision for each block by computing Spearman's rank correlation coefficient on Dataset_877.

Table 6.6: Correlation between features and block quality decision calculated on database_877.

Features	f_1	f_2	f_3	f_4	f_5	f_6
ρ	0.046	0.005	0.038	0.22	0.22	0.20
Features	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}
ρ	0.20	0.20	0.13	0.19	0.20	0.26

Table 6.7: EERs on levelled quality score groups.

	Group 1	Group 2	Group 3
Quality score	0-6	7-24	25-100
Samples number	375	264	238
Number of genuine scores	287	181	161
Number of imposter scores	22960	11765	10626
EER	25.6%	20.9%	6.8%

There are 418,430 blocks in Dataset_877 as listed in Table 6.3. Thus 418,430 values for each feature can be computed, meanwhile 418,430 binary decisions can be produced by the SVM classifier. The Spearman's rank correlation can be calculated for the two sets of data. Table 6.6 shows the correlation results for each feature on Dataset_877. We can see the DAI descriptor based features have stronger correlation with the binary decision comparing to the pixel based features.

6.5.5 Purpose verification of quality assessment: EER under different levels of quality scores

Recall that the purpose of sample quality assessment is to select high quality samples for recognition use. To verify whether this purpose is achieved by the proposed approach or not, we calculate EERs under three levels of quality scores using VeriFinger 6.0 comparator. The sample with maximum quality score is always selected as the reference sample for each finger in all experiments. We divided the quality score range [0, 100] into three subranges: [0, 14], [15, 33], and [34, 100] for the testing set Dataset_877. The experimental results are given in Table 6.7. We observe that EERs are significantly reduced along the increase of sample quality, which demonstrates the effectiveness of our proposed quality assessment approach in predicting the quality of fingerprint samples generated by the three smartphone cameras used in our experiments.

6.6 Conclusion and future work

To evaluate the quality of a fingerprint sample captured by a smartphone camera, we proposed an effective quality assessment approach, which processes a captured fingerprint sample by a block-based feature extraction pipeline. An accurate block ridge pattern descriptor Differential-Autocorrelative-Integration (DAI) was proposed for extracting quality features from each image block. In total 12 quality features in three types, namely pixel-based, DAI based, and DAI spectrum based, are extracted from each image block to form a 12-dimensional quality feature vector. SVM is trained and used to make a binary de-

cision "qualified" or "non-qualified" for each feature vector. In addition, a 2100-sample smartphone camera fingerprint database is created to test the proposed approach.

In addition to better correlation with the ground-truth sample quality and lower block false detection rate, our approach differs from existing fingerprint quality assessment approaches in the following aspects:

(1) The proposed approach directly detects high-quality foreground blocks and discards those low-quality foreground blocks and background blocks, therefore needing no segmentation of the foreground in advance. We call it a 'one-stop-shop' approach in this sense. This could be favored by mobile devices with constrained computation resources since accurate segmentation against complicate backgrounds usually requires intensive computation or performs unstably under varied illumination or backgrounds. Nevertheless, the proposed approach can work in harmony with an accurate and stable pre-segmentation algorithm if any;

(2) The sample processing pipeline proposed in this paper, including the block orientation alignment, block-based quality feature vector generation, block-based SVM classifier, and the scoring rule for a sample, is structured in a way that the different processing steps can be easily maintained. This makes the proposed approach in essence open to any improvement in performance. For instance, new quality features proposed in the future can be easily plugged into the pipeline for performance testing. We had already done this to two standardized features (LCS and FDA) in this paper.

Note that the 12 quality features proposed in this paper should not be deemed as the best ones for the purpose of smartphone camera fingerprint sample quality assessment. We adopt them only for characterizing a block pattern from different quality-related aspects.

Though targeting at smartphone camera fingerprint samples, the proposed approach can be reasonably generalized to other biometric system using touchless fingerprint sensors requiring effective and efficient sample quality control in unpredictable working environments, such as portable touchless fingerprint identification terminals used by law enforcement staffs.

Moreover, we studied the characteristics of smartphone camera fingerprint samples, explained why traditional approaches are incapable towards such samples, and verified these explanations by experimental results. Accordingly, three performance evaluation metrics for quality assessment on smartphone camera fingerprint samples were suggested based on their complementary focuses.

Future work is planned in the following aspects:

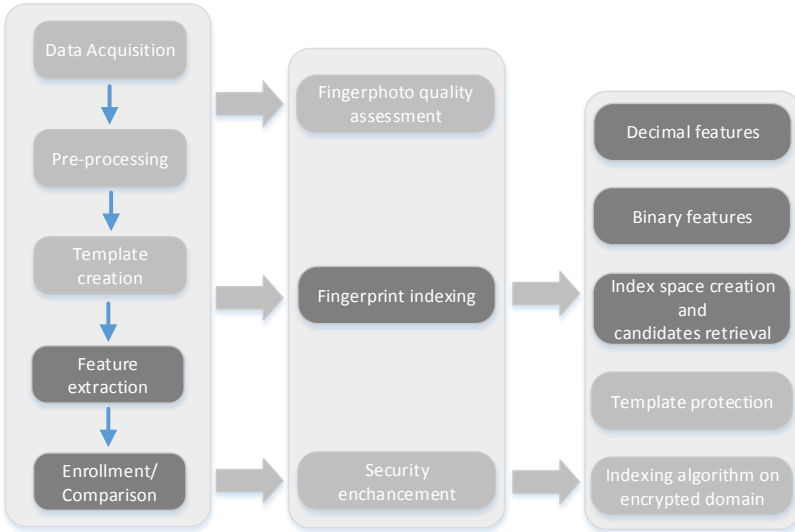
(1) Real-time sample quality assessment from a smartphone's preview video sequence will be investigated;

(2) New efficient quality features, especially features to qualify high-curvature blocks;

(3) For better user convenience, methods for automatic resolution alignment and interactive focusing will be integrated.

Part III

Fingerprint Indexing



This part focuses on fingerprint indexing which intended to answer two research questions RQ_2: **Besides the existing feature extraction methods in literature, what features can still be extracted from the fingerprint template and outperform the existing ones?** and RQ_3: **How to build the index space and retrieve potential candidates for fingerprint indexing algorithm?**

This part is composed of three chapters corresponding to three fingerprint indexing approaches. The first approach in Chapter 7 proposed approach is a score-level fusion approach which combines a newly designed fingerprint indexing method and a state-of-the-art fingerprint indexing method. The proposed feature extraction method uses minutia information and ridge information around the location of each minutia to develop a feature vector including 9 components. The experimental result show the feasibility of the proposed fingerprint indexing approach.

The fingerprint indexing approach developed in Chapter 8 only relies on minutia information to extract a fixed-length decimal feature vector from a minutiae vicinity which is formed by a central minutia and its three closest minutiae. The feature vector generated in this approach consists of 12 components calculated from minutia location formation, and 12 components calculated from minutia direction information. In this chapter, we also proposed to divide a single index space into four separate index tables based on the minutia's direction.

The work in Chapter 9 also focuses on using minutia information in order to be robust against fingerprint sample translation and rotation, but the proposed approach generates a binary template which is further used for creating index space. The advantage of a binary template is that operating on binary values would be potentially faster than operating on decimal values, while the efficiency is one of the major requirements for the large-scale system. In addition, the proposed approach has also been evaluated in a large-scale synthetic dataset, and the results are presented as a benchmark for a secure fingerprint indexing approach in Chapter 11.

The work in Chapter 7 was published in [129]: GUOQIANG LI, BIAN YANG, CHRISTOPH BUSCH. "A score-level fusion fingerprint indexing approach based on minutiae vicinity

and minutia cylinder-code". In Biometrics and Forensics (IWBF), 2014 International Workshop on (pp. 1-6). IEEE.

The work in Chapter 8 was published in [131]: GUOQIANG LI, BIAN YANG, CHRISTOPH BUSCH. "A Novel Fingerprint Indexing Approach Focusing on Minutia Location and direction". In Identity, Security and Behavior Analysis (ISBA), 2015 IEEE International Conference on (pp. 1-6). IEEE.

The work in Chapter 9 was published in [130]: GUOQIANG LI, BIAN YANG, CHRISTOPH BUSCH. "A Fingerprint Indexing Scheme with Robustness against Sample Translation and Rotation". In Biometrics Special Interest Group (BIOSIG), 2015 International Conference of the (pp. 1-8). IEEE.

Chapter 7

A Score-level Fusion Fingerprint Indexing Approach based on Minutiae Vicinity and Minutia Cylinder-Code

Abstract

Due to the uniqueness and permanence properties of the biometric fingerprint characteristic, a number of large-scale biometric applications (such as the Visa Information System (VIS) in Europe, US-VISIT / IDENT system in the USA and the Aadhaar project in India) are based on fingerprint recognition. These systems generally contain millions of fingerprint samples. In order to improve the efficiency in seeking for suitable candidate reference data in such large-scale databases, studying indexing techniques for fingerprints is desirable. In this paper, we design a new indexing method using the features extracted from minutia details including location, direction and ridge information. Then a score-level fusion indexing approach is proposed by combining this new method with the minutia cylinder-code (MCC) indexing method. The results demonstrate the improvement of the proposed approach based on experiments on several public fingerprint databases.

7.1 Introduction

Due to the properties regarding uniqueness and permanence of fingerprints, the use of fingerprint identification is an essential component in governmental applications. Several national projects have adopted fingerprint identification such as the Visa Information System (VIS) in Europe, the US-VISIT / IDENT program and India's Aadhaar project. These systems generally contain millions or even billions of fingerprints samples. For example, US-VISIT program includes 68 millions enrolled applicants with fingerprints samples from ten fingers [165], and India's Aadhaar project [15] is planning to cover all residents in India where the population is over 1.2 billion. On the other hand commercial of the shelve fingerprint comparison subsystems (e.g the Verifinger comparator used in the work) can compare up to 60,000 fingerprints per second [32]. This indicates that at least 100 seconds are required to compare 6 millions fingerprints which is only a fraction of a national database. Additional time for other procedures in the context of an identification system must be considered such as sample capture and pre-processing. Obviously, it is important to improve the efficiency while seeking for suitable references in a large-scale database, ensuring at the same time high recognition accuracy. In order to achieve this goal, fingerprint indexing techniques, which reduce the response time by selecting a number of candidates from the reference database for further comparison, have been studied during the last two decades.

Fingerprint indexing methods proposed in the literature can be roughly classified into three categories based on the features used in their approaches:

- local feature based: some researchers extracted the features from the local ridge-line orientation [76, 51]. Most of researchers have been working on deriving features from groups of fingerprint's minutiae using Delaunay triangulation [137, 50, 138, 150, 172].

- global feature based: the whole orientation field, ridge-line frequency and singular points have been used to construct feature vectors using for indexing fingerpirnt database [201, 110].
- other features based: Li et al. proposed a method based on the symmetric filters [136], scale invariant feature transformation (SIFT) also has been used to develop a fingerprint indexing scheme in article [182].

Our proposed method will follow the local feature based approaches which is based on extracting features from minutiae details and triplets formed by minutiae. However we will use the triplets contained in a minutiae vicinity, which is defined in [213], instead of applying Delaunay triangulation, since fingerprint sample noise and other distortions can seriously affected the whole structure by using Delaunay triangulation [197]. We extract nine features from a triplet which is contained in a minutiae vicinity and use these features to design an indexing scheme, which we call Minutiae-Vicinity-Index (MV-Index) method. Meanwhile, with the advantages of parallel processing techniques and high performance of multi-processors, it is feasible to implement two separate indexing methods and fuse the similarity scores to produce the candidates without effective overhead for the identification transaction. Therefore, we propose a score-level fusion indexing approach in this paper. The second indexing method used for fusion is minutia cylinder-code indexing method (called MCC-Index method), which is considered as a state-of-the-art indexing approach proposed by Cappelli et al. in [62]. We will use 'MV-MCC fusion'to denote this score-level fusion indexing approach in our description.

The rest of this paper is organized as follows: the proposed approach will be described in Section 7.2; Section 7.3 will introduce experimental results and Section 7.4 will conclude this paper.

7.2 Proposed indexing approach

7.2.1 Structure of MV-MCC fusion approach

In our proposed approach, MV-Index method and MCC-Index method will be used to create two separate index spaces during enrolment stage as illustrated in Figure 7.1. During biometric identification ($i.e.$, retrieval stage), two sets of similarity scores can be produced by searching the MV-Index space and MCC-Index space respectively. We normalize these two groups of similarity scores into range the [0, 100] before processing them to the fusion component. The fused similarity score can be obtained by simply combing the two normalized similarity scores generated by MV-Index and MCC-Index. At last, the candidates can be determined by selecting top-X samples after sorting the fusion scores. The details on the MCC-Index were published recently by Cappelli et al. [62]. Thus we concentrate here on the description of how to create the MV-index space and how to retrieve candidate entries from the MV-Index space.

7.2.2 Creation of an index space using MV-Index

7.2.2.1 Feature Extraction

According to the definition in [213], a minutiae vicinity is a basic unit which is formed by four minutiae including a center minutia O and its three closest neighboring minutiae O_1, O_2 and O_3 sorted by ascending order based on their Euclidean distance with O. Figure 7.2 illustrates a minutiae vicinity and four non-redundant triplets.

Assuming there is a fingerprint sample including n minutiae with four properties which are location, direction, ridge curvature and the ridge density around the location of the minutia neighbors. We can use the neighbors of a center minutiae to compose n minutiae vicinities, where each minutiae vicinity consists of four non-redundant triangles (i.e.

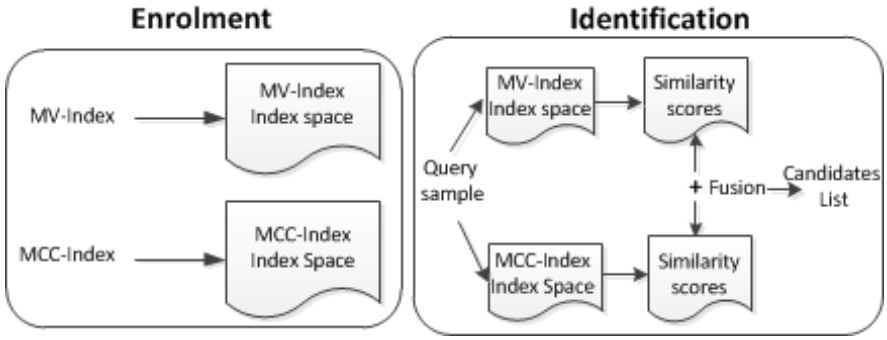


Figure 7.1: Structure of proposed approach.

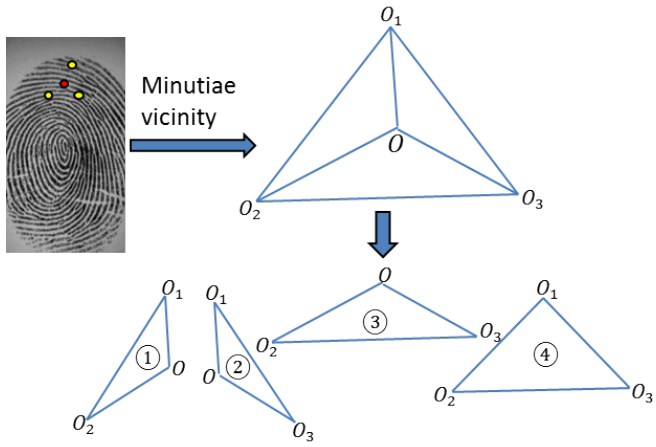


Figure 7.2: Minutiae Vicinity

triplets of minutiae). Further we will extract 9 features from each triplet to constitute a feature vector (called MV based feature vector). 4 feature vectors can be obtained from a minutiae vicinity. In total, $4 * n$ feature vectors can be derived from a fingerprint sample. Each feature vector contains 9 components which can be classified into 3 categories:

- 4 features are calculated by combining the following geometric traits of the triplet: three side lengths (l_1, l_2, l_3), three internal angles $(\alpha_1, \alpha_2, \alpha_3)$ and the triangle area. Instead of directly using these geometric traits, we choose the combination of these traits in terms of robustness, which has been demonstrated by Liang, Ross and others [138, 172].

$$f_1 = cos(max(\alpha_1, \alpha_2, \alpha_3)) \tag{7.1}$$

$$f_2 = \sqrt{\ell * (\ell - l_1) * (\ell - l_2) * (\ell - l_3)} \tag{7.2}$$

where $\ell = \frac{l_1+l_2+l_3}{2}$.

$$f_3 = \frac{4(l_1 + l_2 + l_3)\sqrt{(l_1 + l_2 + l_3)}}{\sqrt{(l_1 + l_2 - l_3)(l_1 + l_3 - l_2)(l_2 + l_3 - l_1)}} \tag{7.3}$$

$$f_4 = \frac{max(l_1, l_2, l_3)}{min(l_1, l_2, l_3)} \tag{7.4}$$

- 3 features are computed for each triplet based on orientational differences between the minutia direction (d_1, d_2, d_3) of the constituting points.

$$f_5 = abs(d_1 - d_2) \tag{7.5}$$

$$f_6 = abs(d_2 - d_3) \tag{7.6}$$

$$f_7 = abs(d_3 - d_1) \tag{7.7}$$

  where abs denotes the absolute value.

- 2 features are the average values of ridge density (r_1, r_2, r_3) and ridge curvature (c_1, c_2, c_3) from three minutiae. The values of ridge density and curvature are obtained by NeuroTechnology Verifinger 6.0. The value ranges of ridge density and curvature are both $[0, 255]$.

$$f_8 = (r_1 + r_2 + r_3)/3 \tag{7.8}$$

$$f_9 = (c_1 + c_2 + c_3)/3 \tag{7.9}$$

7.2.2.2 MV-Index space creation

There are two steps in the stage of MV-Index space creation:

First step Generate the clusters. A training set which is a set of fingerprint samples will be used to generate MV based feature vectors. The unsupervised learning scheme $K\text{-}means$ is applied to cluster these feature vectors into K clusters. Each cluster will be represented by a single MV based feature vector called cluster center. We empirically select $K = 2400$ in our experiments.

Second step Create the MV-Index space, which will be represented by a matrix $\boldsymbol{M}$ whose size is $R * K$, where R is the number of subjects enrolled in the database, and K is the number of clusters. In fact, the rows of this matrix indicate the indices of the subjects and columns denote the indices of clusters.

Multiple samples from the same subject can be enrolled into database in our method. Firstly, MV based feature vectors will be extracted from each sample. We assign each feature vector to the closest cluster by computing Euclidean distance between this feature vector and cluster center. Assuming the index of this subject is r and the index of closest cluster is k, then we will set $\boldsymbol{M}(r, k) = 1$ if $\boldsymbol{M}(r, k) == 0$ ($\boldsymbol{M}(r, k)$ is initiated as $\boldsymbol{M}(r, k) = zeros(R, K)$) which means no feature vector has been assigned to this cluster yet. Our algorithm will go through each feature vector and each sample. Since these feature vectors and samples are from the same subject (r), they will only change the values in r^{th} row of the index space matrix $\boldsymbol{M}$. At last, a K-dimension binary vector which is the r^{th} row in matrix $\boldsymbol{M}$ will be used to represent this subject as seen in Figure 7.3. The details of MV-Index space creation are described in **Algorithm** 7.1.

7.2.3 Retrieving candidate entries with the MV-Index space

When a fingerprint probe sample is presented to the MV-Index space, a sorted list of similarity scores will be returned. This score list can be directly used to select the candidates from the biometric reference database or the score can be fused with similarity scores generated from a complementary indexing method such as the MCC-Index method in our work. During retrieval processing in MV-Index space, the MV based feature vectors will be extracted from the probe sample first. Then each feature vector is assigned to the closest cluster using Euclidean distance rule as the same procedures during enrolment. A K-dimension binary string can be generated to represent this probe sample. Computing the

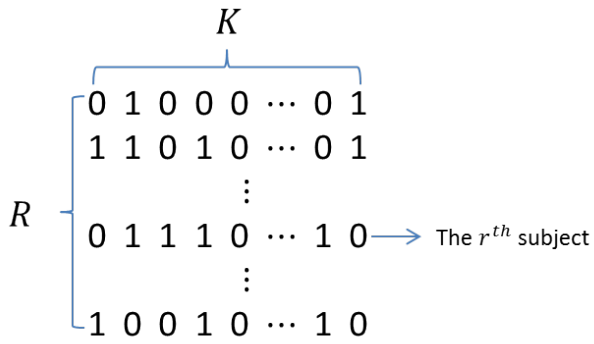


Figure 7.3: MV-Index space is denoted by a Matrix $\boldsymbol{M}_{R\times K}$, where R is the number of subjects and K is the number of clusters. The r^{th} row represents the r^{th} subject.

Algorithm 7.1 Minutiae vicinity based index space creation

Require: Fingerprint samples used for training set;
$R * M$ Reference samples used for enrolment: R subjects, M samples for each subject;
Ensure: MV-Index space: a matrix $\boldsymbol{M}_{R\times K}$
Initiate MV-Index space matrix: $\boldsymbol{M}_{R\times K} = zeros(R, K)$
Extract MV feature vectors from training set, and apply K-means scheme to generate K clusters;
for each subject r $(1 \leq r \leq R)$ **do**
  for each sample m $(1 \leq m \leq M)$ **do**
    Extract MV based feature vectors $\{V_{(r,m,1)}, ..., V_{(r,m,T_{r,m})}\}$ from sample m, where $T_{r,m} = 4 * n$, assuming sample m includes n minutiae;
    for each feature vector $V_{(r,m,t)} (1 \leq t \leq T_{r,m})$ **do**
      Find the closest cluster k using Euclidean distance rule;
      if $\boldsymbol{M}(r, k) \neq 1$ **then**
        $\boldsymbol{M}(r, k) = 1$
      end if
    end for
  end for
end for

logical conjunction between this binary string against each row in the MV-Index space matrix $\boldsymbol{M}_{R\times K}$ will output R similarity scores. The candidate entries can be determined by sorting these similarity scores in a descending order and selecting the top X scores. The procedures of retrieving candidate entries in the MV-Index space are explained in **Algorithm** 7.2.

7.3 Experimental set-up and results

7.3.1 Performance measures and experimental setting

According to the description in ISO/IEC FDIS 19795-1 [101], the performance of indexing methods is usually measured by $Pre\text{-}selection\ error\ rate$ and $Penetration\ rate$. The $Pre\text{-}selection\ error\ rate$ is defined as a proportion of enrolled references where the enrolled reference corresponding to the query sample (or probe sample) is not involved in

Algorithm 7.2 Candidate retrieval from minutiae vicinity based index space

Require: MV-Index space: a matrix $\boldsymbol{M}_{R\times K}$;
Probe sample P which includes p minutiae;
K clusters;
Ensure: A sorted similarity scores list: $SL = \{(r, S_r)\}$, where r is the index of reference, and S_r is the corresponding similarity score;
1: Initiate $S_r = 0, (r = \{1, 2, ..., R\})$;
2: Initiate a K-dimension binary string $\boldsymbol{B}$, where $\boldsymbol{B}_k = 0, (k = \{1, 2, ..., K\})$;
3: Extract T_P feature vectors $\{V_1, V_2, ..., V_{T_P}\}$ from probe sample P, where $T_P = 4 * p$;
4: **for** each feature vector V_t **do**
5: Find the closest cluster k using Euclidean distance rule;
6: **if** $\boldsymbol{B}_k \neq 1$ **then**
$\boldsymbol{B}_k = 1$
7: **end if**
8: **end for**
9: **for** each row r in matrix $\boldsymbol{M}_{R\times K}$ **do**
10: $S_r = |\boldsymbol{B} \cap \boldsymbol{M}(r, :)|$;
11: where '$\cap$' denotes the logical conjunction 'AND', '$|\cdot|$' is to count the number of 1 in a binary string;
12: **end for**
13: Sort $\{S_1, S_2, ..., S_R\}$ in descending order to produce the score list $SL = \{(r, S_r)\}$;
14: Select the top X scores to output the candidates;

the pre-selected candidates which will be used for further comparing with the probe sample. The $Penetration\ rate$ is a proportion of enrolled references in a database where the system has to search. We will evaluate the performance of proposed method using these two measures.

Our experiments include two parts due to two types of setting: Setting One used the public database FVC_2004_DB4 as training set whose fingerprint samples were synthesized in order to be comparative with the approach proposed by Ross et al. [172]; Setting two used the database FVC_2002_DB2_a as training set. The details of databases preparation will be described in the following section. The commercial NeuroTechnology Verifinger 6.0 SDK has been applied to extract the fingerprint templates including minutiae location, direction, ridge density and ridge curvature which were used in our experiments.

7.3.2 Performance evaluation of Setting One

In order to benchmark our proposed method with the indexing approach by Ross et al. [172], which has chosen the Delaunay triangulation to form triplets, we selected the same training set (public database FVC_2004_DB4) and test sets (public databases FVC_2004_DB1 and FVC_2004_DB2) with [172]. Each database contains 880 samples from 110 subjects (8 samples per subject). We divided each test set into two parts: the first three samples from the same subject were used to create the index space, and the rest of five samples were used as probe samples during identification stage. Table 7.1 shows these dataset preparation in Setting One. We conducted the experiments using indexing approach in [172], MV-Index and MCC-Index methods. MCC-Index space has been generated by MCC sdk2.0 released by Cappelli et al. [61, 62, 79]. We reflected the $Hit\ rate$ in [172] which indicates a percentage of query samples that are correctly retrieved to $Pre\text{-}selection\ error\ rate$, which means the system has 5% $Pre\text{-}selection\ error\ rate$ if its $Hit\ rate$ is 95%. Table 7.2 and Table 7.3 list the performance in terms of $Pre\text{-}selection\ error\ rate$ and $Penetration\ Rate$ calculated from two test databases respectively. We can observe that MV-MCC fusion method has achieved the best performance. It reduces the $Penetration\ Rate$ from 29.5% achieve by MCC-Index

Table 7.1: Dataset preparation of Setting One. Each test set has been divided into two parts: the first three samples from the same subject are used to enrol as references, the rest of samples are used as probes during identification.

	Dataset		Subjects' number	Samples' number
Training set	FVC_2004_DB4		110	880
Test set	FVC_2004_DB1	References	110	330
		Probes	110	550
	FVC_2004_DB2	References	110	330
		Probes	110	550

Table 7.2: Performance evaluation of MV-MCC fusion, MCC-Index, MV-Index and Ref. [172] on database FVC_2004_DB1.

Pre-selection Error Rate (%)		20	15	10	5
Penetration Rate(%)	Approach in [172]	40.04	43.03	45.97	48.75
	MV-Index	12	17.38	26	48.5
	MCC-Index	<1	2.94	7	16.5
	MV-MCC fusion	<1	<1	1.19	8.9

Table 7.3: Performance evaluation of MV-MCC fusion, MCC-Index, MV-Index and Ref. [172] on database FVC_2004_DB2.

Pre-selection Error Rate (%)		20	15	10	5
Penetration Rate(%)	Approach in [172]	40.79	43.61	46.45	49.34
	MV-Index	10	18.25	32.33	45.75
	MCC-Index	1.8	4.93	15	29.5
	MV-MCC fusion	<1	<1	2.4	9.75

method to 9.75% at 95% $Hit\ rate$ for database FVC_2004_DB2. And the MV-Index method outperforms the approach proposed in [172].

7.3.3 Performance evaluation of Setting Two

In a practical scenario, using one sample to enrol is quite common in a biometric identification system. Thus we choose only the first sample from each subject to enrol in database during index space creation, the remaining samples from the same finger were used as probe samples during identification stage. The training set in this part was the samples from public database FVC_2002_DB2_a which includes 800 samples from 100 subjects. The test sets involve three public database FVC_2004_DB1_a (800 samples from 100 subjects), FVC_2004_DB2_a (800 samples from 100 subjects) and FVC_2006_DB3_a (1680 samples from 140 subjects). Table 7.4 lists the database information used in this experiment. Figure 7.4 – Figure 7.6 illustrate the performance of the proposed approach and MV-Index on these three databases. We can see the proposed MV-MCC fusion approach reduced

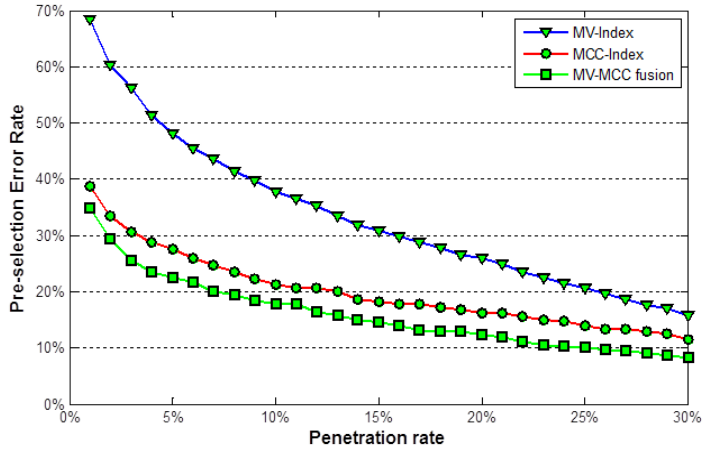


Figure 7.4: Performance evaluation of MV-MCC fusion, MCC-Index and MV-Index on database FVC_2004_DB1_a

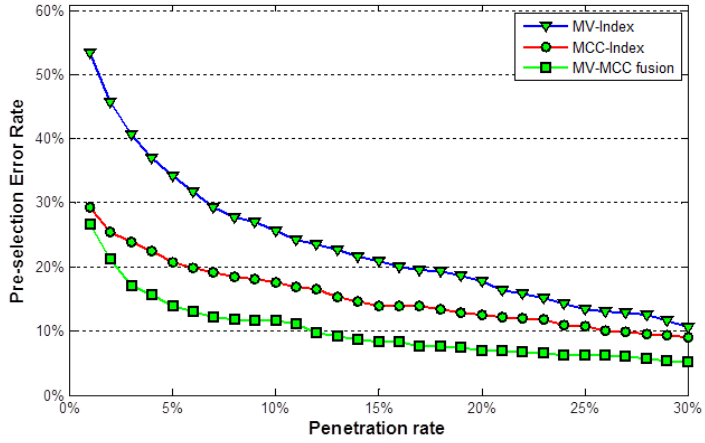


Figure 7.5: Performance evaluation of MV-MCC fusion, MCC-Index and MV-Index on database FVC_2004_DB2_a

Table 7.4: Dataset preparation of Setting Two. Each test set has been divided into two parts: the first sample from the same subject are used to enrol as references, the rest of samples are used as probes during identification.

	Dataset		Subjects' number	Samples' number
Training set	FVC_2002_DB2_a		100	800
Test set	FVC_2004_DB1_a	References	100	100
		Probes	100	700
	FVC_2004_DB2_a	References	100	100
		Probes	100	700
	FVC_2006_DB3_a	References	140	140
		Probes	140	1540

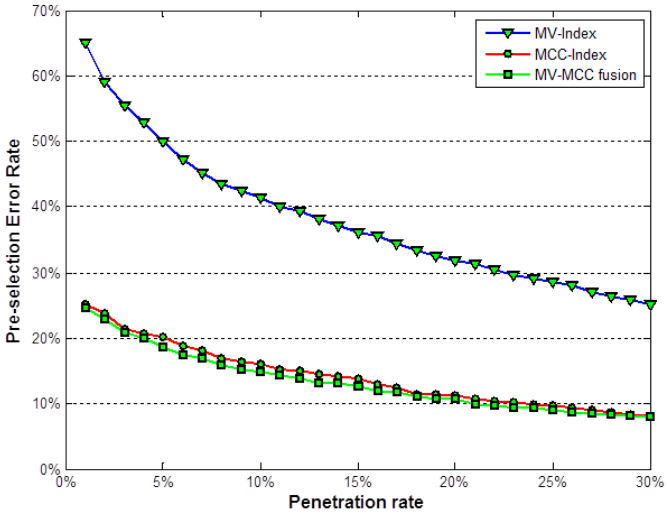


Figure 7.6: Performance evaluation of MV-MCC fusion, MCC-Index and MV-Index on database FVC_2006_DB3_a

the *Pre-selection error rate* in all three database comparing to MCC-index method and MV-Index method. Especially in database FVC_2004_DB2_a, the improvement is more noticeable.

7.4 Conclusion

In this paper, we proposed a score-level fusion indexing approach by combing a new designed indexing method and minutia cylinder-code indexing method. The new designed indexing method extracts a feature vector including 9 components from minutiae details and a triplet which is contained in a minutiae vicinity. *K-means* learning scheme has been

applied to cluster the training features. A binary string with fixed length can be generated by assigning each features vector from the subject's samples to its closest cluster. We store this binary string to create the index space and to represent the subject. Experiments have been conducted on several public databases from FVC_2004 and FVC_2006. The results demonstrate the improvement of proposed approach. Our future work will focus on improving the performance of minutiae vicinity based indexing method.

Chapter 8

A Novel Fingerprint Indexing Approach Focusing on Minutia Location and Direction

Abstract

Biometrics identification systems containing a large-scale database have been gaining increasing attention. In order to speed up searching in a large-scale fingerprint database, fingerprint indexing algorithm has been studied and introduced into biometrics identification system. One critical component of a fingerprint indexing algorithm is the feature extraction method. Majority of researchers developed the features by combining minutia with other information, such as ridge, singularities, orientation filed, etc. Instead, this paper will focus on only using minutia location and direction to extract features. The performance of proposed fingerprint indexing approach was evaluated on several public databases by being compared to the start-of-the-art fingerprint indexing method - minutia cylinder-code (MCC) - indexing as a benchmark. The experimental results show that the proposed approach gives equivalent performance or even outperforms MCC indexing method on the tested databases.

8.1 Introduction

Deploying and investing on biometrics identification systems has been gaining increasingly attention recently, such as the Visa Information System (VIS) [165] in Europe, the US-VISIT / IDENT program and India's Aadhaar project [15]. A common characteristic of these systems is containing a large-scale database storing the biometric data of the subjects. This characteristic requires biometrics indexing techniques to facilitate searching in the large-scale database, especially for de-duplication checking which only uses biometrics data for searching. In this paper, we focus on fingerprint indexing technique, since fingerprint is the most commonly used modality in biometrics identification systems. Fingerprint indexing technique is to reduce the number of candidate identities which will be further used by the verification algorithm. Figure 8.1 shows a brief structure of fingerprint indexing approach which contains two major stages: enrolment and candidates retrieval. The purpose of fingerprint indexing approach is to output a list of candidates accurately and efficiently. As seen in Figure 8.1, the feature extraction is a critical component in a fingerprint indexing algorithm. After reviewing the articles in the literature, we can roughly classified the feature extraction methods into three categories:

- local feature based: some researchers extracted the features from minutiae location and direction information [137, 138, 150, 172, 62], ridge density, local ridge-line orientation [76, 51], etc;
- global feature based: some researchers explored the features extracted from the whole orientation field, ridge-line frequency or singular points [201, 110].
- other features based: another researchers worked on transformation algorithms or filters based feature extraction methods, for example, using symmetric filters in paper [136] and scale invariant feature transformation (SIFT) in paper [182].

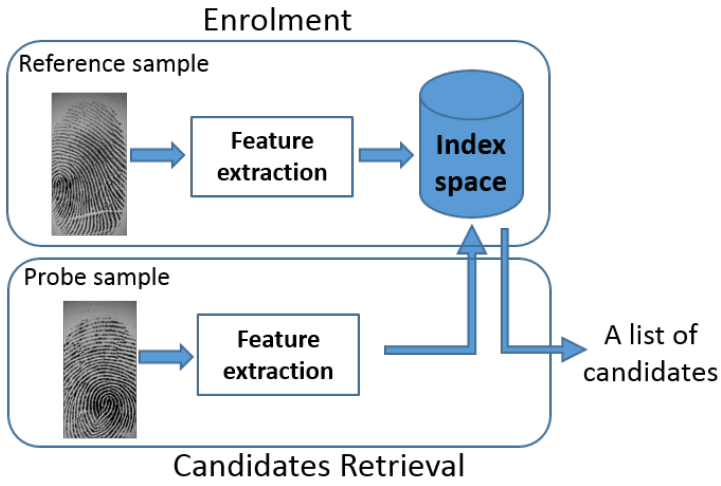


Figure 8.1: A brief structure of fingerprint indexing approach.

Majority of these feature extraction methods combine minutia location and direction with other information, such as ridge curvature, orientation filed, singularities, etc. However, it will be favourable that we can extract features only using minutia information when it comes to large-scale database, since minutia information is most robust and commonly used . Therefore this paper will focus on extracting features based on minutia location and direction. A typical method that only uses minutia information is minutia cylinder-code (MCC) indexing method proposed by Cappelli et al. [62], which also can be considered as a state-of-the-art fingerprint indexing method (denoted by MCC-Index in this paper). MCC-index method adopts MCC feature developed in paper [61] and builds the index space based on Locality Sensitivity Hash (LSH). We also use this method as a benchmark to evaluate the performance of proposed approach.

The main contributions of this paper are: we adapted the concept of minutiae vicinity defined in paper [213] to form a set of minutiae vicinities around a central minutia in order to be resilient to the fingerprint sample variation; we extract a new feature vector consisting of 24 components from a minutiae vicinity to create the index space; we proposed a new method to create index space which is to generate four separate index tables according to the minutia direction, and the similarity score is outputted by fusing four scores generated from four index tables. The remaining of this paper is organized as follows: Section 8.2 describe the details of proposed approach; Section 8.3 evaluates the performance and discusses the experimental results; the conclusion is drawn in Section 8.4.

8.2 Proposed fingerprint indexing approach

As depicted in Figure 8.1, a fingerprint indexing scheme consists of three basic components: feature extraction, index space creation and candidate retrieval. The details of these three components are described in the following subsections.

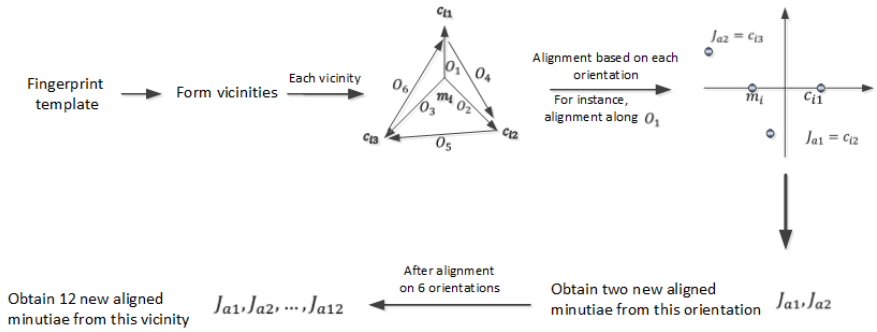


Figure 8.2: Procedures of generating new aligned minutia.

8.2.1 Feature extraction

The feature extraction method used in this work was inspired by the features developed by Yang et al. [215] which were extracted based on a minutiae vicinity (MV) defined by four minutiae including a central minutia and three neighboring minutiaein paper [213]. The concept of minutiae vicinity used in this paper is slightly different with the definition defined in paper [213] where they consider 3 'closest' neighboring minutiae. Then each minutia, which is considered as central minutia, will lead to only one minutiae vicinity. In order to be robust to missing minutiae or spurious minutiae in fingerprint samples which commonly occur, we consider 5 closest minutiae to form a set of minutiae vicinities around a central minutia. For instance, we assume a minutia m_i which is one of minutiae in a fingerprint sample S. We sort the rest of minutiae in an ascending order according to the Euclidean distance to m_i. $(c_{i1}, c_{i2}, c_{i3}, c_{i4}, c_{i5})$ indicate the 5 closest neighboring minutiae. Then a set of minutiae vicinities of m_i can be formed by applying the following rules:

- First minutiae vicinity is formed by $(m_i, c_{i1}, c_{i2}, c_{i3})$;
- Second minutiae vicinity is formed by $(m_i, c_{i2}, c_{i3}, c_{i4})$ assuming c_{i1} is missing;
- Third minutiae vicinity is formed by $(m_i, c_{i1}, c_{i3}, c_{i4})$ assuming c_{i2} is missing;
- Forth minutiae vicinity is formed by $(m_i, c_{i1}, c_{i2}, c_{i4})$ assuming c_{i3} is missing;
- Fifth minutiae vicinity is formed by $(m_i, c_{i3}, c_{i4}, c_{i5})$ assuming (c_{i1}, c_{i2}) are both missing;
- Sixth minutiae vicinity is formed by $(m_i, c_{i2}, c_{i4}, c_{i5})$ assuming (c_{i1}, c_{i3}) are both missing;
- Seventh minutiae vicinity is formed by $(m_i, c_{i1}, c_{i4}, c_{i5})$ assuming (c_{i2}, c_{i3}) are both missing.

After forming the minutiae vicinities, 12 new aligned minutiae can be obtained from each minutiae vicinity by applying the procedures depicted in Figure 8.2. In each minutiae vicinity, we define 6 orientations which are connected between minutiae pairs illustrated in Figure 8.2. A new coordinate system can be constructed by using one orientation (defined by two minutiae) as X-axis and the middle point of this orientation as the origin. Another two minutiae are aligned to this new coordinate system, thus we can obtain two newly aligned minutiae from each orientation. For instance, we do alignment along orientation O_1 defined by m_i and c_{i1} as seen in Figure 8.2. Then we will have two newly aligned minutiae (J_{a1}, J_{a2}) which correspond to the original minutiae (c_{i2}, c_{i3}). In total, we will obtain

12 newly aligned minutiae $(J_{a1}, J_{a2}, ..., J_{a12})$ after operating alignments on 6 orientations respectively. Each new minutia contains three attributes: minutia's X coordinate $J_{(aj,x)}$, minutia's Y coordinate: $J_{(aj,y)}$ and minutia's direction $J_{(aj,d)}$, where $j \in \{1, 2, ..., 12\}$. In consideration of computational complexity, we calculate Manhattan distance between minutia and its corresponding origin indicated by Equation 8.1.

$$H_{aj} = abs(J_{(aj,x)}) + abs(J_{(aj,y)}) \tag{8.1}$$

A 24-D feature vector is constructed by concatenating these Manhattan distances and minutia directions to represent one minutiae vicinity. This feature vector f_i is denoted in Equation 8.2. Obviously, The number of feature vectors of a fingerprint sample depends on the number of minutiae in this sample.

$$f_i = (H_{a1}, ..., H_{a12}, J_{(a1,d)}, ..., J_{(a12,d)}) \tag{8.2}$$

8.2.2 Index space creation

The index space creation consists of two steps: first step is training stage which is to generate a classifier by applying unsupervised learning scheme $K - means$; second step is to build index space. In order to minimize the influence of minutia direction during classification, we propose to build four separate index tables based on minutiae direction. As illustrated in Figure 8.3, we divide the minutia direction range (0,360°) into four subdivisions:

- First subdivision: (0, 45°) and (315°, 360°);
- Second subdivision: (45°, 135°);
- Third subdivision: (135°, 225°);
- Forth subdivision: (225°, 315°);

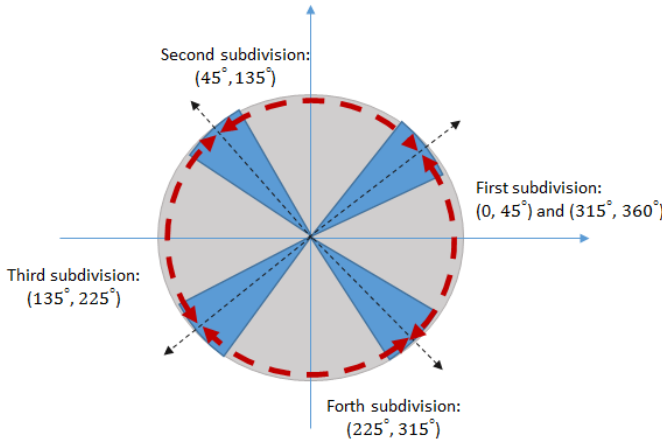


Figure 8.3: Minutiae will be distributed into four subsets according to their direction. The blue areas are over-lapping ranges, which means the minutia will be distributed into two subsets if its direction is located in the blue area.

Let's assume there are L feature vectors $(f_1, ..., f_L)$ extracted from the sample S. Since one minutiae vicinity produces one feature vector and one minutiae vicinity contains one

central minutia, we can link each feature vector f_l with a corresponding central minutia m_i. Afterwards, we can distribute feature vectors into four subsets according to the central minutia's direction. In addition, we consider an overlapping area as seen in Figure 8.3 where minutiae will be distributed into two subsets in order to be robust with sample rotation. We set the range of this overlapping area is $20°$, since fingerprint samples used in fingerprint identification system are generally captured using dedicated sensors which don't cause strong rotation. However, improving the robustness against strong rotation samples will be our future work. As the range of this overlapping are is $20°$, the overlapping area between first subdivision and second subdivision is ($35°$, $55°$). If a feature vector f_l has a central minutia with direction $50°$, this feature vector f_l will be distributed into both first subset and second subset. Then four index tables will be created by using four subsets respectively.

During training stage, training feature vectors extracted from training samples are distributed into four training subsets using aforementioned rules. Then we generate K clusters for each training subset by applying unsupervised learning scheme $K - means$. Each cluster can be represented by its centroid. $(Q(1,1), Q(1,2), ..., Q(1,K))$ denote K clusters generated from first training subset. Similarly, we have $(Q(2,1), Q(2,2), ..., Q(2,K))$, $(Q(3,1), ..., Q(3,K))$, $(Q(4,1), Q(4,2), ..., Q(4,K))$ for 2^{nd} subset, 3^{rd} subset and 4^{th} subset respectively.

The procedures of creating each index table are similar with the process in paper [172]. During the enrolment stage, a set of feature vector $(f_1, ..., f_L)$ can be extracted from a reference sample S associated with an unique ID A. Firstly, these feature vectors are assigned into four subsets: first subset $(f_{(1,1)}, ..., f_{(1,T)})$, second subset $(f_{(2,1)}, ..., f_{(2,U)})$, third subset $(f_{(3,1)}, ..., f_{(3,V)})$ and forth subset $(f_{(4,1)}, ..., f_{(4,W)})$. Secondly, each feature vector will be assigned into a closest cluster by calculating the Euclidean distance between this feature and the centroid of the cluster, then the sample ID will be recorded in this closest cluster. For instance, let's assume the closest cluster for feature vector $f_{(1,1)}$ is $Q(1,2)$, then ID A will be recorded in a bin $B(1,2)$. These procedures are repeated for every reference sample which will be enrolled in the index space. Finally, an index space including four index tables will be created and each index table is composed of K bins where each bin stores a list of fingerprint sample IDs.

8.2.3 Candidates retrieval

Figure 8.4 illustrates the procedures of candidates retrieval. When a probe sample presents in the fingerprint identification, a set of feature vectors will be extracted from the probe sample. Then these feature vectors will be distributed into four subsets by using the same rules described in the previous subsection. For instance, we assume there are H feature vectors in the first subset. Each feature vector from these H feature vectors will be assigned into a closest cluster in $(Q(1,1), Q(1,2), ..., Q(1,K))$. After that, we will locate G target clusters. Obviously, G is no more than H, since multiple feature vectors might be assigned into same cluster. Then we count the occurring frequency for each sample ID stored in these target clusters to output similarity scores. These procedures will be repeated in other three index tables. Afterwards, we fuse the similarity scores outputted from four index tables to output a final score. The candidate entries can be determined by selecting top-X largest similarity scores.

8.3 Performance evaluation

8.3.1 Performance metrics

In accordance with ISO/IEC FDIS 19795-1 [101], two metrics have been defined to evaluate the performance of biometrics indexing algorithm:

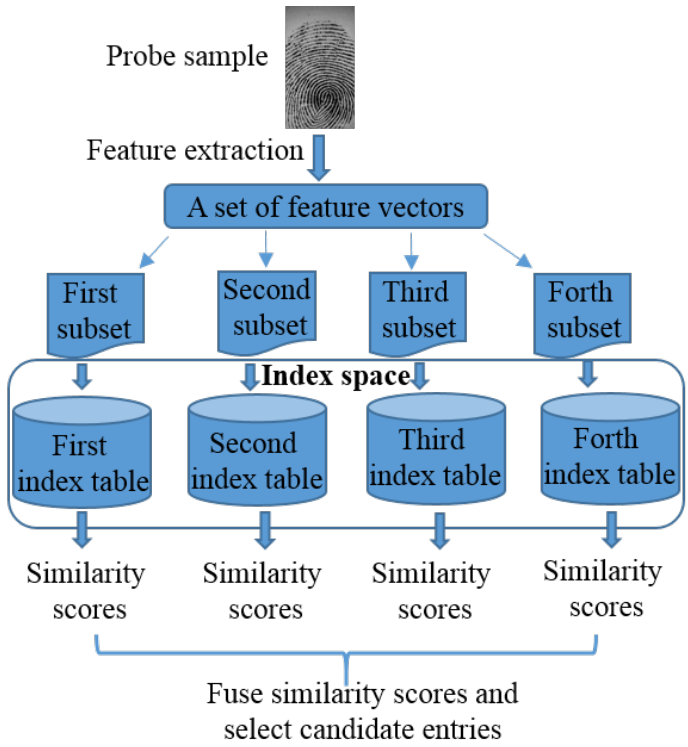


Figure 8.4: Procedures of candidates entries retrieval.

- Pre-selection error rate: a proportion of enrolled references where the enrolled reference corresponding to the query sample (or probe sample) is not involved in the pre-selected candidates which will be used for further comparing with the probe sample.
- Penetration rate: a proportion of enrolled references in a database where the system has to search.

The goal of a biometrics indexing algorithm is to achieve lower pre-selection rate at the same penetration rate. The performance of the proposed approach will be evaluated by using these two metrics comparing to minutia cylinder-code (MCC) indexing method.

8.3.2 Database preparation

The experiments were conducted on several public databases selected from $FVC2002$, $FVC2004$ and $FVC2006$. As indicated in $FVC2004$ website [13], the sample quality in $FVC2004$ is generally worse than the sample quality in $FVC2002$ due to the perturbations. In addition, there is no information scientifically talking about the sample quality in $FVC2006$ comparing to $FVC2004$. However, we can easily observer (by viewing original sample images) that the sample quality in $FVC2006\_DB2$ is worse than the sample quality in $FVC2004$. In general, it is understandable to say that the fingerprint sample quality is getting challenging from $FVC2002$ to $FVC2006$. In our experiments, $FVC2002\_DB2\_A$ was chosen as training set, and the rest of databases were used for testing sets. The first

sample of each subject was enrolled in index space, and the remaining of samples were used as probe samples.

8.3.3 Experimental results

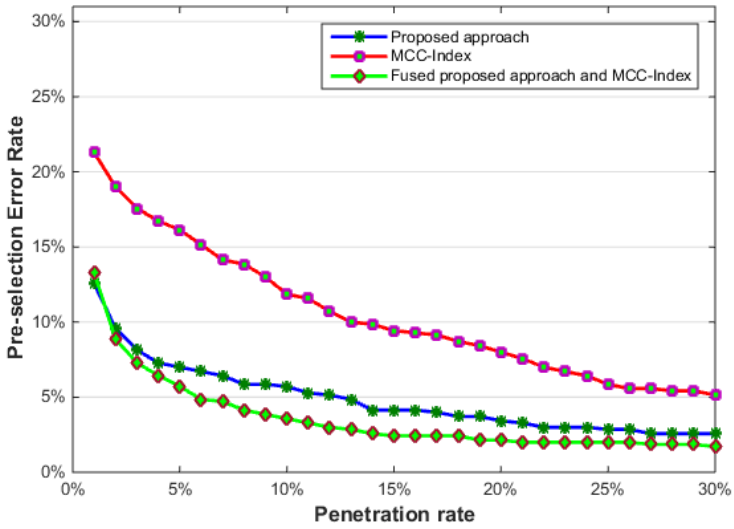


Figure 8.5: Penetration rate VS Pre-selection rate testing on $FVC2002\_DB1\_A$.

In our previous work [129], we have demonstrated that MCC-Index method outperforms than the fingerprint indexing approach proposed in [172], thus we choose MCC-Index method as the benchmark in this experiment. In addition, we also evaluate the performance by fusing proposed approach and MCC-Index at score level which was initially proposed in our previous work [129] as well. The idea of this fusion method is to build two index spaces using proposed approach and MCC-index respectively during enrolment stage, then fusing the similarity scores outputted from these two index spaces in retrieval stage. We use 'fused proposed approach and MCC-index' to indicate this fusion method in the following graphical results. The clusters' number (K) for each index table is 2000 in our experiments. Fingerprint template extractor used in this work is NeuroTechnology Verifinger extractor 6.0 [32]. The results of MCC-Index were obtained by MCC sdk v1.4 released by Cappelli et al. [61, 62, 79].

Figure 8.5 - Figure 8.8 demonstrate the results for testing on $FVC2002\_DB1\_A$, $FVC2004\_DB1\_A$, $FVC2004\_DB2\_A$ and $FVC2006\_DB2\_A$ respectively. We can observer that the fusion method always achieved best performance which is understandable. The proposed approach outperforms MCC-Index on $FVC2002\_DB1\_A$ and $FVC2004\_DB1\_A$, especially on $FVC2002\_DB1\_A$ where shows the significant improvement in terms of reducing pre-selection error rate. The proposed approach achieved better performance than MCC-Index on $FVC2004\_DB2\_A$ when the penetration rate is more than 6%. The performance of proposed approach is slightly worse than MCC-Index on $FVC2006\_DB2\_A$. One possible reason of proposed approach didn't perform well on $FVC2006\_DB2\_A$ is that the sample quality difference between training set ($FVC2002\_DB2\_A$) and testing set ($FVC2006\_DB2\_A$) is quite large. Overall, the proposed approach shows the equivalent performance or even better performance than MCC-Index method on these databases.

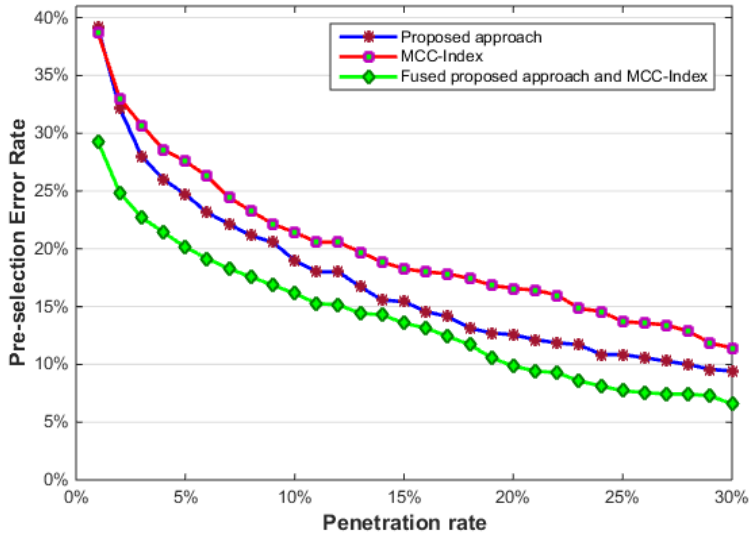


Figure 8.6: Penetration rate VS Pre-selection rate testing on $FVC2004\_DB1\_A$.

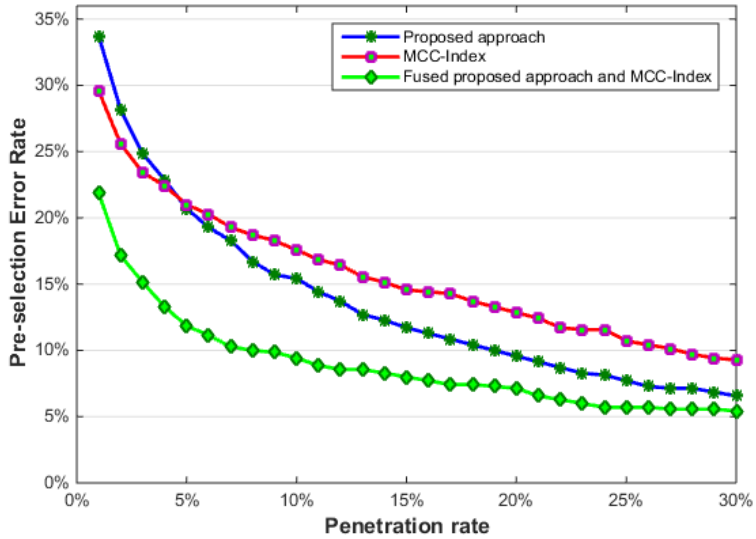


Figure 8.7: Penetration rate VS Pre-selection rate testing on $FVC2004\_DB2\_A$.

8.4 Conclusion

In this paper, a novel fingerprint indexing approach only using minutia location and direction is presented. The proposed approach develops a 24-D feature vector to represent a minutiae vicinity, and uses these feature vectors to build an index space including 4 sep-

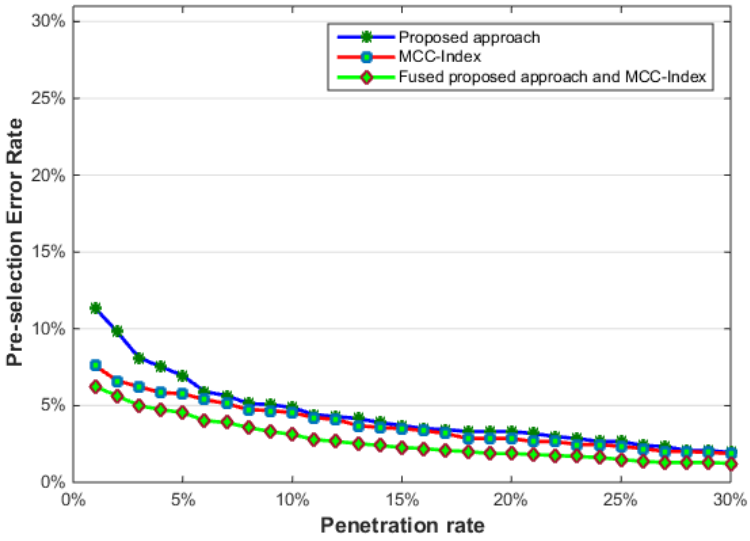


Figure 8.8: Penetration rate VS Pre-selection rate testing on $FVC2006\_DB2\_A$.

arate index tables according to the direction of the central minutia in a minutiae vicinity. During retrieval stage, a list of candidates will be obtained by selecting those identities associated in top-X largest similarity scores which are fused by four scores generated from four index tables. The experiments demonstrate that the proposed approach achieved equivalent performance or even better performance than MCC-Index method on tested databases. Our future work will study minimizing the impact of sample quality difference between training set and testing set, as well as improving the robustness against strong rotation samples.

Chapter 9

A Fingerprint Indexing Scheme with Robustness against Sample Translation and Rotation

Abstract

Automatic fingerprint identification systems (AFIS) are getting prevalent around the world, and the size of fingerprint databases involved in AFIS is continuously growing. Thus, studying fingerprint indexing algorithms is desirable in order to facilitate the search process in a large-scale database. In this paper, we firstly propose a feature extraction method to generate a binary template based on minutia information. A fingerprint indexing is designed by combining this binary template and Locality Sensitive Hashing indexing algorithm developed in a state-of-the-art fingerprint indexing method (minutia cylinder-code based indexing method). Experiments have been conducted on several public databases with different settings. The results show that the proposed approach achieves competitive performance or even better performance when benchmarked to the state-of-the-art fingerprint indexing method.

9.1 Introduction

Fingerprint recognition system has been increasingly gaining attention around the world. Many systems have been deployed such as FBI's Integrated Automated Fingerprint Identification System (IAFIS), the European Visa Information System (VIS) and many other systems are under construction. Depending on the application context, there are two types of fingerprint recognition systems [143].

Verification system: this system carries out a one-to-one comparison to verify whether the identity is the person who he/she claims. A typical scenario is to compare the fingerprint data stored in a Europen passport with the fingerprint data captured from the subject, who holds the passport.

Identification system: this system identifies a person by searching the whole database, which results in a one-to-N comparisons process. A typical scenario is to check whether a criminal suspect has been recorded in FBI's IAFIS by using his/her probe fingerprint samples for the query.

According to the information published on FBI's website [7], the FBI IAFIS contains enrolled fingerprint from more than 100 million subjects. Thus it is almost impossible to conduct a one-by-N comparisons in such a large-scale database. Therefore, studying fingerprint indexing techniques is desirable in order to reduce the number of candidate identities, which will be further considered by a verification algorithm [138].

A variety of fingerprint indexing approaches have been presented in the literature. Extracting appropriate features for building indexing tables is the core of a fingerprint indexing approach. The approaches in the literature can be grouped into three categories based on their feature extraction methods: local feature based – primarily focusing on using minutia [172, 62] or local ridge information [138, 51]; global feature based – using orientation field and singular points as a global reference points [201]; other feature based – such as using symmetric filters [136] or scale invariant feature transformation (SIFT) [182].

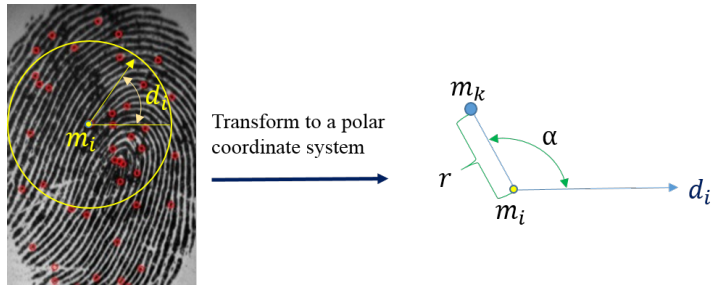


Figure 9.1: Local alignment: all minutiae included in the yellow circle are aligned with the central minutia in a polar coordinate system whose reference point is m_i and reference angle is the direction (denoted by d_i) of m_i. This central minutia and another minutiae included in the circle are named as a minutiae-disk.

Instead of exploring global features, we are focusing on only using minutia location and direction to extract a compact feature vector, since minutia information has been recognized as most reliable and basic feature representing fingerprints, and a standardized definition of this feature is given by ISO/IEC 19794-2:2011 [99]. In addition, the majority of existing fingerprint indexing approaches generate the features with real values, which might lead to more computational complexity comparing to binary features. Thus we explore to extract a binary feature vector in this paper, and further build the indexing tables by using Locality Sensitive Hashing (LSH) indexing method which was developed by Cappelli et al. [62, 63] and has been proven to be suitable for binary feature vectors.

The remainder of this paper is structured as follows: Section 9.2 introduces the proposed feature extraction method; the details of creating indexing tables and candidates retrieval are described in Section 9.3; Section 9.4 reports the experimental results under different settings; the conclusion is drawn in Section 9.5.

9.2 Feature extraction method for fingerprint indexing

The feature extraction method is the critical component in a fingerprint indexing scheme due to the fingerprint sample variations caused at acquisition stage. The proposed feature generation method generates a set of fixed-length binary vectors for a fingerprint template. The number of binary vectors for a fingerprint template depends on the minutiae' number in this template. The proposed feature extraction method consists of three stages: local alignment, training and binary vectors generation. The details will be discussed in the following subsections.

9.2.1 Local alignment and quantization

Instead of detecting a singular point or considering the ridge information surrounding the minutia, we focus on using a local alignment concept to extract a binary vector, which can represent this local area. The basic idea of the local alignment is considering each minutia as a reference point, and then nearby minutiae are aligned with respect to this reference point (called central minutia). As illustrated in Figure 9.1, all minutiae included in the yellow circle are aligned with the central minutia in a polar coordinate system. This central minutia and another minutiae included in the yellow circle are defined as a minutiae-disk.

We assume a fingerprint template T including n minutiae $\{m_1, m_2, ..., m_n\}$, and each minutia comprises three properties: $m_i(x, y, d)$, where x and y are the minutia location and d is the minutia direction. A minutiae-disk (MD_i) can be formed for each minutia

m_i. A polar coordinate system is defined by using m_i as reference point and the direction (denoted by d_i) of m_i as reference angle. Then each minutia m_k included in the minutiae-disk will have a new coordinate $m_k^{'}(r^{'}, \alpha^{'})$ denoted in Equation 9.1 and 9.2.

$$r^{'} = DIS(m_k, m_i) \tag{9.1}$$

where DIS is Euclidean distance between the two minutiae.

$$\alpha^{'} = \frac{(atan2(m_k(y) - m_i(y), m_k(x) - m_i(x)) + 2\pi) * 180}{\pi} \tag{9.2}$$

where function $atan2$ is 'Four-quadrant inverse tangent' defined in [11].

In addition, the minutiae angle difference $\theta^{'}$ between m_i and m_k is denoted by the following equation:

$$\theta^{'} = |m_k(d) - m_i(d)| \tag{9.3}$$

In order to further tolerate the sample variation, three attributes $(r^{'}, \alpha^{'}, \theta^{'})$ are quantified by using Equation 9.4∼9.6.

$$r = floor(r^{'}/5) \tag{9.4}$$

$$\alpha = floor(\alpha^{'}/5) \tag{9.5}$$

$$\theta = floor(\theta^{'}/5) \tag{9.6}$$

where, function $floor(X)$ returns the nearest integer less than the variable X.

Eventually, an aligned minutia $m_k^{'}$ with three attributes (r, α, θ) is created. Since the proposed feature generation method applies local alignment for each minutia, this indicates that each minutiae-disk will be associated with a minutia which is called central minutia. The radius of the minutiae-disk is denoted as R.

9.2.2 Training and binary vectors generation

A training step is required in the proposed feature extraction method prior to the binary vector generation. The unsupervised learning scheme $K - means$ is chosen for this training step, since it has been proven to be appropriate for fingerprint indexing by other researchers [172, 129]. The input of $K - means$ is a set of (r, α, θ) vectors generated from the training samples. $K - means$ classifies these (r, a, θ) vectors into K clusters, and each cluster is represented by its centroid $\{C_1(r, \alpha, \theta), C_2(r, \alpha, \theta), ..., C_K(r, \alpha, \theta)\}$.

The proposed feature extraction method generates a fixed-length binary vector for each minutiae disk. Since one minutia will form one minutiae disk, the number of binary vectors for a template is equal to the number of minutiae in this template. Figure 9.2 illustrates the procedures of generating the binary vector for a minutiae-disk. There are four steps involved in this process:

Step 1: Apply local alignment on this minutiae-disk to generate alignment minutiae: $m_1^{'}(r, \alpha, \theta), m_2^{'}(r, \alpha, \theta), ..., m_J^{'}(r, \alpha, \theta)$.

Step 2: Initiate a K bits binary vector with all components set to 0.

Step 3: Assign each $m_k^{'}(r, \alpha, \theta), k \in (1, 2, ..., J)$ to nearest three clusters (closest cluster, second closest cluster and third closest cluster) according to their Euclidean distances.

Step 4: Flip 0 to 1 if there is a $m_k^{'}$ assigned to the corresponding cluster. Eventually, a binary vector is generated to represent this minutia-disk.

Note that only the first change will take effect even if multiple $m_k^{'}$ have been assigned to the same cluster. The reason of choosing nearest three clusters is to tolerate sample intra-class variations.

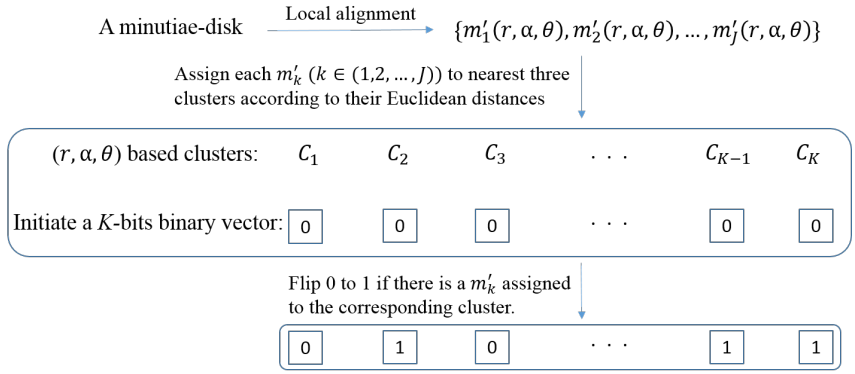


Figure 9.2: Procedures of generating the binary vector for a minutia-disk.

9.3 The indexing algorithm

Cappelli et al. [62] have proofed that Locality Sensitive Hashing (LSH) is suitable to index the binary vector. We follow their techniques proposed in paper [62] to build the indexing tables and retrieve the candidates by using our newly generated binary vectors. The following subsections give the details of indexing tables creation and candidates retrieval.

9.3.1 Creating indexing tables

Algorithm 9.1 Indexing tables creation

Require: Minutiae templates of enrolled subjects: $\{T_1, T_2, ..., T_E\}$;
Hash functions:$\{f_{H_1}, f_{H_2}, ..., f_{H_\Lambda}\}$ (Λ is the number of hash functions);
Ensure: Indexing tables: $H_1, H_2, ..., H_\Lambda$

```
for each template T_i (i ∈ 1,2,...,E) do
    Generate binary template from minutia template by using proposed feature extraction method: {T(i,1), T(i,2), ..., T(i,J)}(J is the number of binary vector generated from minutiae temple T_i)
    for each binary vector T(i,j)(j ∈ 1,2,...,J) do
        for each hash function f_{H_λ} do
            b = f_{H_λ}(T(i,j)))
            if CountOneBits(b) ≥ min_{OneBits} then
                record (i,j) in b − th bucket of indexing table H_λ.
            end if
        end for
    end for
end for
```

Before describing the algorithm of creating LSH-based indexing tables for fingerprint templates, it is necessary to introduce the techniques of LSH indexing method. Figure 9.4 gives an example of creating indexing tables by using a set of hash functions $\{f_{H_1}, f_{H_2}, ..., f_{H_\Lambda}\}$, where the number of hash functions is $\Lambda = 3$, and the number of bits selected by each hash function is $\eta = 3$. Let assume there is a binary (T_1, V_1) which denotes the first binary vector of the first fingerprint template. Each hash function will randomly select 3 bits from (T_1, V_1), then calculate the decimal value based on selected bits and store the pair (1,1) in a

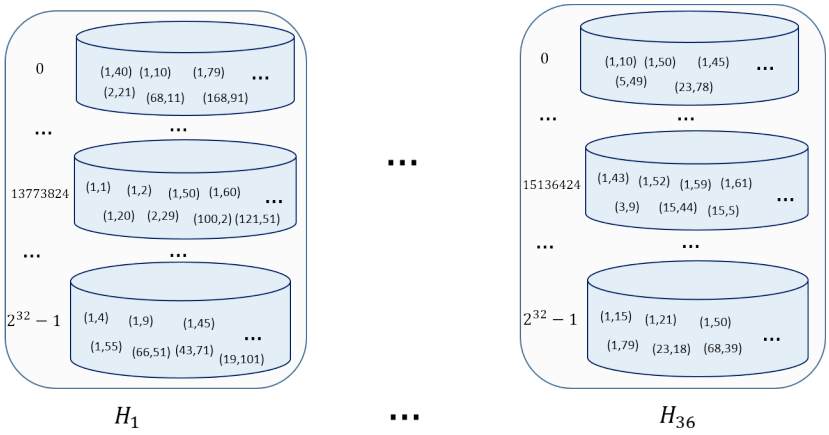


Figure 9.3: An example of created indexing tables, where the pair (i, j) indicates the $j-th$ binary vector of $i-th$ fingerprint template.

corresponding bucket. For instance, the decimal value calculated from f_{H_2} is 3, then (1,1) will be stored in the third bucket in hash table H_2. The number of hash tables is equal to the number of hash functions, and the number of buckets in each hash table is 2^{η}.

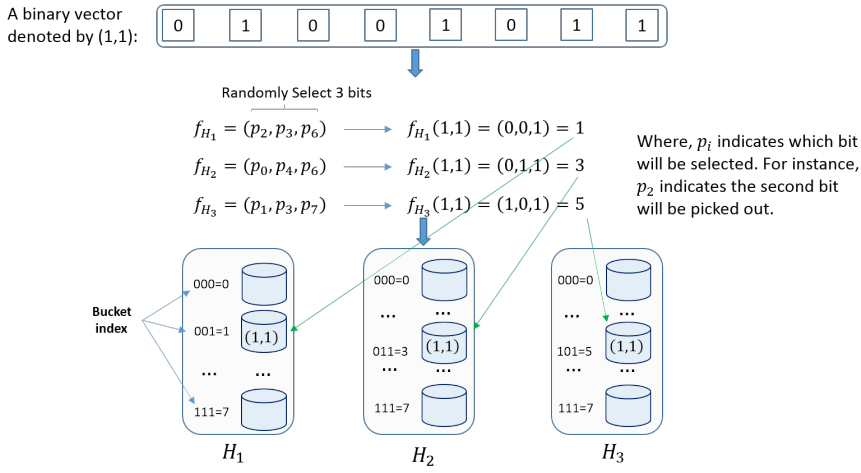


Figure 9.4: An example of Locality Sensitive Hashing (LSH) indexing algorithm.

During indexing tables creation stage, we apply the similar procedures illustrated in Figure 9.4 to enrol fingerprint minutia template. **Algorithm** 9.1 gives the details of creating indexing tables for a set of fingerprint minutiae templates: $\{T_1, T_2, ..., T_E\}$. The first step of enrolling these minutia template is to generate the binary template by using proposed feature extraction method. The function $CountOneBits(b)$ is to count the number of 1 bits in selected bits, for instance $CountOneBits(1010001) = 3$. The pair (i, j) will be recorded only when $CountOneBits(b)$ is not less than a parameter $min_{OneBits}$. Figure 9.3 gives

an example of the indexing tables after completing enrolment. In addition, the original minutiae templates need to be stored somewhere else (minutiae template can be indexed by their template ID), since they will be used during candidate retrieval stage.

9.3.2 Candidates retrieval

Algorithm 9.2 lists the procedures of retrieving candidates for a probe sample P. The same hash functions used in enrolment are used as input for candidates retrieval. Another inputs are: indexing tables $\{H_1, H_2, ..., H_\Lambda\}$, enrolled minutiae templates $\{T_1, T_2, ..., T_E\}$ as well as the minutia template of the probe sample P. The function '$Mated(m_\omega, m(i,j))$' is to measure whether minutia m_ω from probe sample and minutia $m(i,j)$ from the reference sample meet a pre-defined geometric constraint. If they satisfy the geometric constraint, the similarity score between this probe sample and the reference sample will increases 1. '$Mated(m_\omega, m(i,j))$' is defined in Equation 9.7. In order to reduce the computational complexity, we don't normalize the similarity score which is different to the candidates retrieval method used in paper [62].

$$Mated(m_\omega, m(i,j)) = \begin{cases} true & \text{if } DIS((m_\omega, m(i,j)) \leq \rho \text{ and } |m_\omega(d) - m_{(i,j)}(d)| \leq \sigma \\ false & \text{otherwise.} \end{cases} \tag{9.7}$$

where,$|m_\omega(d) - m_{(i,j)}(d)|$ is the direction difference between two minutiae.

Algorithm 9.2 Candidates retrieval

Require: Indexing tables: $H_1, H_2, ..., H_\Lambda$;
Hash functions:$\{f_{H_1}, f_{H_2}, ..., f_{H_\Lambda}\}$;
Minutiae template of enrolled subjects: $\{T_1, T_2, ..., T_E\}$;
Minutiae template of probe sample: P.
Ensure: Candidate entities.
1: Generate the binary template for probe sample: $V_1, V_2, ..., V_\Omega$ (Ω is the number of binary vectors);
2: Initiate an array to store similarity score: $S[E]$;
3: **for** each binary vector V_ω **do**
4: Assume m_ω is the central minutia associated with binary vector V_ω
5: **for** each hash function f_{H_λ} **do**
6: $b = f_{H_\lambda}(V_\omega)$
7: **if** $CountOneBits(b) \geq min_{OneBits}$ **then**
8: **for** each pair (i,j) in $b-th$ bucket of indexing table H_λ **do**
9: Assume $m(i,j)$ is the central minutia associated with the pair (i,j);
10: **if** $Mated(m_\omega, m(i,j))$ == true **then**
11: $S[i] = S[i] + 1$;
12: **end if**
13: **end for**
14: **end if**
15: **end for**
16: **end for**
17: Sort $S[E]$ by descending order, and select the top-N as candidate entities.

9.4 Experimental settings and results

In order to evaluate the performance of proposed indexing approach, a couple of experiments have been conducted on several public databases. In accordance with ISO/IEC

Table 9.1: Parameters setting for all experiments.

Parameter	value	Remark
R	300 pixels	the radius of the minutiae-disk
K	1024	the length of binary vector
Λ	48	the number of hash functions
η	32	the number of bits selected by hash function
ρ	256	minutia distance threshold
σ	45	minutia direction difference threshold
$min_{OneBits}$	2	the number of '1' bits in a binary vector

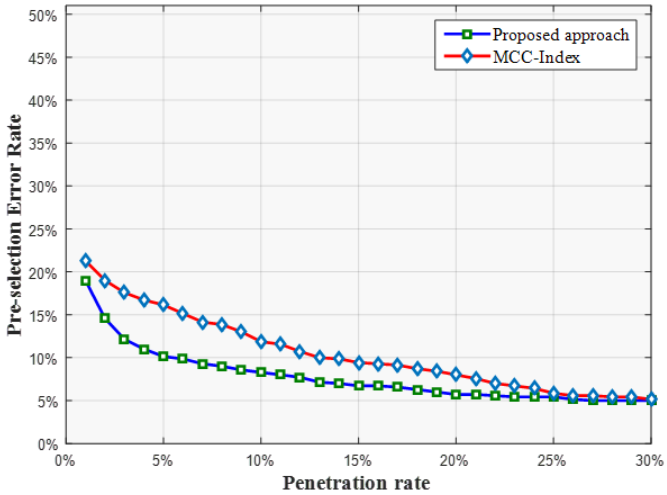


Figure 9.5: Experiment on $FVC2002\_DB1\_A$.

19795-1 [101], the performance of fingerprint indexing algorithm is reported by two criteria: penetration rate and pre-selection error rate. Penetration rate is a proportion of enrolled references in a database where the identification system has to search. A pre-selection error occurs when the enrolled reference corresponding to the probe sample is not included in the pre-selected candidates. Generally speaking, the better fingerprint indexing approach will achieve lower pre-selection error rate at the same penetration rate comparing to other approaches. The minutia cylinder-code based indexing method (shortly called MCC-Index) [62] was used as benchmark under same protocol in our experiments.

9.4.1 Databases preparation and common settings for all experiments

Several FVC databases are selected for the experiments: $FVC2002$ [143], $FVC2004$ and $FVC2006$ [60]. The details for the respective database will be described in the following sections. The minutia templates were extracted by a commercial product 'NeuroTechnology Verifinger extractor 6.0' [32]. The experimental results of MCC-Indexing method were

generated by MCC sdk v1.4 [61, 62, 79]. And Table 9.1 lists the settings of some parameters used for all experiments.

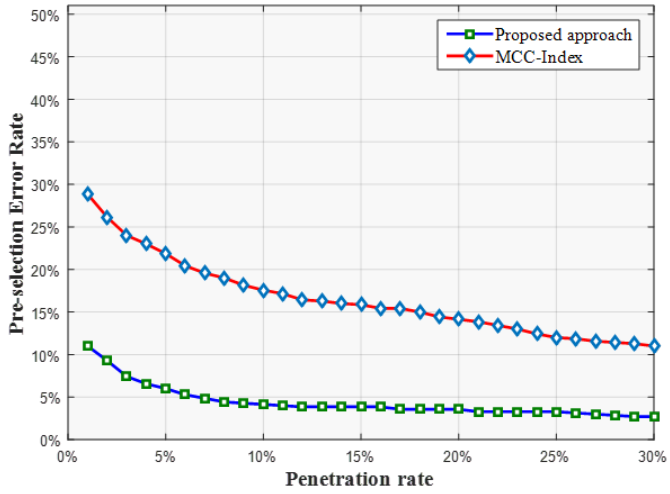


Figure 9.6: Experiment on $FVC2002\_DB2\_A$.

9.4.2 Experiments on $FVC2002$

We run the experiments on $FVC2002\_DB1$ and $FVC2002\_DB2$ respectively. There are two subsets in $FVC2002\_DB1$ as well as in $FVC2002\_DB2$. We used $FVC2002\_DB1\_B$ (consisting of 80 samples) as a training set for and $FVC2002\_DB1\_A$ as a test set which comprises 800 sample from 100 fingers. The first sample of each finger was enrolled in the indexing tables and the rest of samples were used for searching, since the quality of first sample is relatively better than the rest of samples in $FVC2002$. The similar settings were applied on $FVC2002\_DB2$: $FVC2002\_DB2\_B$ was used for the training set; $FVC2002\_DB2\_A$ was used for the test set; the first sample was used for enrolment, and the rest of samples were used for probe samples. Figure 9.5 and Figure 9.6 demonstrate the performance running experiment on $FVC2002\_DB1$ and $FVC2002\_DB2$. The figures show the significant improvements of proposed approach on these databases.

9.4.3 Experiments on $FVC2004$

In order to establish similar settings as for the experiments on $FVC2002$, we used the $FVC2004\_DB1\_B$ as a training set for the test set $FVC2004\_DB1\_A$, and used the $FVC2004\_DB2\_B$ as a training set for the test set $FVC2004\_DB2\_A$. Again we enrolled the first sample of each finger to the indexing tables as we did for $FVC2002$. Figure 9.7 and 9.8 show that the MCC-Index method outperformed our proposed approach. Then we observed the sample images of $FVC2004$. We found that the first sample of each finger doesn't have higher quality, and even it can be deemed as partial fingerprint comparing other sample from the same finger as seen in Figure 9.9. This 'partial sample' trait might have more impact on the proposed approach than MCC-Index method, since the radius of minutiae-disk is 300 pixels in proposed approach and MCC-Index method used 70 pixels.

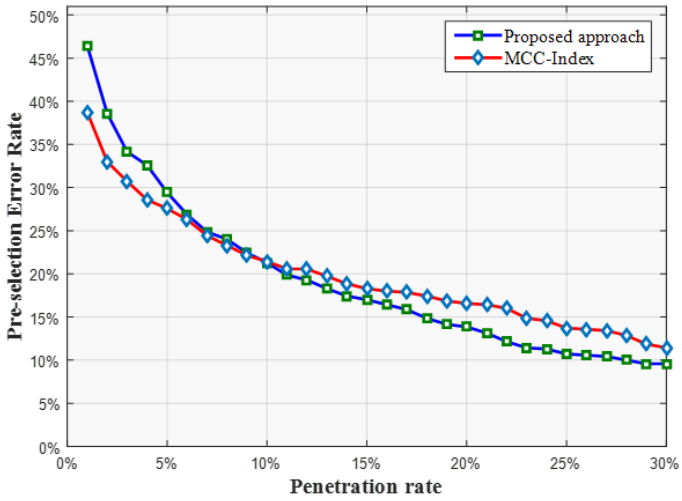


Figure 9.7: Performance evaluation on $FVC2004\_DB1\_A$: the **first sample** of each subject was enrolled in indexing tables.

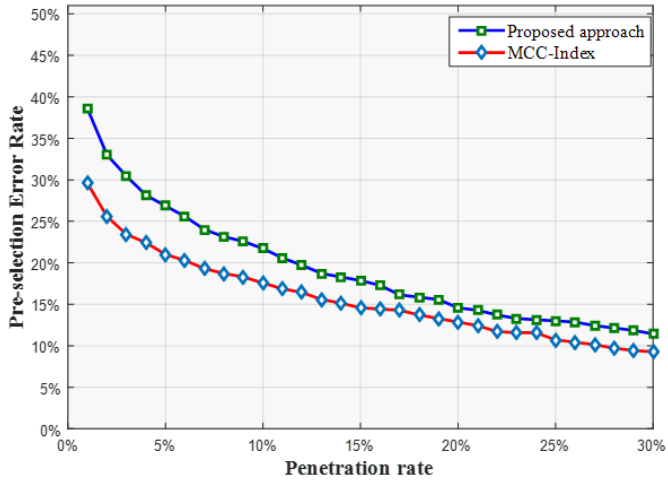


Figure 9.8: Performance evaluation on $FVC2004\_DB2\_A$: the **first sample** of each subject was enrolled in indexing tables.

In order to investigate the impact of 'partial sample', we enrolled the forth sample of each finger and used the rest of sample as probes. Figure 9.10 and Figure 9.11 depict the results of using forth sample as enrolled template. The performance of proposed approach are both improved, especially on $FVC2004\_DB1\_A$.

Figure 9.9: Fingerprint samples selected from $FVC2004\_DB1\_A$.

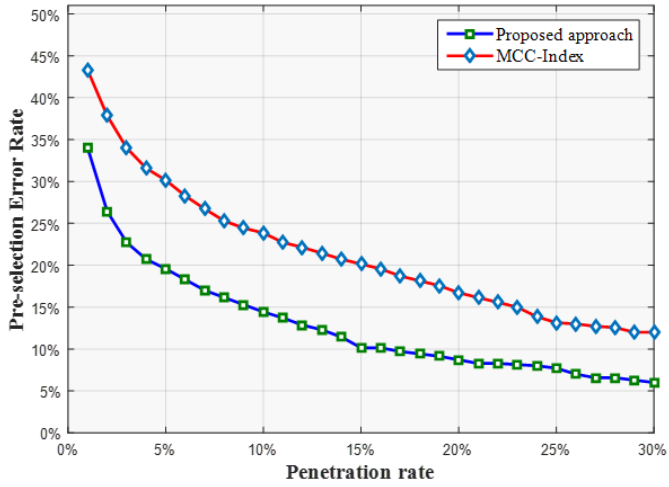


Figure 9.10: Performance evaluation on $FVC2004\_DB1\_A$: the **forth sample** of each subject was enrolled in indexing tables.

9.4.4 Experiments on $FVC2006$

FVC_2006_DB2 was selected to evaluate the performance. The training set is $FVC\_2006\_DB2\_B$ which consists of 120 samples, and the test set is $FVC\_2006\_DB2\_A$ consisting of 1680 samples which were captured from 140 fingers (12 sample per finger). The first sample of each finger was used for enrolment. Another 11 samples were chosen as probe samples for searching. In total, there are 1540 probe samples. Figure 9.12 shows the improvement of the proposed approach. The improvement is relatively low, since the performance of MCC-Index method is already a good baseline.

9.5 Conclusion

In this paper, a fingerprint indexing algorithm is designed by only using minutia location and direction information. It is invariant to sample translation and rotation, since the proposed approach applies the local alignment on each minutia to generate a binary vector rather than using a global reference point. Based on the binary vectors for the template, an indexing approach is designed by combining LSH indexing algorithm developed in MCC-Index method. The experiments on several public database have demonstrated that the

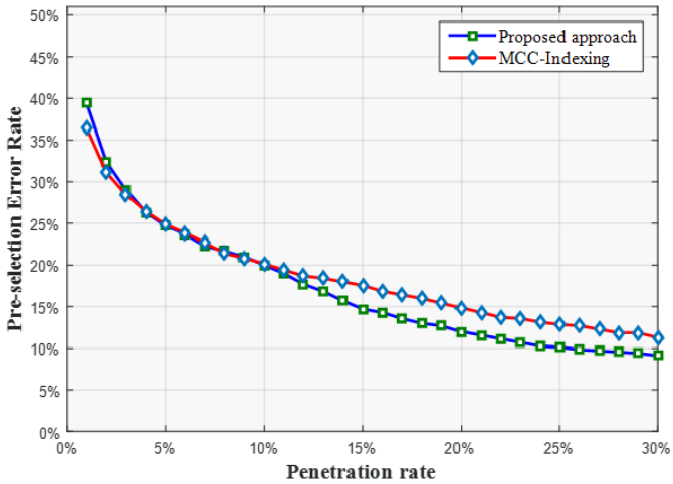


Figure 9.11: Performance evaluation on $FVC2004\_DB2\_A$: the **forth sample** of each subject was enrolled in indexing tables.

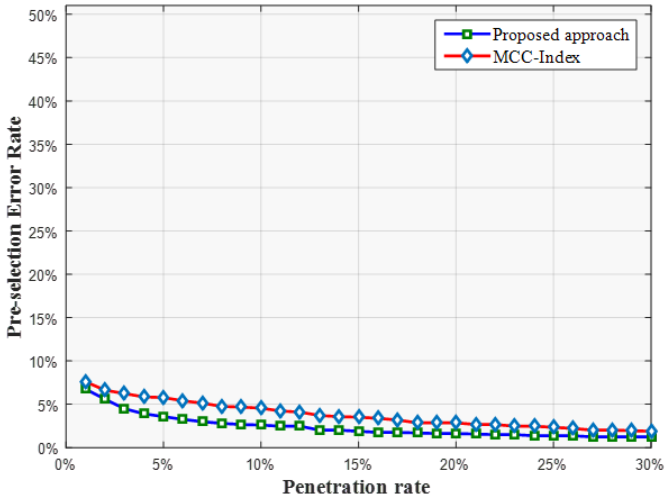
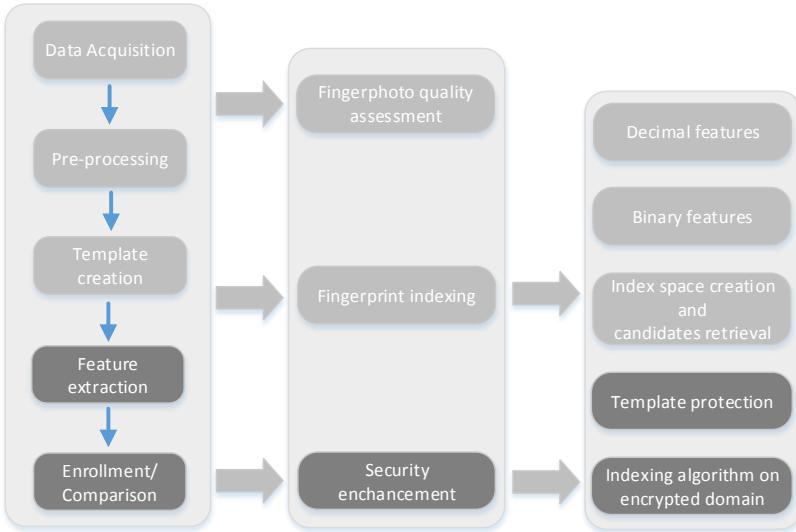


Figure 9.12: Performance evaluation on $FVC2006\_DB2\_A$.

proposed approach achieved comparative performance or even better performance than the state-of-the-art fingerprint indexing method. Our future work will extend the experiment to larger-sized databases as well as investigate the impact of the radius of minutiae-disk in order to make the proposed approach more robust to the partial fingerprint sample.

Part IV

Security Enhancement



This part addresses the security issues in a fingerprint identification system in order to answer the research question RQ_4: **How to embed the privacy-preserving capability for the large-scale fingerprint identification system while still keeping the performance?**

In order to achieve the privacy-preserving capability, we developed two approaches to protect the user's biometric data.

In Chapter 10, we proposed a fingerprint template protection approach based on Bloom filters which has been successfully applied to protect face and iris templates. Before applying Bloom filters, we also designed a pre-alignment module and adapted the binary template generation scheme developed in [215], since Bloom filters requires a binary template as input.

In Chapter 11, we proposed a secure fingerprint indexing approach. This proposed approach is based on the fingerprint indexing approach developed in Chapter 9 by adding an encryption module before the binary template generation module, which implies that all information stored in the index space are encrypted. Meanwhile, the proposed encryption module adopts a standard encryption algorithm for protecting the biometric data. This feature indicates that the security of proposed approach is guaranteed by a standard encryption algorithm. According to the experiments conducted on several public datasets and a large-scale synthetic dataset, the proposed secure fingerprint indexing approach still maintains a very good performance compared to the fingerprint indexing approach without considering security mechanism in Chapter 9.

The work in Chapter 10 was published in [134] GUOQIANG LI, BIAN YANG, RATHGEB CHRISTIAN AND CHRISTOPH BUSCH, 2015, March. Towards generating protected fingerprint templates based on bloom filters. In Biometrics and Forensics (IWBF), 2015 International Workshop on (pp. 1-6). IEEE.

The work in Chapter 11 has been accepted by 'The 15th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (IEEE TrustCom-16)'.

Chapter 10

Towards Generating Protected Fingerprint Templates based on Bloom filters

Abstract

In order to satisfy the requirements for security and privacy of biometric enrolment data records, it is essential to protect the reference data by applying appropriate template protection schemes. Bloom filters have been applied successfully on iris biometrics and face biometrics and achieved good result in terms of irreversibility and biometric performance. In this paper we study, whether it is feasible to employ Bloom filters on fingerprint templates. In order to be resilient with fingerprint sample variations, a pre-alignment process is applied prior to binary template generation. After generating the binary template matrix, we propose to subdivide the matrix and achieve a variable size of the binary template. Experiments were conducted on public databases to confirm the proposed ideas. According to experimental results, applying Bloom filters on fingerprint template doesn't degrade the accuracy of the fingerprint recognition system. Therefore, we can conclude that it is feasible to apply Bloom filters on fingerprint biometrics.

10.1 Introduction

Fingerprint recognition has been widely adopted to authentication systems in order to verify the identity claim of an individual. From the security and privacy perspective, securing the fingerprint reference data is essential because of the permanence properties of the biometric fingerprint characteristic. Unlike conventional passwords, which can be re-enrolled using a new password after leakage [196] this more challenging for biometric reference data. In addition, it has been proven that the original fingerprint information and potentially sensitive medical information can be reconstructed from a fingerprint template [60, 78]. Therefore, studying biometric template protection schemes has received increasing attention in the biometric community. In accordance with the international standard ISO/IEC 24745 [103], a biometric template protection method need to meet two major requirements:

- Irreversibility: it should be infeasible to reconstruct the original biometric template from the protected template;
- Unlinkability: different versions of protected templates can be generated from the same sample, but should not allow cross-matching.

A variety of biometric template protection schemes have been proposed in literature. These approaches can be roughly classified into two categories: biometric cryptosystem and cancelable biometrics (also refers to feature transformation) [171]. The idea of biometric cryptosystem is to protect or retrieve the cryptographic key by using biometric data. The comparison process is operated by verifying the hash result of extracted key against stored hash data. There are two types of fingerprint cryptosystems, which are based on fuzzy vault [156, 114] and fuzzy commitment [191] respectively. The majority of these approaches require some public information (called the helper data) to properly align fingerprint samples, which is critical and challenging to achieve.

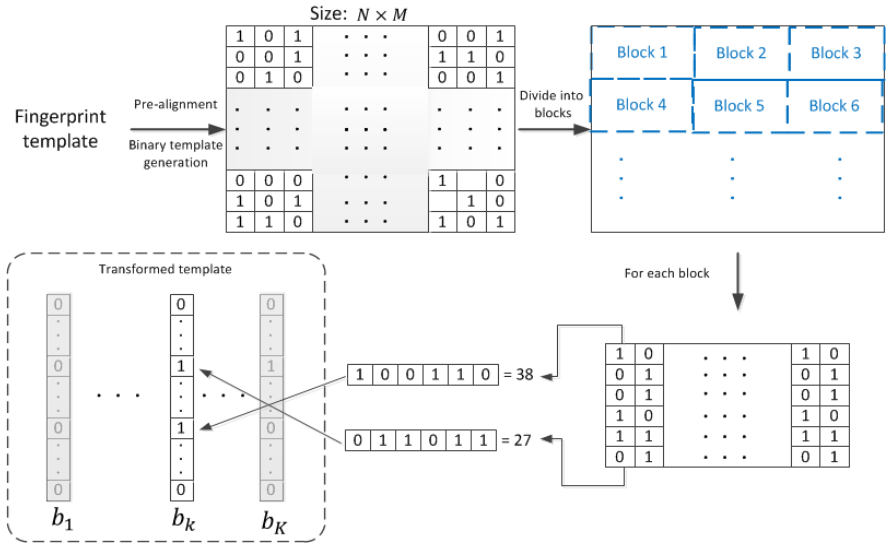


Figure 10.1: The process of transformed template generation by applying Bloom filters on fingerprint template

Ratha et al .[169] promoted the concept of cancelable biometrics, which can meet the two requirements of irreversibility and unlinkability. The idea of cancelable biometrics is to generate as many protected template (or called transformed template) as needed by issuing a new transformation key, and the comparison process can be operated on transformed templates. Researchers [168, 122, 67] have employed this concept to generate cancelable fingerprint templates. However, these approaches caused a significant degradation in biometric performance. Another feature transformation approach based on minutia cylinder-code representation [79] achieved good performance, but it doesn't guarantee the unlinkability.

Bloom filters has been introduced in the field of research, deriving an iris template protection scheme by Rathgeb et al. [69, 84]. The irreversibility can be guaranteed by mapping multiple codewords to an identical position, and unlinkability based on application-specific secret are current research topics with promising results [69]. Since applying Bloom filters on iris templates and also on face templates is feasible, it inspired us to investigate the application of Bloom filters on fingerprint templates. Comparing to iris template whose size is fixed, the size of a fingerprint template is generally variable and large. This presents a challenge to apply Bloom filters on fingerprint template. In this paper, we addressed this challenge and explore introducing the concept of Bloom filters on fingerprint templates. The remainder of this paper is organized as follows: Section 10.2 describes the details of pre-alignment, binary template generation and the mapping to Bloom filters; the experimental results of performance evaluation are reported in Section 10.3. Section 10.4 discusses future works and concludes this paper.

10.2 Applying Bloom filters to fingerprint

As we mentioned earlier, the purpose of cancelable biometrics is to transform the fingerprint template into a protected domain where the matching process can take place. Figure 10.1 illustrates the process of generating this transformed template by applying Bloom filters on fingerprint template. The first step of proposed approach is a fingerprint pre-

alignment module where only minutiae located in a circle will be used for the binary template generation as shown in Figure 10.2. The reason for adding this pre-alignment module is that the minutiae included in the circle are more robust and reliable than the minutiae closed to border during the fingerprint sample acquisition. The circle's radius r is adjustable according to the resolution of fingerprint sample. The centre point (C_x, C_y) of this circle is the reference point of each sample image. This reference point can be efficiently detected by using a simple rule:

- (1) if only one core point is detected by fingerprint template extractor (we chose NeuroTechnology Verifinger 6.0 extractor [32]), this core point will be considered as reference point;
- (2) if multiple core points are extracted, the uppermost core point will be chosen as reference point;
- (3) if the extractor doesn't detected any core point, then the reference point will be calculated using equation 10.1-10.2.

$$C_x = \min(m_{(i,x)}) + \frac{\max(m_{(i,x)}) - \min(m_{(i,x)})}{2} \tag{10.1}$$

$$C_y = \min(m_{(i,y)}) + \frac{\max(m_{(i,y)}) - \min(m_{(i,y)})}{2} \tag{10.2}$$

where $m_{(i,x)}$ is the X coordinate of minutia m_i and $m_{(i,y)}$ is the Y coordinate of minutia m_i.

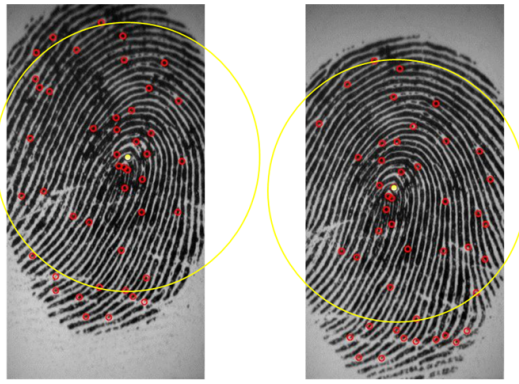


Figure 10.2: Pre-alignment: only minutiae (marked by red circle) which are located in the yellow circle will be used for binary template generation

After fingerprint alignment, the proposed approach adapted the binary generation scheme developed by Yang [215]. Figure 10.3 depicts the procedures of this scheme which will output a binary template with size $N \times M$, where N is a fixed value for all samples and M relies on the number of minutiae in each sample. The binary template is composed by the $N-$dimensional binary vectors generated from each minutiae vicinity. A minutiae vicinity is a basic unit which is formed by four minutiae including a center minutia and its three closest neighboring minutiae sorted by ascending order based on their Euclidean distance with the center minutia [213]. Each minutiae vicinity contains 6 orientations which are defined between minutiae pairs as seen in Figure 10.3. If we use each orientation as X axis in a new coordinate system, the remaining minutia pair can be geometrically-aligned. For instance in Figure 10.3, a new aligned minutiae pair J_{a1}, J_{a2} can be obtained if we

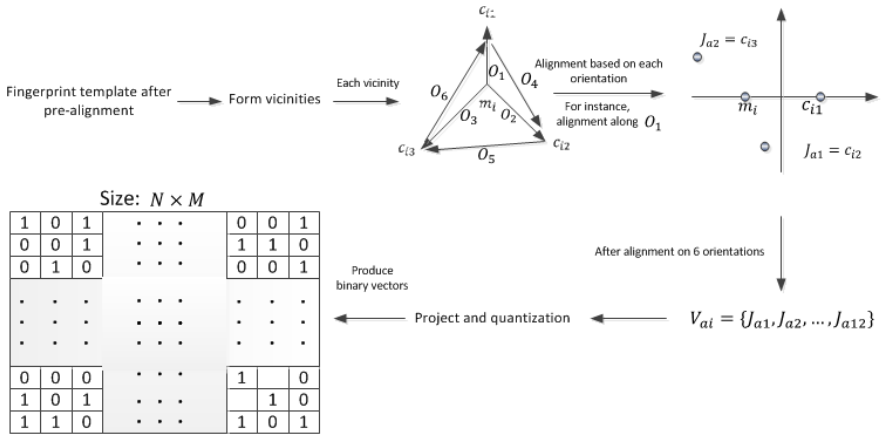


Figure 10.3: Procedures of binary template generation

use the orientation O_1 as a new coordinate system. Thus 6 new minutia pairs can be obtained after this geometric alignment. A 36-dimensional vector V can be composed by concatenating the coordinates x, y and angle information from these 12 new minutiae. This 36-dimensional vector will be used as input for projection and quantization which is performed by the Equation 10.3.

$$t = Q(R^T V) \tag{10.3}$$

where R^T consists of 16 random matrices used for all samples, $Q(\cdot)$ is a quantizer (positive as 1 and non-positive as 0) to output an $36 * 16$ bits binary string H. The post-processing consists of two steps: firstly, the first half of H is XORed by the latter half to downsize the binary string to $H/2$ bits; secondly, a N bit binary can be produced by discarding the last $H/2 - N$ bits, where we set $H/2 > N$ in binary template generation.

Since Bloom filters operate on a binary block with word size w, we propose the binary matrix B is divided into a set of blocks from both horizontal and vertical direction as shown in Figure 10.1. From horizontal direction, the columns are partitioned into 3 pieces separated at p^{th} column and q^{th} column. From vertical direction, the binary matrix will be divided into N/w parts. For instance, the first block is $B(1 : w, 1 : p)$ and the second block is $B(1 : w, (p + 1) : q)$. The total number of blocks is $3 * N/w$. Mapping each block BM_i into a Bloom filters b_i is similar to employ Bloom filters on iris recognition in paper [69]. A Bloom filter b is a bit array with length $2^w - 1$ and initially all bits to 0. The bit at position h_x of Bloom filter b will be flipped to 1 if the decimal value of a column is equal to h_x. The bit will remain at 1 even if there are multiple columns mapped to the same position. This is also the reason why Bloom filters meets the irreversibility requirement.

During the comparison phase, the dissimilarity score is calculated by using Equation (10.4) for two transformed templates, where we assume R as reference and P as probe.

$$DS(R, P) = \sum_{i,j=1}^{K} \frac{HD(b\_R_i, b\_P_i)}{|b_i| + |b_j|} \tag{10.4}$$

where $|b\_R_i| \neq 0, |b\_P_i| \neq 0$, K is the number of Bloom filters, b_R_i is the Bloom filter in reference template R and b_P_i is the corresponding Bloom filter in probe template P. $|b|$ denotes the amount of bits with value 1 in a Bloom filter b.

10.3 Performance evaluation

To evaluate the performance of template protection scheme, researchers generally apply the stolen-token case [190] which still guarantee the irreversibility. The following Equal Error Rate (EER) is calculated under this assumption. In addition, a corresponding unprotected EER is also calculated by directly using binary template without Bloom filters in order to analyse the impact of applying Bloom filters. A comparison score from these binary templates is calculated as the number of match cases of all columns in the reference template and all columns in the probe templates. We consider two columns are matched if the Hamming distance between these two columns is less than a threshold TH (empirically we set TH to 40). The experiments were conducted on FVC_2002_DB1A [143], FVC_2002_DB2A [143] and $MCYT\text{-}fingerprint$-100 [19]. The fingerprint extractor adopted in our experiments is NeuroTechnology Verifinger 6.0 Extractor [32] which sorts the minutia by its coordinate Y in default. The details of experimental setting and results are introduced as follows.

10.3.1 Experiments on FVC database

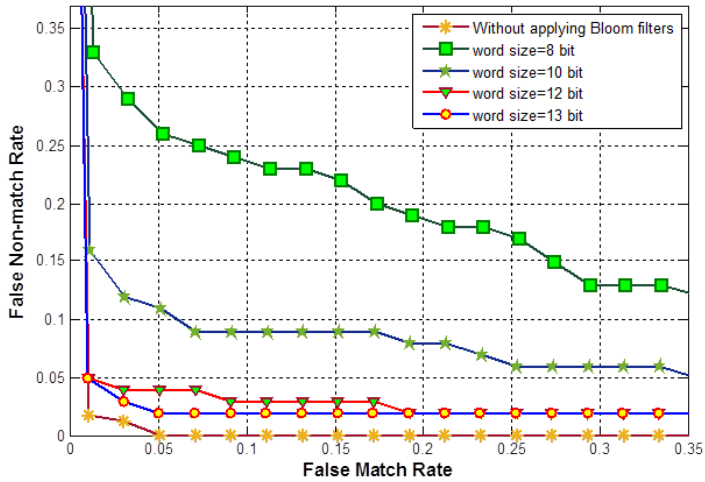


Figure 10.4: DET curve on $FVC2002\_DB1A$ under different word size (Setting one)

The performance was evaluated on FVC_2002_DB1A and FVC_2002_DB2A respectively. FVC_2002_DB1A consists of 800 samples which were captured from 100 fingers with 8 samples per finger. These samples have the size 388*374 pixels and are generally sorted by the sample quality in descending order. We designed two types of experiments to study the performance variation under different setting:

- Setting one: investigate the performance impact by varying the word size at 8, 10, 12 and 13. In this case, the first sample of each finger is enrolled as reference sample, and the second sample of each finger is used for probe sample.
- Setting two: investigate the performance impact by using different sample quality. In this setting, the first sample of each finger is still enrolled as reference sample, but

Table 10.1: EERs on $FVC2002\_DB1A$ under different word size (Setting one).

w	Blocks' number	EER after Bloom filters	EER without Bloom filters	EER difference
8	96	0.19	0.02	-0.17
10	75	0.09	0.02	-0.07
12	63	0.04	0.02	-0.02
13	57	0.03	0.02	-0.01

Table 10.2: EERs on $FVC2002\_DB1A$ using different probe samples (Setting two).

Probe samples	EER after Bloom filters	EER without Bloom filters	EER difference
Second sample	0.03	0.02	-0.01
Third sample	0.07	0.02	-0.05
Sixth sample	0.14	0.05	-0.09

Table 10.3: EERs on $FVC2002\_DB2A$ under different word size (Setting one).

w	Blocks' number	EER after Bloom filters	EER without Bloom filters	EER difference
8	96	0.16	0.01	-0.15
10	75	0.03	0.01	-0.02
12	63	0.01	0.01	0
13	57	0.005	0.01	+0.05

the probe sample will be chosen from second sample, third sample and sixth sample respectively. And the word size w is fixed at 13.

The radius r of the circle which is used in pre-alignment processing is set to 190 in FVC_2002_DB1A, and $p = 45, q = 90$. Figure 10.4 illustrates Detection Error Trade-off (DET) curve under setting one for FVC_2002_DB1A. We can observe that the performance significantly improves as long as word size w increases. On the other hand, the computational complexity also rises with word size. Therefore, increasing the word size has to stop at some point where the system can afford the complexity. Table 10.1 lists the ERRs after applying Bloom filters and ERRs without Bloom filters under Setting one. We can see accuracy performance slight decrease at word size $w = 13$. Table 10.2 gives the EERs under Setting two. Observed from the results, the fingerprint quality has heave impact on the accuracy performance which would be a challenging work in the future.

These two settings were also applied on database FVC_2002_DB2A which has the sample image with size 296×560 pixels. The parameters in this database were set as $r = 210, p = 45, q = 90$. Figure 10.5 illustrates the DET curve under Setting one using different word sizes. Table 10.3 lists the EERs for this setting, and Table 10.4 gives the EERs for using different samples as probe. We can see that the biometric performance even

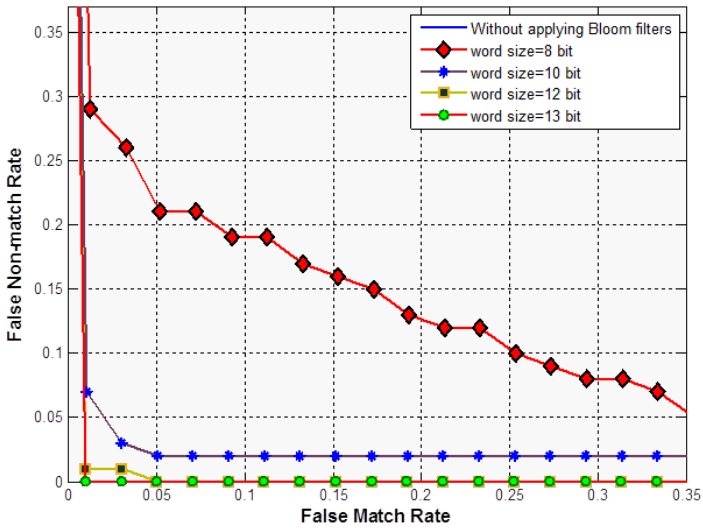


Figure 10.5: DET curve on $FVC2002\_DB2A$ under different word size (Setting one)

Table 10.4: EERs on $FVC2002\_DB2A$ using different probe samples (Setting two).

Probe samples	EER after Bloom filters	EER without Bloom filters	EER difference
Second sample	0.005	0.01	+0.005
Third sample	0.03	0.003	-0.027
Sixth sample	0.11	0.06	-0.05

slightly better than the performance without using Bloom filters, although the performance still suffers from the low sample quality.

10.3.2 Experiments on MCYT100

The experiments were also conducted on $MCYT\text{-}fingerprint\text{-}100$ [19] which consists of 100 subjects with 10 fingers used for fingerprint sample acquisition. We chose the sample with size 256×400 captured by an optical capture device which is model UareU from Digital Persona [19]. We selected the third sample of each finger as reference, and the second sample of each finger as probe due to the observation that these two samples have better quality comparing to the remaining samples. The rest of parameters were set as $r = 115, p = 45, q = 90$. Table 10.5 lists the ERRs for ten fingers respectively. The results show that using proposed approach on 6^{th} finger doesn't lose any information after applying Bloom filters comparing to the performance without Bloom filters. The performance on the rest of fingers slightly decreases.

Finger ID	EER after Bloom filters	EER without Bloom filters	EER difference
0	0.01	0.003	-0.007
1	0.04	0.02	-0.02
2	0.04	0.03	-0.01
3	0.07	0.03	-0.04
4	0.05	0.02	-0-03
5	0.03	0.01	-0.02
6	0.03	0.03	0
7	0.08	0.04	-0.04
8	0.09	0.07	-0.02
9	0.06	0.03	-0.03

Table 10.5: ERRs on database $MCYT$-$fingerprint$-100 running for ten fingers respectively

10.4 Conclusion

Due to the concerns of security and privacy on biometric data, we studied applying Bloom filters to protected the fingerprint template in this paper. A pre-alignment process is deployed before generating the binary template in order to be robust with the fingerprint sample translation. In addition, we proposed to divide the binary template matrix from both horizontal direction and vertical direction since the size of fingerprint binary template is large and variable. Experiments were conducted on $FVC2002\_BD1A$, $FVC2002\_BD2A$ and $MCYT$-$fingerprint$-100 respectively. According to the performance evaluation, the biometric performance doesn't degrade after applying Bloom filters if the fingerprint sample has good quality. Therefore, we can conclude that it is feasible to apply Bloom filters on fingerprint biometrics. Moreover, the biometric performance is still suffering from poor quality fingerprint images based on the experimental results. Our future work will focus on improving proposed approach which can be resilient to the low quality samples.

Chapter 11

A Fingerprint Indexing Algorithm on Encrypted Domain

Abstract

Fingerprint indexing has been extensively studied, and a number of approaches have been proposed in the literature. However, the vast majority of proposed approaches are based on original fingerprint templates without applying any protection mechanism. Secure fingerprint indexing algorithm has been rarely investigated. This paper presents a secure fingerprint indexing algorithm whose security is enhanced by a standard encryption algorithm. The proposed approach generates a binary template and creates an index space based on encrypted minutiae information. No original biometric information needs to be stored in the database, thus the adversary is unable to reverse the plain minutiae information without knowing the secret key. According to our experiments on both public datasets and a large-scale synthetic dataset, the proposed approach still maintains a very good performance in terms of low pre-selection error rate at small penetration rate.

11.1 Introduction

A biometrics system is an authentication system, which can automatically recognize individuals based on their behavioral characteristics (such as gait, keystroke, signature, etc.) or biological characteristics (such as fingerprint, face, iris, vein, voice, etc.) [143]. Fingerprint recognition is one of most studied biometric methods since the first research paper on fingerprint automatic comparison was published in 1963 [194]. In addition, fingerprint recognition based authentication is also the most widely deployed in the industry and government, especially in the forensics area. For instance, in 1969 U.S. FBI (Federal Bureau of Investigation) initiated an automated fingerprint recognition system [209], and it is hosting more than 70 million subjects with criminal background and 34 million civil prints according to the FBI's IAFIS website [7]. It may be time-consuming to perform an exhaustive comparison with all fingerprints stored in this large-scale database. In order to avoid an exhaustive searching, a common idea is to reduce the search space which can be accomplished by two methods for fingerprint: classification and fingerprint indexing. A famous classification method is called Henry classification system which categorizes the fingerprints into five classes: right loop, left loop, arch, whorl and tented arch [116, 194]. However, the fingerprints are unevenly distributed to these five classes (31.7% right loop, 33.8% left loop, 27.9% whorl, 3.7% arch and 2.9% arch) [143, 205]. Some researchers also proposed other classification methods by using different features extracted from the fingerprint [65, 119, 195, 110]. A common disadvantage of these classification methods is the limited number of classes. Fingerprint indexing is a technique which accesses a subset of the database by using a key value (also named index value) rather than searching whole database [47]. The purpose of fingerprint indexing is to output a short list of candidate identities which can be further used by a verification algorithm or human expert for manual verification.

Researchers have proposed quite a number of fingerprint indexing approaches, which generally consist of two components: (1) feature extraction; (2) index space creation and

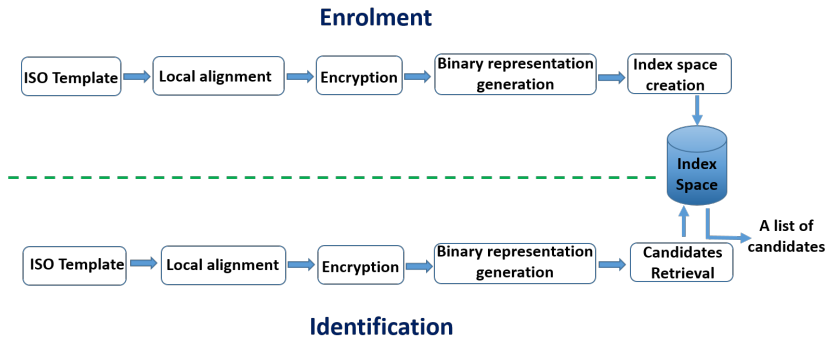


Figure 11.1: Block diagram of proposed fingerprint indexing approach which takes ISO standard fingerprint template as input and produces a short list of candidates identities as output.

candidates retrieval. Based on the features used for creating the index space, fingerprint indexing approaches can be coarsely classified into four categories.

- Global feature based approaches: orientation field [201, 110], ridge frequency [58] and singular points [139, 142] are commonly used for feature extraction. An unsupervised classification method (such as k-means) is commonly adopted for index space creation. This indicates that the index space also contains some plaintext information, for instance, a trained feature vector which represents each cluster after applying continuous classification.

- Local feature based approaches: minutiae and local ridge information are used to extract invariant features. Most of researchers have worked on using local features for designing a fingerprint indexing approach. For instance, proposed approaches in [76, 172, 51] chose combining minutiae and local ridge information for feature extraction. Proposed approaches in [50, 81, 96, 223, 131] focused on minutia information for feature extraction. A state of the art fingerprint indexing approach called MCC-based (minutiae cylinder-code) fingerprint indexing approach [62] only uses minutia information for feature extraction and LSH (Locality Sensitive Hashing) for index space creation, however, this approach requires an original minutia template for comparison after selecting a subset from the whole index space.

- Sweat pore based approaches: this type of approach generally combines sweat pore with minutia for feature extraction, $e.g.$ proposed approaches in [183] and [161]. During index space creation, the index values are calculated from these extracted features.

- Other approaches: A SIFT(reduced scale invariant feature transformation) based approach detects minutiae and special points for feature extraction [182]. A symmetrical filters based approach generates a fixed length feature vector based on the [135].

The vast majority of published fingerprint indexing approaches are designed on plaintext fingerprint template without considering any security mechanism. Even though most of these approaches don't directly store original minutia templates, they still need to store some indirect information (such as cluster centroid after classification) which is calculated from the original minutia templates. These indirect information may also leak people's biometric data once they are compromised.

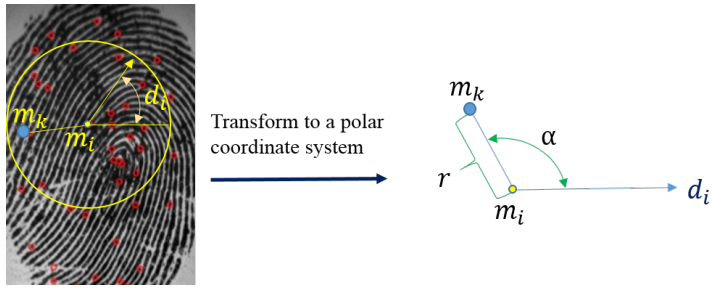


Figure 11.2: Local alignment: all minutiae included in the yellow circle are aligned with the central minutia in a polar coordinate system whose reference point is m_i and reference angle is the direction (denoted by d_i) of m_i. This central minutia and another minutiae included in the circle are named as a minutiae-disk.

Our work will focus on a secure fingerprint indexing approach which has been rarely studied so far. In the literature, we found a secure fingerprint indexing approach that was published in 2013 by Hartloff et al. [88] who proposed to calculate distances and certain angles from minutiae template. These distances and certain angles are further used for generating a set of paths, then the proposed approach applies fuzzy vault on these paths for creating the index space. The security capability of this proposed approach relies on fuzzy vault scheme which is vulnerable to brute force attack [145]. Recently, researchers also proposed solutions to address this drawback [189, 57], however, these improved fuzzy vault schemes haven't been adapted for fingerprint indexing. In this paper, we present a fingerprint indexing approach operating on the encrypted domain. The security goal of the proposed approach is to thwart the adversary from reversing the plain minutiae information without knowing the secret key. The proposed approach does not prevent linkability attack, since linkability is unavoidable in the indexing applications. The rest of this paper is structured as follows: Section 11.2 describes the details of the proposed approach; the performance of the proposed approach is reported in Section 11.3; at the end, Section 11.4 concludes this paper.

11.2 Fingerprint indexing on encrypted domain

Fingerprint indexing approach is a critical component for a large-scale fingerprint identification system which generally consists of enrolment stage (offline stage) and identification stage (online stage) as seen in Figure 11.1. The proposed secure fingerprint indexing approach takes ISO standard fingerprint templates as input and produces a short list of candidates identities as output, and it is composed of five modules: local alignment, encryption, binary representation generation, index space creation and candidates retrieval. The following subsections introduces these modules in details. Some of these modules follow the techniques based on our previous work in [130].

11.2.1 Local alignment

Due to the sample variants which commonly occur during data acquisition, it is difficult to properly align two samples that are stemming from the same source. Singular point based fingerprint alignment is a possible way to solve this problem, but accurately detecting singular point is quite challenging and a falsely detected singular point may have strongly negative impact for the subsequent processing steps. Instead of considering singular point based fingerprint alignment, we propose a local alignment method. The basic idea of this

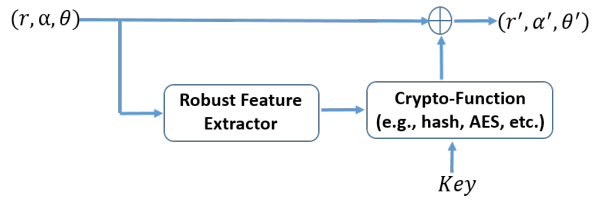


Figure 11.3: Work flow of proposed encryption module: a standard encryption algorithm ($e.g.$, hash, AES, etc.) is used to encrypt three attributes.

method is considering each minutia as a reference point and to transform to a polar coordinate representation, then nearby minutiae are aligned with respect to this reference point (called central minutia). As illustrated in Figure 11.2, all surrounding minutiae in the yellow circle are aligned with the central minutia in a polar coordinate system. This central minutia and another minutiae included in the yellow circle are defined as a minutiae-disk. Thus the number of minutiae-disks is equal to the number of minutiae in the fingerprint.

Mathematically, we assume a fingerprint template T which contains n minutiae $\{m_1, m_2, \ldots, m_n\}$ with three properties for each minutia: $m_i(x, y, d)$, where x and y are the minutia location and d is the minutia direction. A minutiae-disk (MD_i) can be formed for each minutia m_i. A polar coordinate system is defined by using m_i as reference point and the direction (denoted by d_i) of m_i as reference angle. Then each minutia m_k included in the minutiae-disk (yellow circle in Figure 11.2) will have a new coordinate $m_k^{'}(r, \alpha)$ denoted in Equation 11.1 and Equation 11.2.

$$r = DIS(m_k, m_i) \tag{11.1}$$

where DIS is the Euclidean distance between the two minutiae.

$$\alpha = \frac{(atan2(m_k(y) - m_i(y), m_k(x) - m_i(x)) + 2\pi) * 180}{\pi} \tag{11.2}$$

where function $atan2$ is 'Four-quadrant inverse tangent' defined in [11].

The minutiae direction difference θ between m_i and m_k is denoted by the following equation:

$$\theta = |m_k(d) - m_i(d)| \tag{11.3}$$

The units of these three attributes are pixel, degree of arc (range: 0-360), and degree of arc (range: 0-360). Let's assume there are J minutiae in a minutiae-disk except the central minutia. After the local alignment, we will have these J minutiae with new attributes denoted by $(m_1(r, \alpha, \theta), m_2(r, \alpha, \theta), ..., m_J(r, \alpha, \theta))$. The following encryption mechanism will operate on each aligned minutia.

11.2.2 Encryption

The proposed encryption mechanism adapts a standard encryption algorithm ($e.g.$, hash , AES, etc.) to secure fingerprint template as seen in Figure 11.3. The input of this encryption function relies on a 128 bits random value (called $NONCE$) and three robust attributes (r^*, α^*, θ^*) generated by a robust feature extractor which takes (r, α, θ) as inputs. These three robust attributes are calculated in Equations (11.4) - (11.6). A block cipher encoder (e.g., AES or 3DES) in the ECB mode or the CBC mode is chosen with using a secret key Key which encrypts all fingerprint samples enrolled in the database. Note that here we do not have the general security concern on the ECB mode assuming the three robust attributes are

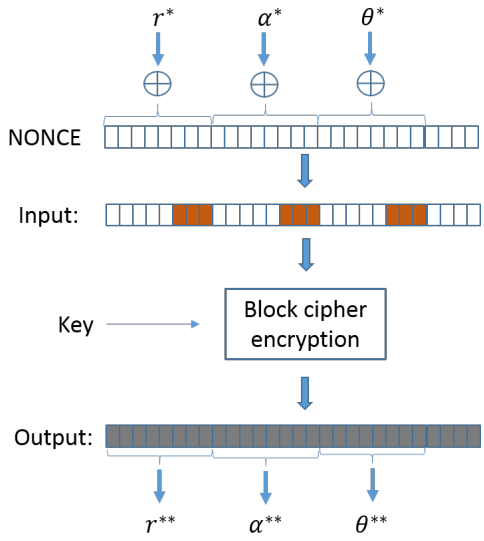


Figure 11.4: Input and output for a standard encryption algorithm: input relies on a 128 bit $NONCE$ and three robust attributes (r^*, α^*, θ^*). 9 bits in $NONCE$ will be affected after combining (r^*, α^*, θ^*). Output is the decimal format of selected bits.

largely different if the original minutiae attributes are different in one template. Figure 11.4 illustrates the details of input and output of a block cipher encryption. The input of the block cipher encryption is the combination of these three robust attributes and $NONCE$. Since the range of (r^*, α^*, θ^*) is integer values amongst (0,7), there are 9 bits of $NONCE$ that will be affected after this combination. The outputs (r^{**}, α^{**}, θ^{**}) are the decimal value of selected bits (8 bits for each attribute) as seen in Figure 11.4. At the end, the whole encryption module's output $(r^{'}, \alpha^{'}, \theta^{'})$ is the result from a modulo addition operation between (r, α, θ) and (r^{**}, α^{**}, θ^{**}), as seen in Figure 11.3. After the encryption process, the minutiae information $((m_1(r, \alpha, \theta), m_2(r, \alpha, \theta), ..., m_J(r, \alpha, \theta))$ in a minutiae-disk is secured and denoted by $(m_1^{'}(r^{'}, \alpha^{'}, \theta^{'}), m_2^{'}(r^{'}, \alpha^{'}, \theta^{'}), ..., m_J^{'}(r^{'}, \alpha^{'}, \theta^{'})$ which will be further used to generate a binary vector in the next subsection.

$$r^* = \lfloor r/45 \rfloor \; mod \; 8 \tag{11.4}$$

$$\alpha^* = \lfloor \alpha/45 \rfloor \; mod \; 8 \tag{11.5}$$

$$\theta^* = \lfloor \theta/45 \rfloor \; mod \; 8 \tag{11.6}$$

11.2.3 Binary representation generation in the encrypted domain

The proposed binary representation generation method produces a fixed length binary vector from each minutiae-disk based on encrypted minutiae information. The procedures of binary vector generation for each minutiae-disk is adapted from the binary vector generation method proposed in the article [130]. These procedures are also illustrated in Figure 11.5.

Before generating a binary vector, there is a training phase which will classify a set of $(r^{'}, \alpha^{'}, \theta^{'})$ into K clusters where each cluster is represented by its centroid: $\{C_1(r^{'}, \alpha^{'}, \theta^{'}), C_2(r^{'}, \alpha^{'}, \theta^{'}), ..., C_K(r^{'}, \alpha^{'}, \theta^{'})\}$. The unsupervised learning scheme $K - means$ is

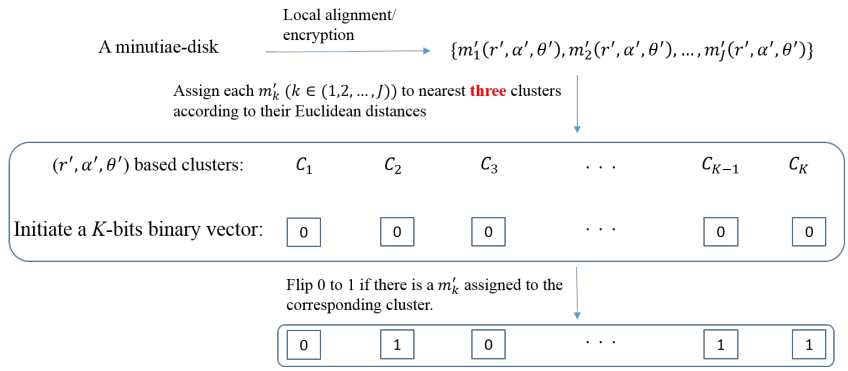


Figure 11.5: Procedures of binary vector generation based on encrypted minutiae information.

used for this classification purpose, since it has been proven to be appropriate for fingerprint indexing [172, 129].

As illustrated in Figure 11.5, there are four steps to generate a fixed length binary vector for a minutiae-disk:

Step 1: Apply local alignment and encryption mechanism on a minutiae-disk to generate encrypted minutiae: $m'_1(r',\alpha',\theta'), m'_2(r',\alpha',\theta'), ..., m'_J(r',\alpha',\theta')$.

Step 2: Initiate a K bits binary vector with all components set to 0.

Step 3: Assign each $m'_k(r',\alpha',\theta'), k \in (1,2,...,J)$ to nearest three clusters (closest cluster, second closest cluster and third closest cluster) according to their Euclidean distances.

Step 4: Flip 0 to 1 if there is a m'_k assigned to the corresponding cluster. Eventually, a binary vector can be generated to represent this minutia-disk.

In step 3, the reason of choosing nearest three clusters is to tolerate sample intra-class variations. In step 4, only the first change will take effect even if multiple m'_k have been assigned to the same cluster. At the last, the number of binary vectors generated for a fingerprint sample is equal to the number of minutiae in this sample. This binary template is an encrypted template, since we directly generate it from encrypted minutiae information. And this secure binary template will be used for index space creation and candidates retrieval in the following subsection.

11.2.4 Index space creation and candidates retrieval

Locality Sensitive Hashing (LSH) is selected for creating index space. **Algorithm** 11.1 gives the details of index space creation method whose inputs are ISO standard fingerprint templates and a set of hash functions. Each of these hash functions will randomly select η bits from each binary vector. The result of each hash function is the input for a function called $CountOneBits(b)$ whose purpose is to count the number of 1 bits in the binary vector, for instance $CountOneBits(1010001) = 3$. The pair (i, j), where i is the template ID and j is the minutiae ID, will be recorded only when $CountOneBits(b)$ is not less than a parameter $min_{OneBits}$. Note that the difference between this index space creation method with the one used in [130] is that we don't need to store original minutia templates. This difference also leads to the changes during candidates retrieval.

Algorithm 11.2 gives the procedures of retrieving candidates identities from an index space which is represented by a set of hash tables denoted by $\{H_1, H_2, ..., H_\Lambda\}$. The hash functions used during candidates retrieval are the same as used in **Algorithm** 11.1. An-

Algorithm 11.1 Indexing tables creation on encrypted domain

Require: ISO standard templates of enrolled subjects: $\{T_1, T_2, ..., T_E\}$;
Hash functions:$\{f_{H_1}, f_{H_2}, ..., f_{H_\Lambda}\}$ (Λ is the number of hash functions);
Ensure: Indexing tables: $H_1, H_2, ..., H_\Lambda$

for each ISO standard template $T_i (i \in 1, 2, ..., E)$ **do**
  Generate binary template RB_i by using proposed method including local alignment module, encryption module and binary representation generation module. We assume $RB_i = \{T(i,1), T(i,2), ..., T(i,J)\}$, where is the number of binary vectors generated from minutiae temple T_i of RB_i
  for each binary vector $T(i,j) (j \in 1, 2, ..., J)$ **do**
    for each hash function f_{H_λ} **do**
      $b = f_{H_\lambda}(T(i,j)))$
      if $CountOneBits(b) \geq min_{OneBits}$ **then**
        record (i,j) in $b-th$ bucket of indexing table H_λ.
      end if
    end for
  end for
end for

other input for candidates retrieval is the probe template P which is also compliant with the standardized ISO/IEC minutiae format [99]. Since the original minutiae templates are not stored in the database, the similarity score will rely on the collisions in those targeted buckets.

Algorithm 11.2 Candidates retrieval on encrypted domain

Require: Indexing tables: $H_1, H_2, ..., H_\Lambda$;
Hash functions:$\{f_{H_1}, f_{H_2}, ..., f_{H_\Lambda}\}$;
ISO standard template of probe sample: P.
Ensure: Candidate entities.

Generate binary template PB for probe sample by applying as same procedures as used in enrolment stage. We assume $PB = \{V_1, V_2, ..., V_\Omega\}$, where Ω is the number of binary vectors of PB;
Initiate an array to store similarity score: $S[E]$;
for each binary vector V_ω **do**
  Assume m_ω is the central minutia associated with binary vector V_ω
  for each hash function f_{H_λ} **do**
    $b = f_{H_\lambda}(V_\omega)$
    if $CountOneBits(b) \geq min_{OneBits}$ **then**
      for each pair (i,j) in $b-th$ bucket of indexing table H_λ **do**
        $S[i] = S[i] + 1$;
      end for
    end if
  end for
end for
Sort $S[E]$ by descending order, and select the top-N as candidate entities.

Table 11.1: Parameters setting for all experiments.

Parameter	value	Remark
R	300 pixels	the radius of the minutiae-disk
K	1024	the length of binary vector
Λ	48	the number of hash functions
η	32	the number of bits selected by hash function
ρ	256	minutia distance threshold
σ	45	minutia direction difference threshold
$min_{OneBits}$	2	the number of '1' bits in a binary vector
encryption algorithm	AES-128	a standard encryption algorithm

11.3 Performance evaluation

11.3.1 Evaluation metrics and settings

The main purpose of experiments is to evaluate whether the proposed secure fingerprint indexing approach still maintains a good accuracy performance. The fingerprint indexing approach in [130] generates the binary templates without considering any protection mechanism, thus this approach can be considered as a plaintext domain based fingerprint indexing algorithm. We will use this plaintext domain based approach as a benchmark to investigate the performance variation of the proposed approach.

In accordance with ISO/IEC 19795-1 [97], the performance of fingerprint indexing algorithm is reported by two criteria: penetration rate and pre-selection error rate. Penetration rate is a proportion of enrolled references in a database where the identification system has to search. A pre-selection error occurs when the enrolled reference corresponding to the probe sample is not included in the pre-selected candidates. Generally speaking, the better fingerprint indexing approach will achieve lower pre-selection error rate at the same penetration rate comparing to other approaches. Table 11.1 lists the parameters used in our experiments. The minutia templates were extracted by a commercial product VeriFinger 7.0 from 'Neurotec Biometric 5.0 SDK' [32].

In addition to the experiments on several public datasets, we also evaluate our approach in a large-scale synthetic fingerprint dataset. The following subsections will report the results on these datasets.

11.3.2 Evaluation on public datasets

There are two datasets selected from $FVC2002$: $FVC2002\_DB1\_A$ and $FVC2002\_DB2\_A$. Each of them is composed of 800 samples from 100 fingers. The first sample of each finger is enrolled in the database during index space creation process. The rest of samples (in total 700 samples) are used as probe samples. Figure 11.6 and Figure 11.7 demonstrate the performance on these two dataset respectively. We can see that the proposed secure fingerprint indexing approach achieves the same performance as the approach in the plaintext domain on $FVC2002\_DB1\_A$. The performance on $FVC2002\_DB2\_A$ is slightly worse than the performance in the plaintext domain.

$FVC2006\_DB2\_A$ is the optical sensor based dataset in $FVC2006$. It has 1680 samples which were captured from 140 fingers. By default, the first sample of each finger is enrolled in the database. The rest of the samples (in total 1540 samples) are used as probe samples during searching. Figure 11.8 demonstrates the performance which indicates that

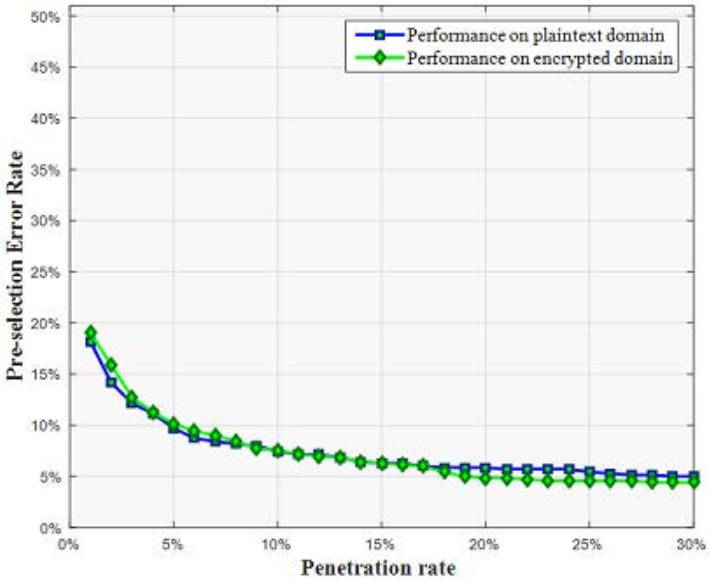


Figure 11.6: Performance evaluation on $FVC2002\_DB1\_A$.

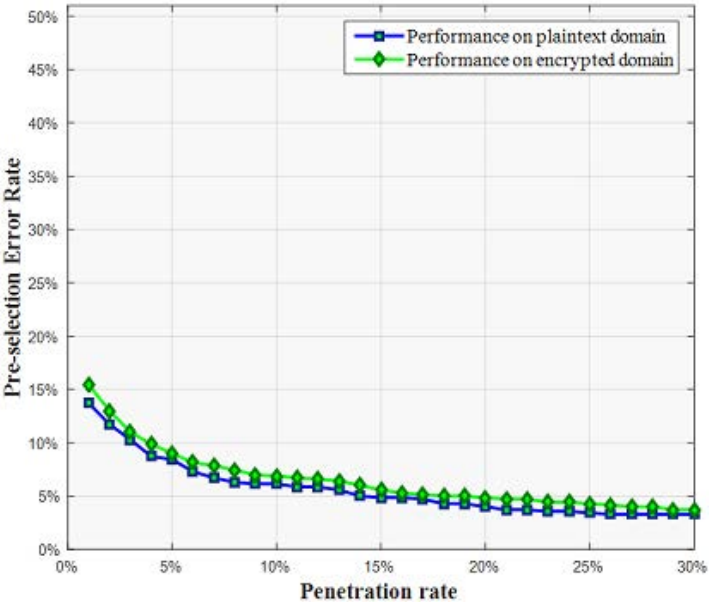


Figure 11.7: Performance evaluation on $FVC2002\_DB2\_A$.

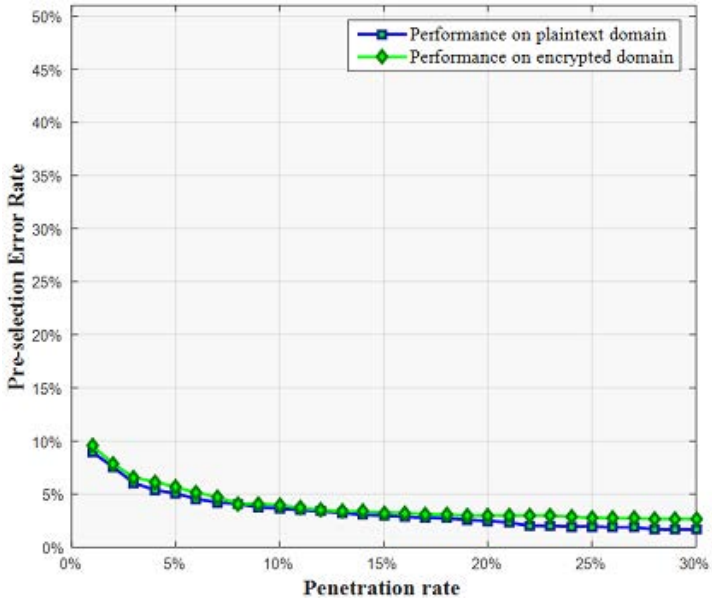


Figure 11.8: Performance evaluation on $FVC2006\_DB2\_A$.

the proposed approach still maintains very good performance after applying a security mechanism.

11.3.3 Evaluation on a large-scale dataset

The large-scale dataset is composed of a reference dataset and a probe dataset. The reference dataset consists of 250,000 subjects with two samples (can be considered captured from left index finger and right index finger respectively) for each subject. Each subject is assigned an unique ID from 1 to 250, 000. The first 400 subjects have real fingerprints selected three datasets: $FVC2004\_DB1\_A$ [143], BIoSec_FO which is distributed by the Biometric Recognition Group of the Universidad Autonoma de Madrid [3], and FP_CM_V300 which is distributed by the Chinese Academy of Sciences [109]. The rest of samples are generated by a synthetic generator SFinGE developed by University of Bologna [28]. The probe dataset consists of 50,000 subjects which have the same sources as the first 50,000 subjects in the reference dataset. In total, there are 600,000 samples involved in our experiment.

There are two samples (from different finger) for each subject for both the reference dataset and the probe dataset. This gives us the opportunity to simulate a practical identification system which normally uses two fingers to identify a subject, thus we apply a score-level fusion on two samples. This fusion method indicates that the output candidates will be decided by the fused similarity scores generated from these two samples.

As same as we have conducted on public datasets, we evaluate the proposed approach on plaintext domain and encrypted domain respectively. Since the purpose of the fingerprint indexing algorithm is to produce a short list of candidate identities which can be further used by verification algorithm or human expert for manual verification, the length of this short list shouldn't be large. We consider to output 200 candidates which turns out a 0.08% penetration rate. Table 11.2 lists the penetration rate and pre-selection rate for the approach in plaintext domain and the approach in encrypted domain. According to these

Table 11.2: Performance evaluation on a large-scale dataset which consists of 250,000 reference subjects and 50,000 probe subjects.

Penetration rate	Pre-selection error rate on plaintext domain	Pre-selection error rate on encrypted domain
0.004%	0.668%	0.61%
0.002%	0.478%	0.44%
0.02%	0.32%	0.294%
0.04%	0.278%	0.27%
0.06%	0.256%	0.254%
0.08%	0.236%	0.234%

results, both of approaches achieved very low pre-selection error rate at small penetration rate. The performance of the approach in encrypted domain even doesn't deteriorate. A possible reason is that the synthetic fingerprint has relatively good quality based on our observation.

11.4 Conclusion

In this paper, a secure fingerprint indexing algorithm is presented based on a standard encryption algorithm. The proposed approach only relies on minutia information which can be obtained from ISO standard fingerprint template. An encryption mechanism is applied on the minutiae information after a local alignment process. A binary templates generation method is designed by using these encrypted minutia information. Index space creation and candidates retrieval are conducted on the binary template without using any original minutia information. We conducted our experiments on both public datasets (real fingerprints) and a large-scale synthetic dataset which consists of 250,000 reference subjects and 50,000 probe subjects. According to the experimental results, we found out that the fingerprint indexing approach on encrypted domain performs slightly worse on real fingerprints datasets compared to the fingerprint indexing algorithm on plaintext domain, but doesn't show any deterioration on the large-scale synthetic dataset. A possible reason is that the synthetic fingerprint has relatively good quality. This also motivates us to improve the performance on low quality fingerprints in the future. Another future work would be to extract features by incorporating other information included in a ISO standard fingerprint template, such as finger quality and minutia quality, etc.

Part V

Conclusions

Chapter 12

Conclusions

We conclude our work in this chapter. Section 12.1 gives a summary of results in this dissertation, and Section 12.2 discusses the future work.

12.1 A summary of results

Because of the wide acceptance for establishing the identity of an individual, we believe that the fingerprint recognition based system will continue to play an important role in the biometric system. In this dissertation, we studied several topics about the fingerprint identification system and developed a couple of approaches which would contribute to improving the performance of large-scale fingerprint identification systems. A summary of our results is as follows.

- We demonstrated that the quality of the fingerphoto taken by the smartphone's camera can be assessed by proposing an approach to qualify the fingerphotos captured in three different real-life scenarios. We assume the proposed approach also has the capability to assess the quality of the fingerphotos taken from other contactless devices (such as webcam, digital camera), since the photos taken by these cameras have the very similar characteristics in terms of background, focusing difficulty and varied illumination conditions.

- We developed three fingerprint indexing approaches based on different feature extraction methods. We also improved the classification-based index space creation method in order to better suit the features generated by our won approaches. By integrating an encryption module into the fingerprint indexing approach proposed in Chapter 9, we designed a fingerprint indexing algorithm in the encrypted domain. The proposed approach enables that no plaintext fingerprint data need be stored in the database, meanwhile it achieves the similar performance (in terms of penetration rate and pre-selection rate) as the fingerprint indexing approach without considering a security measure.

- Besides the fingerprint indexing algorithm in the encrypted domain, we also developed a fingerprint template protection scheme based on Bloom filters, while Bloom filters have been successfully used to protect face and iris data. According to the experimental results, the proposed approach has achieved promising performance on the tested datasets.

- Fingerprint alignment is a challenging topic in a fingerprint recognition algorithm. Several self-alignment modules have been designed in our work. In Chapter 8, we designed a self-alignment module based on a minutiae vicinity. In Chapter 9 and Chapter 11, we designed another self-alignment module which considers every minutiae as the origin in a polar coordinator system. In Chapter 9 , a self-alignment module is developed to exclude the minutiae which are far away to the center of the fingerprint area.

12.2 Future work

With the rising deployment of the fingerprint identification systems in different devices and diverse scenarios, new challenges will continuously emerge, and the existing research aspects also need to be further studied in order to achieve desirable performance in the new applications. In this section, we discuss the future work from two perspectives: (1) future work related to the selected researcher aspects; (2) future work on other research aspects related to a fingerprint identification system.

- **Further evaluation and improvement on the fingerphoto quality assessment approach**.

  Besides the future work discussed in Chapter 4 $\sim$ 6, we think the proposed fingerphoto quality assessment approach needs to be further evaluated and improved. The approached fingerphoto quality assessment approach was evaluated on a dataset collected from 100 fingers under three scenarios: the indoor-scenario, the outdoor-scenario and the dark-scenario. In order to evaluate the robustness under different scenarios, we think the test dataset needs be expanded to include more subjects and more real-life scenarios. In addition, a potential improvement can be achieved by analysing the color of the fingerphoto. The existing quality assessment approaches including our proposed approach extract features from the gray-level image. There is information missing when we convert a color image to a gray-level image, and we can observe the color difference between the finger area and background area. By analysing this color difference, we believe at least the false detection rate can be reduced.

- **Fingerprint indexing algorithm**.

  The database's size of the running fingerprint identification system is continuously growing, which implies that the pre-section error rate could be rising when the number of retrieved candidates remains same (in this case, the penetration rate decreases). In order to reduce the pre-section error rate, it is essential to improve the existing approaches. There are two potential ways to achieving the improvement based on our current approaches: (1) generate more robust features by combining the global feature (such as a reference point); (2) fuse the proposed binary feature with the existing binary features (such as the binary vectors from minutiae cylinder code [61] or the binary vectors from minutiae vicinities [53]). Besides these future work, we insist that studying the fingerprint indexing algorithm with security mechanism deserves more attentions, since the vast majority of researchers are focusing on developing fingerprint indexing algorithm without considering any security measure, and the people are getting concerned about the security and privacy of their biometric data.

- **Performance metrics for evaluating fingerprint indexing algorithm**.

  We think the current performance metrics for evaluating the fingerprint indexing algorithm are not sufficient. In the literature, the majority of fingerprint indexing algorithms report their performance in terms of pre-selection error rate (or another way around: hit rate) and penetration rate. For instance, the penetration rate will be 1% if the number of candidates retrieved by a fingerprint indexing algorithm is 10 and the number of enrolled references is 1000. We think this penetration rate reports the performance in terms of the overall purpose of the fingerprint indexing algorithm (whose output is a shot list of candidates). It doesn't reflect how much actual searching space saved, since reducing the searching space is the key that a fingerprint indexing can improve the efficiency in a large-scale fingerprint identification system. Let's use an example to explain this. Without loss of generality, we assume the number of retrieved candidates is still 10 and the number of enrolled reference is 1000. A index

space consists of 100 clusters. During the enrolment stage, these 1000 reference samples will be assigned into 100 clusters. Note that a reference sample maybe assigned into a number of clusters, and the number of each reference sample is variable. For instance, the first reference sample is assigned into 20 clusters and the second reference sample is assigned into 30 clusters. During candidate retrieval stage, the number of the searching space will be 25 clusters if a probe sample (called the first probe) is assigned into 25 clusters. After locating these 25 clusters, the candidates for the first probe will be the top 10 most appeared reference samples in these 25 clusters. We report the penetration rate at 1%, and a pre-selection error occurs if the mated reference sample is not included in these retrieved 10 candidates. If another probe sample (called the second probe) is assigned into 35 clusters in order to retrieve 10 candidates as well, the searching space will be 35 clusters but we still report penetration rate at 1%. As we can observe, the searching space for the first probe and the second probe is different (25 clusters and 35 clusters respectively) but not reported as a performance indicator. Unlike face and iris which normally have a fixed-number features to represent a sample, it is difficult to develop a fixed-number features from a fingerprint sample. We think this is the reason that leads to miss the information about how much actual searching space saved by the fingerprint indexing algorithm. This is also why the current two metrics are not sufficient and we need to define new performance metrics for evaluating the fingerprint indexing algorithm.

- **Fingerprint template protection algorithm**.

  A common challenge for all biometric template protection schemes is the recognition accuracy degradation after meeting the security requirements (irreversibility, unlinkability and revocability) [155]. We also need to improve the recognition accuracy of our proposed fingerprint template protection approach. Recently, researchers [55, 90] also raised the vulnerabilities for Bloom filters based iris template protection scheme, hence analysing and addressing these vulnerabilities on our won approach are also the future work.

Another interesting future work is to create a large-scale fingerprint dataset as a reference dataset and a latent fingerprint dataset as a probe dataset. These latent fingerprints have mated fingerprints in the reference dataset. Since identifying a latent fingerprint in a large-scale fingerprint database is a typical use case in the law enforcement agencies, evaluating the fingerprint indexing algorithm on such datasets reflects the reality need.

In addition, we think the following research aspects deserve more attentions. The improvement from these research aspects can benefit the above topics as well.

- **Fingerphoto recognition algorithm**.

  Besides the fingerphoto quality assessment, we think the rest of fingerphoto based research aspects (such as, segmentation, feature extraction and comparison algorithm) are also very promising due to the popularity of the mobile devices. However, it is very challenging to process the fingerphoto as we discussed earlier, and the most of existing algorithms are developed for the fingerprint captured from the dedicated sensor. For instance, a minutia is considered as a robust feature in the fingerprint. Majority of the fingerprint recognition algorithms are based on the minutia feature. But we observe that the number of minutiae that can be extracted from a fingerphoto is much fewer than the number of minutiae in a fingerprint, which may cause a very high error rate for the existing recognition algorithms when applying them on the fingerphoto. There are a few fingerphoto recognition algorithms published in the literature as we discussed in Section 2.2, but their algorithms and evaluation are limited on their own datasets. We believe these topics deserve more in-depth studies.

- **Pre-processing algorithms**.

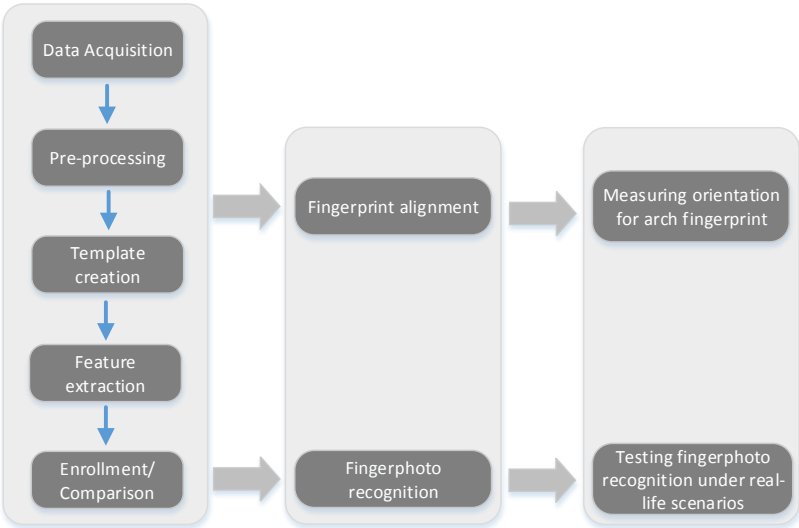
In Section 1.2, we have discussed the importance of pre-processing algorithms including segmentation, orientation field estimation, enhancement, binarization and thinning. These pre-processing algorithms are extremely important for the low-quality fingerprints and the latent fingerprints, especially when the latent fingerprints are frequently used in the fingerprint identification system.

- **Global reference point detection**.

  Detecting a reliable global reference point for all types of fingerprints is challenging. If a reference point can be precisely detected from the fingerprint sample, the subsequent processes can benefit a lot by considering this reference point. For instance, invariant features can be extracted based on the reference point and used by a fingerprint indexing algorithm. The computational complexity of the comparison algorithm can be significantly reduced by using the reference point to avoid cross comparison amongst minutiae. A global reference point is also the cornerstone to properly align two fingerprint samples which are from the same source. We notice that a number of reference point detection approaches have been presented in literature, but the problem is not well solved yet and would be the future work as well.

Part VI

Appendix



Besides the three main research aspects, we also worked on other two topics: measuring orientation for the arch fingerprint and testing fingerphoto recognition under real-life scenarios. We categories these two topics into fingerprint alignment and fingephoto recognition respectively as seen in the above figure.

In Appendix A, we designed an approach to measure the fingerprint orientation of arch fingerprint by using a set of isosceles triangles and an input point with high curvature. As we mentioned in the structure section (Section 1.4), this approach is semi-automated approach which is the reason that we don't include it in the main part. However, this approach can be further improved to an automated approach and integrated into the fingerprint identification system. In Appendix B, we investigated if it is feasible to recognize an individual by using the fingerphoto taken from a smartphone's camera under three real-life scenarios. The experimental results revealed that it is very challenging to use those fingerphotos for recognition due to the unstable and low quality samples. These results motivated us that it is essential to control the quality of these fingerphotos in order to improve the performance. This is also the reason that we'd like to report this work in the appendix.

The work in Appendix A was published in [127] GUOQIANG LI, CHRISTOPH BUSCH, BIAN YANG . "A novel approach used for measuring fingerprint orientation of arch fingerprint". In Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2014 37th International Convention on (pp. 1309-1314). IEEE.

The work in Appendix B was published in [133] GUOQIANG LI, YANG BIAN, R. RAGHAVENDRA AND CHRISTOPH BUSCH. "Testing mobile phone camera based fingerprint recognition under real-life scenarios." NISK 1 (2012): 2.

Appendix A

A novel approach used for measuring fingerprint orientation of arch fingerprint

Abstract

It is no doubt that fingerprint recognition is the most common biometric modality, which can be used to authenticate the identity of a person. Automatic identification systems based on fingerprint recognition have been extensively deployed in the industrial and forensics area. The performance of these systems relies on the accuracy of the fingerprint comparison algorithm, which is still suffering from the potential displacement and rotation that might occur during fingerprint sample capture process. Especially the strong rotation might decrease the identification accuracy. In order to address the rotation issue, two fingerprint samples can be aligned to each other by the analysis of the fingerprint orientation, which is generally represented by the orientation of core point and potentially supported by the connecting line to delta(s). However, for arch fingerprint patterns such a core point doesn't exist, and some researchers consider the point with maximum curvature as a reference point for an arch fingerprint. But measuring a robust fingerprint orientation for an arch fingerprint is still desirable. In this paper, we propose an approach to measure the fingerprint orientation of arch fingerprints using a set of isosceles triangles. We applied the proposed approach to measure the fingerprint orientation for 80 arch fingerprint samples, which are selected from the challenging CASIA Fingerprint Image database Version 5.0. The experimental results show that the proposed approach is feasible to measure the fingerprint orientation for the arch fingerprints, even for a partial imprint with strong rotation.

A.1 Introduction

Automatic fingerprint identification systems (AFIS) have been extensively deployed in diverse application scenarios due to their properties regarding uniqueness and permanence of a fingerprint as a physiological characteristics. High accuracy is the crucial requirement in these AFIS system, especially when the fingerprint identification is used as a tool for law enforcement and forensic investigations. In order to achieve a better performance, properly aligning the reference sample and the probe sample is the critical step in the context of comparing two fingerprint samples, specifically if the fingerprint samples are captured in uncontrolled conditions. There are three major challenges involved in fingerprint sample acquisition: translation, rotation and scaling. Researchers have been working on addressing these challenges since the fingerprint recognition was embedded into automatic identification systems. The most effort focused on translation and rotation issues, because capturing fingerprint images from habituated subjects that are aware of the appropriate pressure generally doesn't lead to the scaling problem. Detecting a singular point as a global feature has been commonly used to process the fingerprint alignment by researchers [204, 144]. In particular, the location of singular points can be used to overcome the translation, and the direction of the singular point (such as the core point orientation, which generally reflects the global orientation of the pattern represented in the sample) can be applied to address the rotation issue. The singular point or reference point detection has been studied over the last two decades. Recently significant achievements have been obtained: tented arch

fingerprint, left-loop fingerprint, right-loop fingerprint and whorl fingerprint can be easily detected by analysis of singular points[64, 184, 142]. But finding a robust reference point and it's orientation for plain arch fingerprint still remains challenging [139]. This is also reflected in commercial of the shelve fingerprint comparison subsystems (e.g. the NeuroTechnology Verifinger 6.0 product), which is unable to extract singularities for the vast majority of arch fingerprint samples in our experiments (this will be described in details in Section A.3). Therefore, it is still desirable to detect a reliable reference point and measure its orientation for an arch fingerprint. Thus our work will focus on measuring a robust fingerprint orientation for these rare but specifically challenging fingerprint patterns.

A variety of approaches have been proposed to detect the reference point and the orientation for arch fingerprint in the literature. Most of methods operate on fingerprint orientation field of the fingerprint sample [139, 198, 118] or curvature measurement [147, 70]. One typical approach of using an orientation pattern was proposed by Liu et al. [139]. They developed a multi-scale analysis of orientation consistency, and a reference point of arch fingerprint can be located by filtering the high orientation consistency. Another method using curvature measurement was presented by Nandakumar et al., who used this alignment method for their fuzzy vault scheme in [156]. Their method relies on a set of points with high curvature values called helper data. In order to obtain this helper data, an orientation field flow curve is extracted first based on the orientation field estimation. This flow curve is a global trait and similar to fingerprint ridges without breaks and discontinuities, thus it will be sensitive to noise [70]. The wrong flow curves caused by noise will impact the calculation of curvature values, and subsequently influences the determination of helper data. Both of these methods were based on the global orientation field, which are sensitive to noise. In contrary to previous methods, we will focus on exploring the local ridge patterns to measure the fingerprint orientation with the help of multiple triangles which are approximating a singular point.

The remaining of this paper is organized as follows: Section A.2 describes the initial idea and procedures of our proposed approach; experimental set-up and results are introduced in Section A.3; the conclusions are drawn in Section A.4.

A.2 fingerprint orientation measurement

A.2.1 Initial idea for arch fingerprint alignment

Since arch fingerprints don't contain a core point in a strict sense [139], researchers consider the point with maximum curvature as a reference point for an arch fingerprint [139, 118]. In order to tolerate the noise that is present in a fingerprint sample, we consider using self-similar triangles (called isosceles triangle) that are defined on this reference point, which can be used to align two fingerprint samples. Figure A.1(a) and Figure A.1(b) illustrate two isosceles triangles detected in the reference sample and probe sample respectively. Rotating the probe sample to align with the reference can be achieved by these two isosceles triangles, as shown in Figure A.1 (c). Measuring the fingerprint orientation is the critical step to accurately align these two samples. Meanwhile, the fingerprint orientation can be calculated for both samples by using detected isosceles triangles. Furthermore, the point with maximum curvature is not strictly required in this idea, since neighboring ridge lines show approximately the same orientation and thus neighboring points (on parallel ridges) with high curvature (as seen in Figure A.2) also can be used as input points to detect the triangles. In consideration of generating a robust fingerprint orientation, the proposed approach will detect a set of self-similar triangles (*i.e.*, isosceles triangles) instead of only one triangle. The details of this approach are described in the following section.

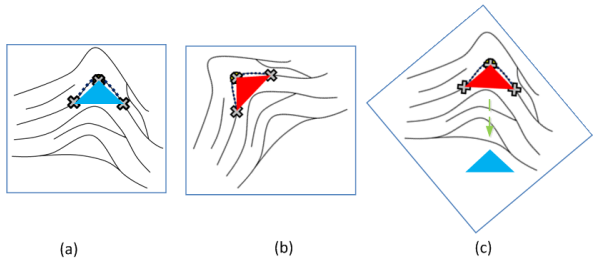


Figure A.1: Initial idea for arch fingerprint alignment: (a) a triangle is detected in the reference sample; (b) a triangle is detected in the probe sample; (c)Aligning two samples based on detected triangles.

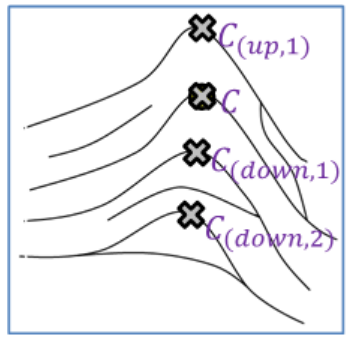


Figure A.2: The proposed approach does not only rely on the point with maximum curvature. For a given example - these three points can be used as input point as well.

A.2.2 Procedures of measuring fingerprint orientation

In accordance with ISO/IEC 19794-2:2011 [99], the fingerprint orientation is measured as an angle with respect to the horizontal axis from right to the left. This angle is generally reflected by the direction of the core point detected in the fingerprint sample. As we know arch fingerprints do not have such singular point. Instead, researchers [142, 77] usually consider that the fingerprint orientation of arch fingerprint is approximated from horizontal axis to the symmetrical axis of the orientation field. Figure A.3 illustrates five fingerprint samples with fingerprint orientation $\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5$ respectively. These five fingerprint samples were captured from the same source and are taken from CASIA-FingerprintV5 database [5].

The proposed approach is based on the binary image which can be obtained from the original fingerprint sample by applying the method provided in [9]. We assume this binary image is denoted by a matrix $\boldsymbol{M}_{m\times n}$. The input data for our proposed method is location of some point (h, g) with high curvature as described in the former section, assuming that holds $1 \leq g \leq m, 1 \leq h \leq n$, and the output is the fingerprint orientation $O, 0 \leq O < 2\pi$. The procedures of measuring this fingerprint orientation include 3 steps that will be described as follows.

Step 1: Detecting the central isosceles triangle

Before detecting the central isosceles triangle along the ridge pattern by using the input

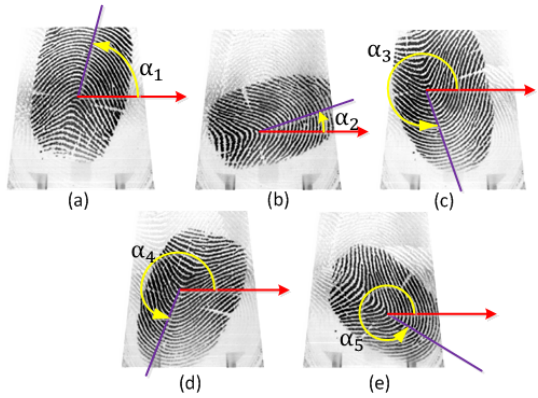


Figure A.3: Illustration of fingerprint orientation. Fingerprint samples are from CASIA-FingerprintV5 [5].

point, it is necessary to adjust the input point (h, g) to the closest ridge unless it is already located on the ridge within tolerance bounds. We assume the input point is denoted by (i, j) after the adjustment. Starting from the input point two neighboring vertexes indicated by (l_i, l_j), (r_i, r_j) are searched that are located on the same ridge line and will constitute the central isosceles triangle. We define a function called '**Detecting_triangle**', which requires input point (i, j) and will return the two vertexes (l_i, l_j), (r_i, r_j). This function also defines the traveling distance ($i.e.$ side length) between (i, j) with these two vertexes as L. The fingerprint orientation range is divided ($O, 0 \leq O < 2\pi$) into four parts which will be called 'tracking direction' in the rest of part: northeastern part ($P_1, 0 \leq O < \pi/2$), northenwestern part ($P_2, \pi/2 \leq O < \pi$), southwestern part ($P_3, \pi \leq O < 3/(2\pi)$), and southeastern part ($P_4, 3/(2\pi) \leq O < 2\pi$). The function '**Detecting_triangle**' is composed of three steps.

$(i-2, j-2)$	$(i-2, j-1)$		$(i-2, j+1)$	$(i-2, j+2)$
$(i-1, j-2)$	$(i-1, j-1)$	$(i-1, j)$	$(i-1, j+1)$	$(i-1, j+2)$
	$(i, j-1)$	$\boldsymbol{(i, j)}$	$(i, j+1)$	
$(i+1, j-2)$	$(i+1, j-1)$	$(i+1, j)$	$(i+1, j+1)$	$(i+1, j+2)$
$(i+2, j-2)$	$(i+2, j-1)$		$(i+2, j+1)$	$(i+2, j+2)$

Figure A.4: Neighborhood of the input point (i, j).

First a starting tracking direction $P_k, k \in \{1, 2, 3, 4\}$ is determined by calculating the

binary values of neighboring points (as seen in Figure A.4) around input point (i, j):

$$O_1 = \sum_{\substack{i-2 \le x \le i-1 \\ j+1 \le y \le j+2}} \boldsymbol{M}(x, y) \tag{A.1}$$

$$O_2 = \sum_{\substack{i-2 \le x \le i-1 \\ j-2 \le y \le j-1}} \boldsymbol{M}(x, y) \tag{A.2}$$

$$O_3 = \sum_{\substack{i+1 \le x \le i+2 \\ j-2 \le y \le j-1}} \boldsymbol{M}(x, y) \tag{A.3}$$

$$O_4 = \sum_{\substack{i+1 \le x \le i+2 \\ j+1 \le y \le j+2}} \boldsymbol{M}(x, y) \tag{A.4}$$

We use the first minimal value from $[O_1, O_2, O_3, O_4]$ as the starting tracking direction. We assume this starting tracking direction is $P_k, k \in \{1, 2, 3, 4\}$, also defined in Equation A.5.

$$P_k = minimum[O_1; O_2; O_3; O_4] \tag{A.5}$$

Secondly, we define an additional function '**Detecting_Vertex**' to find the first vertex (l_i, l_j). The inputs of this function are (i, j) and starting tracking direction P_k. As we defined earlier, the distance threshold between the input (i, j) and the vertex (l_i, l_j) is set to L. The idea of this function is to look for the next point (called intermediate point (p, q)) pixel by pixel along the fingerprint ridge towards the starting tracking direction until it meets the distance threshold L or reaches the valley. The tracking direction will adjust to the neighboring direction if the searching processing reaches the valley. The output vertex (l_i, l_j) will be located once the search processing terminates. We define eight variables to adjust the tracking direction as listed in Equation A.6-A.13. The initial value of the intermediate point (p, q) is the input (i, j). The details of the function '**Detecting_Vertex**' are described in **Algorithm** A.1.

$$W_1 = \boldsymbol{M}(p-1, q-1) + \boldsymbol{M}(p-1, q) + \boldsymbol{M}(p-1, q+1) \tag{A.6}$$

$$W_2 = \boldsymbol{M}(p-1, q) + \boldsymbol{M}(p-1, q+1) + \boldsymbol{M}(p, q+1) \tag{A.7}$$

$$W_3 = \boldsymbol{M}(p-1, q) + \boldsymbol{M}(p-1, q+1) + \boldsymbol{M}(p, q+1) \tag{A.8}$$

$$W_4 = \boldsymbol{M}(p+1, q) + \boldsymbol{M}(p+1, q+1) + \boldsymbol{M}(p, q+1) \tag{A.9}$$

$$W_5 = \boldsymbol{M}(p+1, q-1) + \boldsymbol{M}(p+1, q) + \boldsymbol{M}(p+1, q+1) \tag{A.10}$$

$$W_6 = \boldsymbol{M}(p, q-1) + \boldsymbol{M}(p+1, q-1) + \boldsymbol{M}(p+1, q) \tag{A.11}$$

$$W_7 = \boldsymbol{M}(p+1, q-1) + \boldsymbol{M}(p, q-1) + \boldsymbol{M}(p-1, q-1) \tag{A.12}$$

$$W_8 = \boldsymbol{M}(p, q-1) + \boldsymbol{M}(p-1, q-1) + \boldsymbol{M}(p-1, q) \tag{A.13}$$

Thirdly, another vertex (r_i, r_j) can be detected by using the same function '**Detecting_Vertex**' with the input point (i, j) and a starting tracking direction P_x which is determined by the following equation.

$$P_x = \begin{cases} 1 & \text{if } p \le i, q > j \\ 2 & \text{if } p < i, q \le j \\ 3 & \text{if } p \ge i, q < j \\ 4 & \text{if } p > i, q \ge j \end{cases}$$

Step 2: Detecting the neighboring triangles

In order to achieve a robust fingerprint orientation, the neighboring triangles are detected based on the central isosceles triangle as mentioned earlier at the beginning of this

Algorithm A.1 Function of detecting an isosceles triangle

Require: the location information of an input point: (i, j);
the initial tracking direction: $P_k, k \in \{1, 2, 3, 4\}$;
Ensure: the location information of the vertex:(l_i, l_j);
1: Define eight variables as listed in Equation A.6-A.13;
2: Define the distance between intermediate point (p, q) with the input point (i, j) as: $dis = sqrt((p-i)^2 + (q-j)^2)$
3: **while** $dis < L$ **do**
4: **if** $P_k == 1$ **then**
5: **if** $W_2 < 3$ **then**
6: **if** $\boldsymbol{M}(p-1, q+1)$==0 **then**
7: Set $p = p-1, q = q+1$;
8: Continue (Go to the beginning of the loop);
9: **end if**
10: **if** $\boldsymbol{M}(p, q+1)$==0 **then**
11: Set $q = q+1$;
12: Continue;
13: **end if**
14: **if** $\boldsymbol{M}(p-1, q)$==0 **then**
15: Set $p = p-1$;
16: Continue;
17: **end if**
18: **end if**
19: **if** $W_1 < 3$ **AND** $W_1 <= W_3$ **then**
20: Set $p = p-1, q = q-1$;
21: Adjust the tracking direction: $P_k = 2$;
22: Continue;
23: **else**
24: Set $p = p+1, q = q+1$;
25: $P_k = 4$;
26: Continue;
27: **end if**
28: **end if**
29: **for** $P_k = 2, 3, 4$ **do**
30: The procedures are similar with the $P_k = 1$;
31: **end for**
32: **end while**

section. We define a middle point (b_i, b_j) whose location is computed by Equation A.14 and A.15.

$$b_i = ceil((l_i + r_i)/2) \tag{A.14}$$

$$b_j = ceil((l_j + l_j)/2) \tag{A.15}$$

We consider the direction from middle point to the input point (i, j) as upward direction, in contrast to the direction from the input point to the middle point as downward direction. And there are T triangles, which will be detected from the upward direction and downward direction respectively. These triangles can be found using the same function '**Detecting_triangle**' with several neighboring points around the input point (i, j) (as shown in Figure A.4). These neighboring points can be easily located by searching the neighboring ridges from the input point (i, j) along the upward direction and downward direction respectively.

Step 3: Calculating the fingerprint orientation

In total, there are $2T + 1$ triangles that have been located after the previous steps. The fingerprint orientation can be derived from the set of values resulting from the individual triangles and thus we obtain a robust measure for the orientation angle. The individual fingerprint orientation of each triangle is calculated by using the following equation.

$$o= \begin{cases} \arctan((j - b_j)/(b_i - i)) + \pi & \text{if } b_i \leq i, b_j > j \\ 2\pi - \arctan((j - b_j)/(i - b_i)) & \text{if } b_i < i, b_j \leq j \\ \arctan((b_j - j)/(i - b_i)) & \text{if } b_i \geq i, b_j < j \\ \pi - \arctan((b_j - j)/(b_i - i)) & \text{if } b_i > i, b_j \geq j \end{cases} \quad \text{(A.16)}$$

We assume the fingerprint orientation of the central triangle is A_0. The fingerprint orientations for the upward triangles are denoted as $A_1, A_2, ..., A_T$, and $A_{T+1}, A_{T+2}, ..., A_{2T}$ indicate the fingerprint orientations of the downward triangles. The fingerprint orientation of the pattern in fingerprint sample I can be computed by discarding the outer percentiles ($i.e.$ ignoring those fingerprint orientations which are far from the average value Avg).

$$Avg = \frac{\sum_{0 \leq c \leq 2T} A_c}{1 + 2T} \quad \text{(A.17)}$$

The absolute differences between each individual fingerprint orientation with Avg is denoted by $Diff_c$ in Equation A.18. If the difference value $Diff_c$ is more than a threshold DT, the corresponding triangle will be discarded for calculating the final fingerprint orientation o_I of the fingerprint pattern. Therefore, the final fingerprint orientation o_I is calculated as in Equation A.19.

$$Diff_c = abs(A_c - Avg), 0 \leq c \leq 2T \quad \text{(A.18)}$$

$$o_I = \frac{\sum_{c=0}^{2T} o_c(o_c \leq DT)}{Z} \quad \text{(A.19)}$$

where Z is the number of o_c that are not exceeding the threshold DT.

Figure A.5: Only 2 out of 80 arch fingerprint samples were extracted singular points by 'Verifinger 6.0 extractor'. The core point for each of the two images is marked by a red cross.

A.3 Experimental set-up and results

A.3.1 Database preparation

80 arch fingerprint samples are selected from a very challenging database CASIA-FIDV5.0 (Fingerprint Image Database Version 5.0) [5]. All of them show strong rotation and a diverse quality. These 80 samples were captured from 16 fingers. Figure A.8 illustrates three

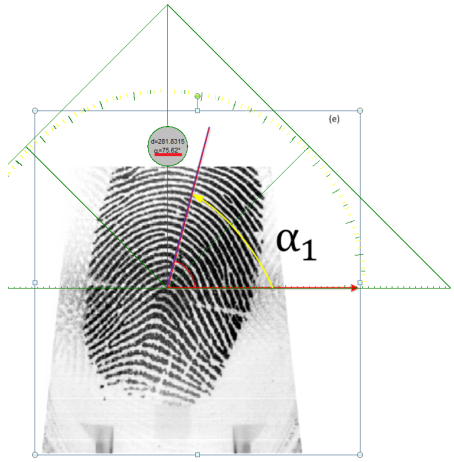


Figure A.6: Generate the ground-truth fingerprint orientation by using a tool 'MB_Ruler'.

partial fingerprint samples, and Figure A.9 shows five samples which were captured from the same source, but with different rotations. Before testing the proposed approach, we used a commercial of the shelve fingerprint feature extractor 'Verifinger 6.0 extractor' to detect the singular points on these 80 arch fingerprint samples. There are only 2 samples for which the commercial algorithm could detect core points. But even these two detected singular points are not accurate, as one easily observe when analyzing Figure A.5.

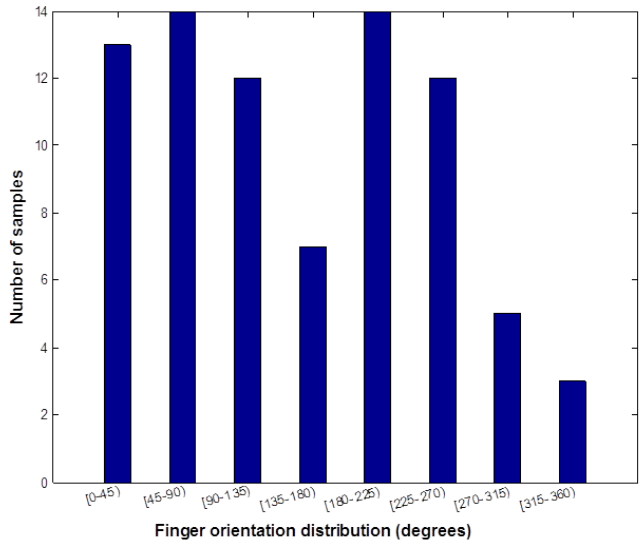


Figure A.7: Fingerprint orientation distribution of database

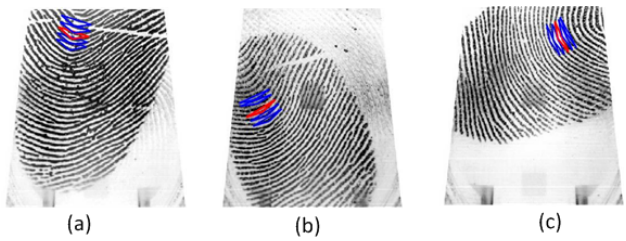


Figure A.8: Partial fingerprint samples with detected triangles. Red triangle denotes the central triangle.

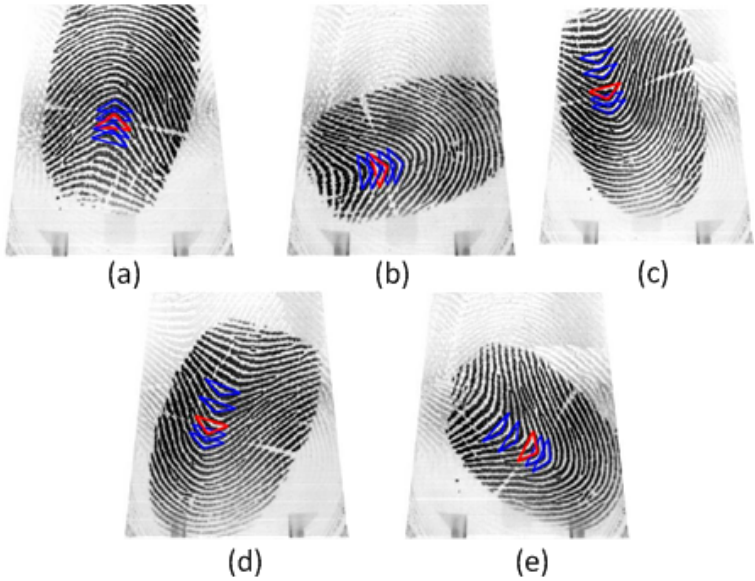


Figure A.9: Fingerprint samples with detected triangles. All five samples are from the same source.

A.3.2 Ground-truth generation

In order to evaluate the performance of our proposed approach, we manually marked up the input points and the fingerprint orientation as ground-truth for our experiments. The input points were determined by human experts visual inspection according to the description in Section A.2. As we introduced earlier, the point with maximum curvature is not required, instead the neighboring points with high curvature also can be used to measure the fingerprint orientation due to the flexibility of proposed approach. The ground-truth fingerprint orientation of the prepared database is obtained by a digital tool called 'MB_Ruler'[18]. 'MB_Ruler' can measure an angle starting at horizontal axis from right to left. This angle just reflect the definition of the fingerprint orientation. Figure A.6 illustrated a sample with ground-truth 75.62 measured by 'MB_Ruler'. Figure A.7 shows the distribution of the ground-truth fingerprint orientation of these 80 arch fingerprint samples.

A.3.3 Performance evaluation

We applied the proposed approach to measure the fingerprint orientation for prepared database. The parameters described in Section A.2 are set as follows: $L = 30, T = 3, DT = 10$. Figure A.9 illustrates five samples with detected triangles, and the central triangle is marked by red color. These five samples are from the same source. Figure A.8 shows that the proposed approach is still able to detected the triangles on partial fingerprint samples. The fingerprint orientation can be calculated by using formula A.19 based on these detected triangles. We computed the difference between these detected fingerprint orientation and ground-truth, and displayed the result in Figure A.10. We can see that there are 93.75% (75 out of 80) samples with angle differences less than 10 degrees. This difference is acceptable, since observation errors during ground-truth generation is also inevitable. Thus the experimental results support the assumption that the proposed approach has the capability of measuring the fingerprint orientation for arch fingerprints, and this fingerprint orientation can be further used to align the fingerprint samples.

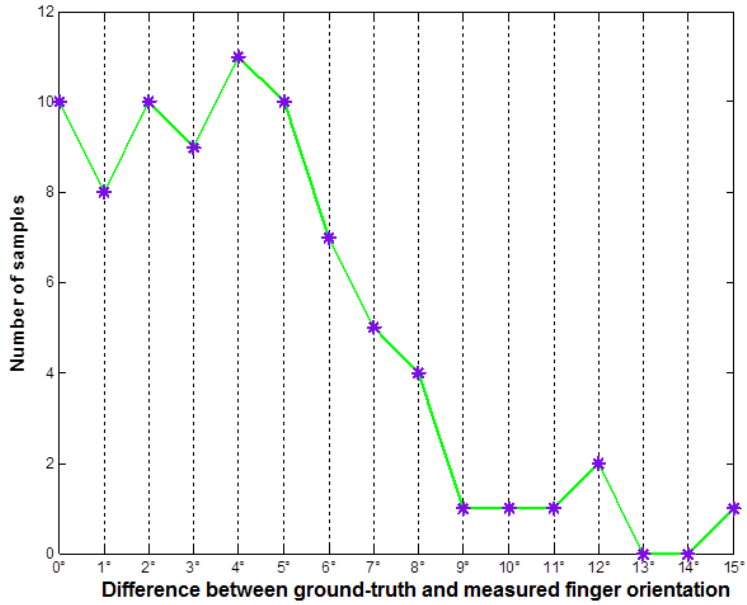


Figure A.10: Differences between ground-truth and measured fingerprint orientation for 80 arch fingerprint samples.

A.4 Conclusion

In this paper, We present an approach to measure the fingerprint orientation for arch fingerprint samples. The proposed approach detects a set of isosceles triangles by using an input point as starting point. Subsequently an orientation value is calculated for each triangle. The fingerprint orientation of the fingerprint sample can be obtained by discarding those orientation values which are far from the average value. The proposed approach has been tested on a very challenging database containing 80 arch fingerprint samples with strong rotation. In order to evaluate the performance, we manually estimate the fingerprint orien-

tation of this database as ground-truth. The results of analysing detected fingerprint orientation with the ground-truth indicates that the proposed approach is capable of measuring the fingerprint orientation for arch fingerprints, which can further be used for fingerprint alignment. Our future work will focus on detecting a more robust input point to be used for the proposed method.

Appendix B

Testing Mobile Phone Camera based Fingerprint Recognition under Real-life Scenarios

Abstract

Fingerprint recognition has been widely used for personal verification in commercial and forensic fields, but its implementation using mobile phones cameras is still an emerging technology. In this paper, we evaluate the feasibility of fingerprints recognition via mobile phone camera under real-life scenarios including (1) in-door with office illumination, (2) natural darkness, and (3) out-door natural illumination with complicated background. We selected three popular smartphones (Nokia N8, iPhone 4, Samsung Galaxy I) to capture fingerprint images. NeuroTechnology and NIST functions (mindtct as minutiae extractor and bozorth3 as minutiae comparator) were adopted to generate ISO standard minutiae templates and compute the comparison scores among different subsets of the generated templates. The subsets are grouped by using different scenarios and different mobile phone cameras. The results of our evaluation indicate that, unlike the in-lab scenario, it is a very challenging task to use mobile phone camera for fingerprint recognition in real life scenarios and thus it is essential to control the image quality during the sample acquisition process.

B.1 Introduction

Nowadays, a majority of smartphones are equipped with a camera with 2 mega pixels or above. The owners of mobile devices can handle their business in multiple locations in a convenient manner and can improve the efficiency of their business tasks by online operation. According to Ciscos report Visual Networking Index (VNI) Global Mobile Data Traffic Update 2011-2016, the number of mobile-connected devices will exceed the population on earth by the end of 2012 [6]. At the same time smartphones known as a personal digital assistant are usually adopted to record users sensitive corporate or personal information, such as social security numbers, credit card numbers, usernames, passwords, etc. However mobile devices can easily be lost or stolen. Claudia Nickel conducted a survey to investigate how people use their mobile devices in her dissertation [1]. The survey shows 12.3% of the participants had their phone lost or stolen once, additional 3.8% even more than once. Such incidents sketch the severity of such a risk. Unfortunately Personal Identification Number (PIN) or passwords are commonly chosen for the authentication of a genuine user. But PIN and passwords may be forgotten or stolen by an individual who observes the owner interacting with his devices and typing the secret number of string. Biometric characteristics cannot be forgotten [101] and thus be suitable as identifiers for personal verification and the subjects need not to memorize them.

Fingerprints are the most common modality in forensic and governmental databases. The US-VISIT program uses fingerprint recognition systems to enforce homeland and border security. Currently, biometric authentication has been globally adopted to implement National ID verification and voter registration methods [173]. Especially for smartphones, it is easy and inexpensive to develop and deploy fingerprint authentication. Motorola has

Table B.1: Specification of cell phone cameras.

Cell phone name	Nokia N8	iPhone 4	Samsung Galaxy S
Mega pixel (max)	12.0	5.0	5.0
Resolution option selected	1536×1936	2592×1936	1600×960
Auto-focus	Yes	Yes	Yes
Image format	JPEG	JPEG	JPEG
ISO control	automatic	automatic	automatic
Flash source	Xenon	LED	no flash
Flash setting	automatic	automatic	no flash
Aperture	f/2.8	f/2.8	f/2.6
Sensor size	1/1.83″	1/3.2″	information not available

unveiled in 2011 the first Android smartphone with a build-in fingerprint reader which allows only the owner to unlock the phone [37]. On the other hand, it could be interesting to use mobile phones camera to capture fingerprints, since an optical camera is already a popular component in a mobile phone nowadays. If such cameras can be exploited for fingerprint capturing, no dedicated fingerprint sensors are needed to squeeze in a phones limited physical space.

A recent research [73] has shown it is feasible to implement fingerprint recognition using mobile phones cameras under the laboratory environment. Other researchers [160, 149] used webcams to capture the fingerprint images under the laboratory environment as well. But how well does it work in real-time scenarios is still of curiosity to us, which is the drive of this work.

In this paper, we show an evaluation of different mobile phones' cameras based fingerprints recognition under various real-life scenarios. Section B.2 introduces our procedures for data collection and off-line performance testing. The evaluation results are described in Section B.3. Section B.4 gives the conclusion.

B.2 Data collection and template comparison settings

B.2.1 Data collection

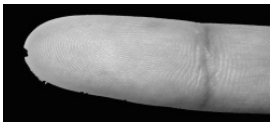
We captured samples from 100 different fingers consisting of 25 groups of right index finger, right middle finger, left index finger and left middle finger. We captured these 4 fingers from a subject because in most cases subjects feel more convenient to take photos on them than on other fingers. Meanwhile, samples captured from index and middle fingers are usually high in quality by empirical observation. We took three photos for each finger. Three smart phones were selected for this experiment - iPhone 4, Samsung Galaxy I, and Nokia N8. Both iPhone 4 and Samsung Galaxy I have a 5 megapixel embedded camera. Nokia N8 features with a 12 megapixel embedded camera (Table B.1). The camera embedded in Samsung Galaxy I is without flash. We refer the Nokia 8, iPhone 4 and Samsung Galaxy I to cam-NOK, cam-IPH and cam-SAM in the remainder of the paper.

We defined three typical scenarios to take finger photos and all of the scenarios correspond to typical convenience use cases of the mobile phone as a pocket device [218]. The first scenario is an in-door scenario with good illumination condition with a desktop as background under an office environment. Figure B.1(a) gives a finger sample taken in this in-door scenario. The second scenario is the natural darkness scenario in which the fingerprint samples are captured in very poor (almost dark) natural illumination such that the

(a) in-door scenario

(b) natural darkness scenario

(c) out-door scenario



(d) cropping image

Figure B.1: Fingerprint samples under different scenarios and cropping pre-processing.

Table B.2: Definition of sessions

Session No.	Interpretation
Session 1	Scenario indoor, Nokia N8 camera
Session 2	Scenario indoor, iPhone 4 camera
Session 3	Scenario indoor, Samsung Galaxy I camera
Session 4	Scenario dark, Nokia N8 camera
Session 5	Scenario outdoor, Nokia N8 camera
Session 6	Scenario outdoor, iPhone 4 camera
Session 7	Scenario outdoor, Samsung Galaxy I camera

flash light has to be activated. We note that in this scenario only Nokia N8 can capture samples with good contrastness while the other two cameras failed to do so at all. This in this scenario only Nokia N8 was employed for testing and the camera flash was automatically turned on in this scenario, Figure B.1(b). The third scenario is the out-door scenario which samples were taken outdoors with complicate background (mainly lawn, lake, houses, and cars in our experiments). Figure B.1(c) shows an example in the third scenario. All of the phones have been used to capture three samples for each finger. In the following sections, scen-IN, scen-DARK, scen-OUT refer to in-door, natural darkness, and out-door scenarios respectively.

In our experiments, 7 sessions, shown in Table B.2, are created for each finger to capture 3 samples. In total there were 2100 (25 subjects $\times$ 4 fingers $\times$ 3 samples $\times$ 7 sessions) fingerprint images captured.

Table B.3: Description of Abbreviations

Abb.	Description
scen-IN	Scenario indoor
scen-DARK	Scenario dark
scen-OUT	Scenario outdoor
cam-NOK	Camera Nokia N8
cam-IPH	Camera iPhone 4
cam-SAM	Camera Samsung Galaxy I
Smp	Sample
Sub	Subject

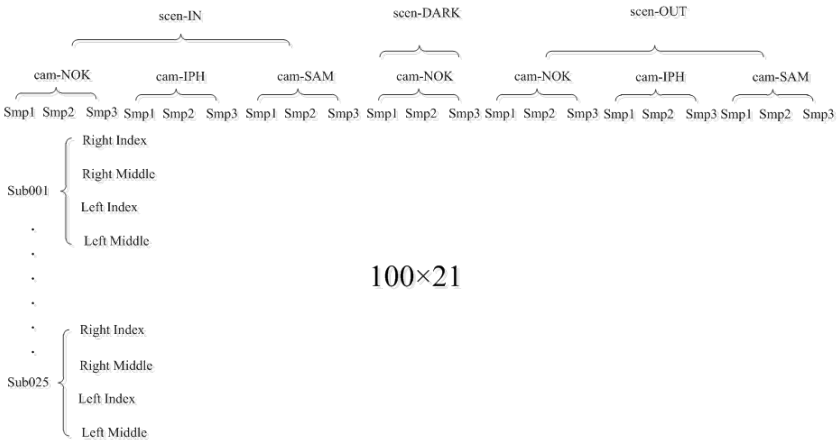
B.2.2 Template comparison settings

Figure B.2(a) gives a matrix composed of all captured samples. The explanation of abbreviations is listed in Table B.3. The columns correspond to 21 captured images (7 sessions 3 samples) for each finger. NeuroTechnology (referred to NT) and NISTs mindtct / bozorth3 have been adopted to extract and compare the minutiae templates from the fingerprint images. Among the three samples captured from each finger, the one with maximum detected minutiae (by NT) is deemed as the good-quality sample. Matrix B (Figure B.2b) is composed by these good-quality samples. Before extracting the template from fingerprint image and comparing the templates, the foreground (finger area) from raw images is cropped manually, assuming in practice such segmentation can be achieved in real time by background subtraction in a video sequence recording the full finger probing process. Figure B.1d shows a fingerprint sample after such a cropping.

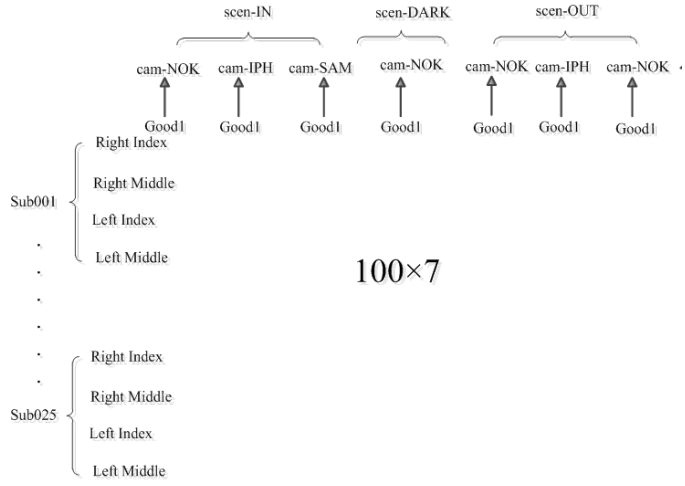
Fingerprint recognition can happen in various scenarios. For instance, a user enrols a finger sample to his/her smartphone at office, and then he/she can verify his/her identity to lock/unlock the screen at office or at home. Comparison experiment I and III are derived from this situation for device access control. In other scenarios we would like to do remote authentication while verification may be done in a supermarket using our smartphone or a third-party's mobile phone. Comparison experiment II and comparison experiment IV are for such application using mobile phone as a wireless terminal for identity authentication. In comparison experiment V, we don't take account of different scenarios and different cameras due to a majority of samples with a low image quality. Genuine scores have been generated by comparing the samples from the same finger. Comparing the samples from different fingers we got the imposter scores. In order to reduce the computational complexity, imposter scores were taken from the sample pair (SmpX, SmpY) that the row index of SmpY is more than the row index of SmpX in matrix B. The details of five comparison experiment methods are described in the following.

Comparison experiment I: intra-camera and intra-scenario. In this case, it compared the samples taken by the same camera in the same scenario. The good-quality samples denoted in Matrix B are selected as reference images. The other two images from the same fingerprint are used to compare to the good-quality sample. Thus the number of genuine scores is 1400. The number of imposter scores is $\sum_{i=1}^{99}(100-i)\times 14$.

Comparison experiment II: inter-camera and intra-scenario. This method compared the samples taken by different cameras in the same scenario. Thus, the images from dark scenario havent been adopted in this case. We only consider the good-quality samples in Matrix B in this experiment. The samples from the cam-IPH in scen-INH are selected as the reference with the probe images from the cam-NOK and cam-SAM in scen-INH. The samples from the cam-SAM in scen-OUT are considered as the reference with the probe images from the cam-NOK and cam-IPH in scen-OUT. The number of genuine scores is



(a) Matrix A composed of all captured images



(b) Matrix B

Figure B.2: Derived matrix using only the best fingerprint samples

Table B.4: Description of Abbreviations

	Reference images	Probe images
1	cam-NOK with scen-DARK	cam-IPH with scen-IN
2	cam-NOK with scen-DARK	cam-SAM with scen-IN
3	cam-NOK with scen-DARK	cam-IPH with scen-OUT
4	cam-NOK with scen-DARK	cam-SAM with scen-OUT
5	cam-IPH with scen-IN	cam-NOK with scen-OUT
6	cam-IPH with scen-IN	cam-SAM with scen-OUT
7	Smp	Sample

400. The number of imposter scores is $\sum_{i=1}^{99}(100-i)\times 4$.

Comparison experiment III: intra-camera and inter-scenario. This method only compared the good-quality samples from the same camera in different scenarios. The samples from the cam-NOK in scen-DAR are considered as the reference with the probe images from the same camera in scen-INH and scen-OUT. The samples from the cam-IPH in scen-INH are considered as the reference with the probe images from the same camera in scen-OUT. The samples from the cam-SAM in scen-OUT are considered as the reference with the probe images from the same camera in scen-OUT. The number of genuine scores is 400 and the number of imposter scores is $\sum_{i=1}^{99}(100-i)\times 4$.

Comparison experiment IV: inter-camera and inter-scenario. This method only compared the good-quality samples taken by different cameras in different scenarios. Table B.4 indicates how to select the reference images and probe images. Thus the number of genuine scores is 700. The number of imposter scores is $\sum_{i=1}^{99}(100-i)\times 7$.

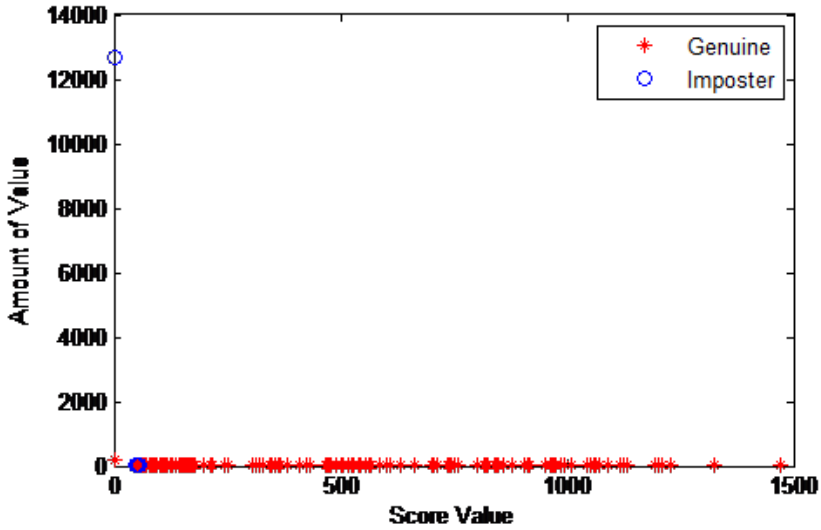
Comparison experiment V: in this experiment, one best-quality sample was manually selected by the operator (mobile phone holder) as a reference from all the 21 samples captured from one finger. This results in 50 fingers with at least 2 samples that can generate minutiae templates by the NeuroTechnology minutiae extractor with sample quality control functionality. This experiment tries to investigate the optimistic accuracy performance.

B.3 Evaluation results

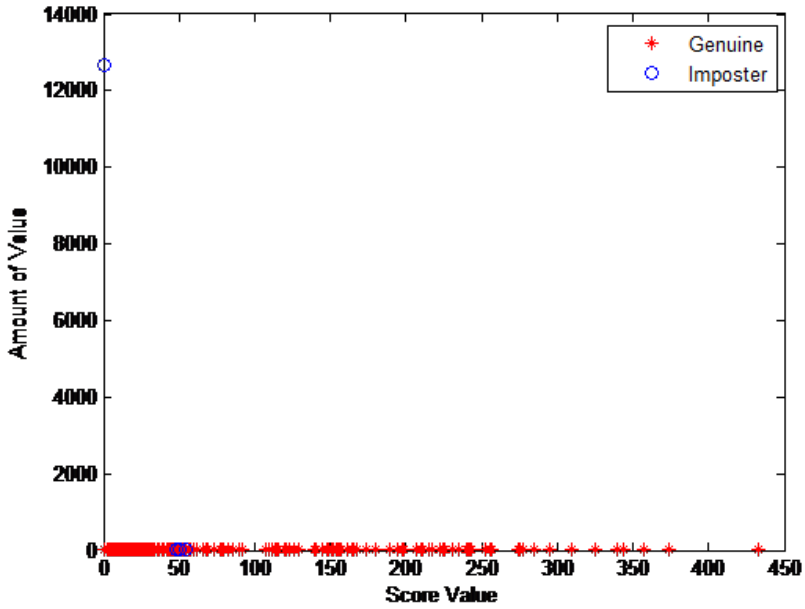
In this paper, we chose the NeuroTechnology and NIST algorithms to calculate the genuine scores and imposter scores. The comparison scores can be benchmarked from three combinations of procedures that are NIST-mindtct minutiae extractor with NIST-bozorth3 comparator, NT extractor with NIST-bozorth3 comparator, and NT extractor with NT comparator.

When benchmarking the algorithm performance of a biometric system, the Equal Error Rate (EER) is usually chosen to compare the accuracy among different algorithms. EER is the rate where False Match Rate (FMR) [101] and False non-Match Rate (FNMR) [101] are equal.

It is different from the laboratory environment that camera and hands are both fixed. It was impossible to avoid hand and camera shaking during taking photos in real-life scenarios. And the cameras usually focused on the background in the outside scenario. It is quite hard to get stable and good quality images. Table B.5 displays the EER value of five comparison experiment methods according to the different comparison experiments.



(a) The statistics of experiment on NT_NT



(b) The statistics of experiment on NT_Nist

Figure B.3: The statistics of score value on comparison experiment V

Table B.5: EER values of five comparison methods.

	NIST_NIST	NT_NIST	NT_NT
Comparison	44.5%	45.4%	41.9%
Comparison	48.7%	49.3%	49.6%
Comparison	47.7%	48.5%	46.9%
Comparison	47.8%	48.5%	49.1%
Comparison	48.3%	25.4%	24.8%

A statistics of genuine scores and imposter scores is showed in Figure B.3 according to the comparison experiment V. X-axis indicates the score value and Y-axis describes the number of each value. In the case for NT extractor and NIST-BOZORTH comparator, the number of imposter scores that are less than 10 is 11567 (in total 12687), and the portion is 91.2%. The amount of zero scores is 12680 and the portion is 99.9% for the NT extractor and NT comparator. Only 7 imposter scores are more than 0.

In general, the value of genuine scores by NIST-bozorth3 comparator is less than the value of NT comparator that operates on the same samples, as seen in Figure B.3. Judged by human eyes, the image quality is getting better along the increase of genuine scores. NeuroTechnology fingerprint template matching algorithm is able to identify fingerprints even if they have only 5-7 similar minutiae [33]. That means NeuroTechnology gets the zero value when only 4 similar minutiae are extracted. That is why the number of zeros of genuine scores generated by NeuroTechnology is much more than the number of zeros of genuine scores generated by NIST-mindtct. Also, NIST-mindtct extractor generates more minutiae than NT extractor usually for the same fingerprint images

B.4 Conclusion and future work

Due to the complicated background, hand and camera shaking during taking photos, the result of experiments indicates it is impossible to get a desirable performance of fingerprint recognition using mobile phone cameras in real-life scenarios even though it might be working well under laboratory environment [73, 160, 149]. Another reason to the undesirable performance reported in this paper could be the fact that existing consumer mobile phone cameras are mostly optimized to capture human face or other more attracting objects in a frame instead of fingerprints. This can be clearly observed from our experiments that the iPhone 4 in general fails to auto focus on the finger area in the outdoor scenario. Therefore, if the application of fingerprint recognition using existing mobile phone camera needs to be feasible in real-life scenarios, it is essential to control the image quality during sample collections. As future work, we will test real-time quality control and enhancement for the raw image generated from the mobile phone cameras, and implements them on mobile phones. Comparisons with build-in fingerprint sensors in mobile phones are also to be tested.

Bibliography

[1] Accelerometer-based biometric gait recognition for authentication on smartphones. http://tuprints.ulb.tu-darmstadt.de/3014/4/20120620_Dissertation_Nickel_final.pdf. Accessed: 2012-07-30.

[2] Autocorrelation. http://www.itl.nist.gov/div898/handbook/eda/section3/eda35c.htm. Accessed: 2015-01-30.

[3] Biometric Recognition Group of the Universidad Autonoma de Madrid. http://atvs.ii.uam.es/. Accessed: 2016-01-30.

[4] Brain wonders. http://www.brainwonders.in/types-of-finger-print.html. Accessed: 2016-01-5.

[5] CASIA-FingerprintV5. http://biometrics.idealtest.org/. Accessed: 2014-01-30.

[6] Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2011-2016. http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.html. Accessed: 2012-02-14.

[7] FBI IAFIS. http://www.fbi.gov/about-us/cjis/fingerprints_biometrics/iafis/iafis. Accessed: 2015-01-30.

[8] FIDO identity authentication. https://fidoalliance.org/. Accessed: 2015-01-30.

[9] Fingerprint enhancement implementation. http://www.csse.uwa.edu.au/~pk/research/matlabfns/. Accessed: 2013-01-30.

[10] Fingerprint image quality: Predicting biometric performance. https://brage.bibsys.no/xmlui/handle/11250/2366306. Accessed: 2016-05-06.

[11] Four-quadrant inverse tangent. http://se.mathworks.com/help/matlab/ref/atan2.html#buct8h0-4. Accessed: 2015-01-30.

[12] FVC2002 database. http://bias.csr.unibo.it/fvc2002/. Accessed: 2014-11-30.

[13] FVC2004 database. http://bias.csr.unibo.it/fvc2004/default.asp. Accessed: 2014-11-30.

[14] Griaule Biometrics. http://www.griaulebiometrics.com/en-us/book/understanding-biometrics/types/segmentation. Accessed: 2016-01-5.

[15] Indian aadhaar (uidai). http://uidai.gov.in/aapka-aadhaar.html. Accessed: 2013-10-30.

[16] Indian UIDAI project official website. http://uidai.gov.in/. Accessed: 2016-01-5.

[17] iPhone fingerprint sensor. http://www.apple.com/iphone-5s/. Accessed: 2015-01-30.

[18] Mb-ruler. http://www.markus-bader.de/MB-Ruler/index.php. Accessed: 2014-01-30.

[19] MCYT100 database. http://atvs.ii.uam.es/index. Accessed: 2014-10-01.

[20] Mobbkey remote secure access control by biometrics and smartphone. http://www.mobbeel.com/products/mobbkey/overview/. Accessed: 2015-01-30.

[21] Mobilepass project. http://www.mobilepass-project.eu/sites/default/files/Towards%20Contactless%20Biometric%20Feature%20Verification%20for%20Mobile%20Border%20Control.pdf. Accessed: 2014-11-30.

[22] Neurotechnology fingerprint sdk. http://www.neurotechnology.com/verifinger.html. Accessed: 2015-01-30.

[23] Neurotechnology verifinger 7.1 matcher. http://download.neurotechnology.com/VeriFinger_SDK_Brochure_2015-06-01.pdf. Accessed: 2016-01-30.

[24] NIST fingerprint image quality 2. http://www.nist.gov/itl/iad/ig/development_nfiq_2.cfm. Accessed: 2016-01-31.

[25] NIST-NBIS function mindtct. http://www.nist.gov/manuscript-publication-search.cfm?pub_id=51097. Accessed: 2012-10-31.

[26] NSTC (National Science and Technology Council). http://www.biometrics.gov/Documents/BioHistory.pdf. Accessed: 2016-01-5.

[27] Philippine Fingerprint Database Breach. http://blog.trendmicro.com/trendlabs-security-intelligence/55m-registered-voters-risk-philippine-commission-elections-hacked/. Accessed: 2016-05-06.

[28] SFinGe (Synthetic Fingerprint Generator. http://biolab.csr.unibo.it/sfinge.html. Accessed: 2016-01-30.

[29] TST BiRD 3. http://www.neurotechnology.com/fingerprint-scanner-tst-biometrics-bird-3.html. Accessed: 2015-01-30.

[30] US Fingerprint Database Breach. https://www.washingtonpost.com/news/the-switch/wp/2015/09/23/opm-now-says-more-than-five-million-fingerprints-compromised-in-breaches/. Accessed: 2016-01-30.

[31] US-VISIT program. http://www.dhs.gov/obim#. Accessed: 2016-01-5.

[32] Verifinger. http://www.neurotechnology.com/verifing-er.html. Accessed: 2013-10-30.

[33] Verifinger algorithm features and capabilities. http://www.neurotechnology.com/verifinger-technology.html#algorithm. Accessed: 2012-07-30.

[34] VIS Date. https://www.sem.admin.ch/sem/en/home/themen/einreise/einfuehrung_vis.html. Accessed: 2016-01-5.

[35] VIS EULISA. http://www.eulisa.europa.eu/Newsroom/News/Pages/VIS-has-processed-20,000,000-visa-applications.aspx. Accessed: 2016-01-5.

[36] VIS Website. http://ec.europa.eu/dgs/home-affairs/what-we-do/policies/borders-and-visas/visa-information-system/index_en.htm. Accessed: 2016-01-5.

[37] Writing fingerprint-enable apps. http://developer.motorola.com/docs/writing-fingerprint-enabled-apps/. Accessed: 2012-07-31.

[38] ABE, N., AND SHINZAKI, T. Vectorized fingerprint representation using minutiae relation code. In *Biometrics (ICB), 2015 International Conference on* (2015), IEEE, pp. 408–415.

[39] ABE, N., YAMADA, S., AND SHINZAKI, T. Irreversible fingerprint template using minutiae relation code with bloom filter. In *Biometrics Theory, Applications and Systems (BTAS), 2015 IEEE 7th International Conference on* (2015), IEEE, pp. 1–7.

[40] ALONSO-FERNANDEZ, F., FIERREZ, J., ORTEGA-GARCIA, J., GONZALEZ-RODRIGUEZ, J., FRONTHALER, H., KOLLREIDER, K., AND BIGUN, J. A comparative study of fingerprint image-quality estimation methods. *Information Forensics and Security, IEEE Transactions on 2*, 4 (2007), 734–743.

[41] ASHBAUGH, D. R. *Quantitative-qualitative friction ridge analysis: an introduction to basic and advanced ridgeology*. CRC press, 1999.

[42] BAI, C., ZHAO, T., WANG, W., AND WU, M. An efficient indexing scheme based on k-plet representation for fingerprint database. In *Intelligent Computing Theories and Methodologies*. Springer, 2015, pp. 247–257.

[43] BANSAL, D., SOFAT, S., AND KAUR, M. Fingerprint fuzzy vault using hadamard transformation. In *Advances in Computing, Communications and Informatics (ICACCI), 2015 International Conference on* (2015), IEEE, pp. 1830–1834.

[44] BAY, H., ESS, A., TUYTELAARS, T., AND VAN GOOL, L. Speeded-up robust features (surf). *Computer vision and image understanding 110*, 3 (2008), 346–359.

[45] BAZEN, A., AND GEREZ, S. Directional field computation for fingerprints based on the principal component analysis of local gradients. In *Proceedings of ProRISC2000* (2000), Citeseer, pp. 215–222.

[46] BAZEN, A. M., AND GEREZ, S. H. Directional field computation for fingerprints based on the principal component analysis of local gradients.

[47] BEBIS, G. Fingerprint indexing. *Encyclopedia of Biometrics* (2015), 643–649.

[48] BEBIS, G., DEACONU, T., AND GEORGIOPOULOS, M. Fingerprint identification using delaunay triangulation. In *Information Intelligence and Systems, 1999. Proceedings. 1999 International Conference on* (1999), IEEE, pp. 452–459.

[49] BHANU, B., AND TAN, X. A triplet based approach for indexing of fingerprint database for identification. In *Audio-and Video-Based Biometric Person Authentication* (2001), Springer, pp. 205–210.

[50] BHANU, B., AND TAN, X. Fingerprint indexing based on novel features of minutiae triplets. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 25*, 5 (2003), 616–622.

[51] BISWAS, S., RATHA, N. K., AGGARWAL, G., AND CONNELL, J. Exploring ridge curvature for fingerprint indexing. In *Biometrics: Theory, Applications and Systems, 2008. BTAS 2008. 2nd IEEE International Conference on* (2008), IEEE, pp. 1–6.

[52] BO, J., PING, T. H., AND LAN, X. M. Fingerprint singular point detection algorithm by poincaré index. *wseas transactions on systems 7*, 12 (2008), 1453–1462.

[53] BRINGER, J., AND DESPIEGEL, V. Binary feature vector fingerprint representation from minutiae vicinities. In *Biometrics: Theory Applications and Systems (BTAS), 2010 Fourth IEEE International Conference on* (2010), IEEE, pp. 1–6.

[54] BRINGER, J., FAVRE, M., PELLE, C., AND DE SAXCÉ, H. Fuzzy vault and template-level fusion applied to a binary fingerprint representation. In *Biometrics Special Interest Group (BIOSIG), 2014 International Conference of the* (2014), IEEE, pp. 1–4.

[55] BRINGER, J., MOREL, C., AND RATHGEB, C. Security analysis of bloom filter-based iris biometric template protection. In *Biometrics (ICB), 2015 International Conference on* (2015), IEEE, pp. 527–534.

[56] BRUNA, J., AND MALLAT, S. Invariant scattering convolution networks. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 35*, 8 (2013), 1872–1886.

[57] BUTT, M., MERKLE, J., KORTE, U., AND BUSCH, C. Correlation-resistant fuzzy vault for fingerprints. In *in Proceedings of the Sicherheit-2016 Conference (Sicherheit)* (2016).

[58] CAPPELLI, R. Fast and accurate fingerprint indexing based on ridge orientation and frequency. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 41*, 6 (2011), 1511–1521.

[59] CAPPELLI, R., AND FERRARA, M. A fingerprint retrieval system based on level-1 and level-2 features. *Expert Systems with Applications 39*, 12 (2012), 10465–10478.

[60] CAPPELLI, R., FERRARA, M., FRANCO, A., AND MALTONI, D. Fingerprint verification competition 2006. *Biometric Technology Today 15*, 7 (2007), 7–9.

[61] CAPPELLI, R., FERRARA, M., AND MALTONI, D. Minutia cylinder-code: A new representation and matching technique for fingerprint recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 32*, 12 (2010), 2128–2141.

[62] CAPPELLI, R., FERRARA, M., AND MALTONI, D. Fingerprint indexing based on minutia cylinder-code. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 33*, 5 (2011), 1051–1057.

[63] CAPPELLI, R., FERRARA, M., AND MALTONI, D. Large-scale fingerprint identification on gpu. *Information Sciences 306* (2015), 1–20.

[64] CAPPELLI, R., LUMINI, A., MAIO, D., AND MALTONI, D. Fingerprint classification by directional image partitioning. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 21*, 5 (1999), 402–421.

[65] CAPPELLI, R., MAIO, D., MALTONI, D., AND NANNI, L. A two-stage fingerprint classification system. In *Proceedings of the 2003 ACM SIGMM workshop on Biometrics methods and applications* (2003), ACM, pp. 95–99.

[66] CHEN, Y., DASS, S. C., AND JAIN, A. K. Fingerprint quality indices for predicting authentication performance. In *Audio-and Video-Based Biometric Person Authentication* (2005), Springer, pp. 160–170.

[67] CHIKKERUR, S., RATHA, N. K., CONNELL, J. H., AND BOLLE, R. M. Generating registration-free cancelable fingerprint templates. In *Biometrics: Theory, Applications and Systems, 2008. BTAS 2008. 2nd IEEE International Conference on* (2008), IEEE, pp. 1–6.

[68] CHOI, K., LEE, D., LEE, S., AND KIM, J. An improved fingerprint indexing algorithm based on the triplet approach. In *Audio-and Video-Based Biometric Person Authentication* (2003), Springer, pp. 584–591.

[69] CHRISTIAN RATHGEB, FRANK BREITINGER, C. B., AND BAIER, H. On the application of Bloom filters to iris biometrics. *IET Biometrics* (2014).

[70] DASS, S. C., AND JAIN, A. K. Fingerprint classification using orientation field flow curves. In *ICVGIP* (2004), pp. 650–655.

[71] DAVIES, S. G. Touching big brother: How biometric technology will fuse flesh and machine. *Information Technology & People 7*, 4 (1994), 38–47.

[72] DE BOER, J., BAZEN, A. M., AND GEREZ, S. H. Indexing fingerprint databases based on multiple features.

[73] DERAWI, M., YANG, B., AND BUSCH, C. Fingerprint recognition with embedded cameras on mobile phones. *Security and Privacy in Mobile Information and Communication Systems* (2011), 136–147.

[74] E. TABASSI, C. L. W., AND WATSON, C. I. NIST-IR 7151 - fingerprint image quality. In *Technical report, NIST* (2004).

[75] ELHAM TABASSI, C. L., AND WATSON, C. I. Nist fingerprint image quality. 9.

[76] FENG, J., AND CAI, A. Fingerprint indexing using ridge invariants. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on* (2006), vol. 4, IEEE, pp. 433–436.

[77] FENG, J., AND JAIN, A. K. Filtering large fingerprint database for latent matching. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on* (2008), IEEE, pp. 1–4.

[78] FENG, J., AND JAIN, A. K. Fingerprint reconstruction: from minutiae to phase. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 33*, 2 (2011), 209–223.

[79] FERRARA, M., MALTONI, D., AND CAPPELLI, R. Noninvertible minutia cylinder-code representation. *Information Forensics and Security, IEEE Transactions on 7*, 6 (2012), 1727–1737.

[80] FERRARA, M., MALTONI, D., AND CAPPELLI, R. A simple and effective two-factor protection scheme for mcc fingerprint templates. *Biometric System Laboratory-University of Bologna, Technical Report* (2014).

[81] GAGO-ALONSO, A., HERNÁNDEZ-PALANCAR, J., RODRÍGUEZ-REINA, E., AND MUÑOZ-BRISEÑO, A. Indexing and retrieving in fingerprint databases under structural distortions. *Expert Systems with Applications 40*, 8 (2013), 2858–2871.

[82] GARRIS, M. D., TABASSI, E., AND WILSON, C. L. NIST fingerprint evaluations and developments. *Proceedings of the IEEE 94*, 11 (2006), 1915–1926.

[83] GHANY, K. K. A., HEFNY, H. A., HASSANIEN, A. E., AND TOLBA, M. F. Kekre's transform for protecting fingerprint template. In *Hybrid Intelligent Systems (HIS), 2013 13th International Conference on* (2013), IEEE, pp. 185–190.

[84] GOMEZ-BARRERO, M., RATHGEB, C., GALBALLY, J., FIERREZ, J., AND BUSCH, C. Protected facial biometric templates based on local gabor patterns and adaptive Bloom filters.

[85] GROTHER, P., AND TABASSI, E. Performance of biometric quality measures. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 29*, 4 (2007), 531–543.

[86] GYAOUROVA, A., AND ROSS, A. A novel coding scheme for indexing fingerprint patterns. In *Structural, Syntactic, and Statistical Pattern Recognition*. Springer, 2008, pp. 755–764.

[87] GYAOUROVA, A., AND ROSS, A. Index codes for multibiometric pattern retrieval. *Information Forensics and Security, IEEE Transactions on 7*, 2 (2012), 518–529.

[88] HARTLOFF, J., DOBLER, J., TULYAKOV, S., RUDRA, A., AND GOVINDARAJU, V. Towards fingerprints as strings: Secure indexing for fingerprint matching. In *Biometrics (ICB), 2013 International Conference on* (2013), IEEE, pp. 1–6.

[89] HE, S., ZHANG, C., AND HAO, P. Clustering-based descriptors for fingerprint indexing and fast retrieval. In *Computer Vision–ACCV 2009*. Springer, 2009, pp. 354–363.

[90] HERMANS, J., MENNINK, B., AND PEETERS, R. When a bloom filter is a doom filter: Security assessment of a novel iris biometric template protection system. In *Biometrics Special Interest Group (BIOSIG), 2014 International Conference of the* (2014), IEEE, pp. 1–6.

[91] HERNÁNDEZ-PALANCAR, J., AND MUÑOZ-BRISEÑO, A. A new fingerprint indexing algorithm for latent and non-latent impressions identification. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*. Springer, 2015, pp. 127–134.

[92] HIEW, B., TEOH, A. B., AND PANG, Y. Digital camera based fingerprint recognition. In *Telecommunications and Malaysia International Conference on Communications, 2007. ICT-MICC 2007. IEEE International Conference on* (2007), IEEE, pp. 676–681.

[93] HONG, L., WAN, Y., AND JAIN, A. Fingerprint image enhancement: algorithm and performance evaluation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 20*, 8 (1998), 777–789.

[94] ICAO. 2006. Machine readable travel documents,*ICAO Technical report*.

[95] ILOANUSI, O., GYAOUROVA, A., AND ROSS, A. Indexing fingerprints using minutiae quadruplets. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference on* (2011), IEEE, pp. 127–133.

[96] ILOANUSI, O. N. Fusion of finger types for fingerprint indexing using minutiae quadruplets. *Pattern Recognition Letters 38* (2014), 8–14.

[97] ISO/IEC 19794-1:2011. Information technology–biometric data interchange formats–part 1: Framework.

[98] ISO/IEC 19794-1:2011. Information technology–biometric sample quality–part 1: Framewrok.

[99] ISO/IEC 19794-2:2011. Information Technology - Biometric Data Interchange Formats - Part 2: Finger Minutiae Data.

[100] ISO/IEC 19794-4:2005. Information technology–biometric data interchange formats–part 4: Finger image data.

[101] ISO/IEC 19795-1:2006. Information technology – Biometric performance testing and reporting – Part 1: Principles and framework.

[102] ISO/IEC 2382-37:2012. Information technology–vocabulary–part 37: Biometrics. *Tech.rep. JTC 1/SC 37/WG 1* (2012).

[103] ISO/IEC 24745:2011. Information Technology - Security Techniques– Biometric Information Protection.

[104] ISO/IEC 29794-4:2010. Information technology–biometric sample quality–part 4: Finger image data.

[105] JAIN, A. K., CHEN, Y., AND DEMIRKUS, M. Pores and ridges: high-resolution fingerprint matching using level 3 features. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 29*, 1 (2007), 15–27.

[106] JAIN, A. K., FLYNN, P., AND ROSS, A. A. *Handbook of biometrics*. Springer Science & Business Media, 2007.

[107] JAIN, A. K., PRABHAKAR, S., HONG, L., AND PANKANTI, S. Filterbank-based fingerprint matching. *Image Processing, IEEE Transactions on 9*, 5 (2000), 846–859.

[108] JAIN, A. K., ROSS, A., AND PRABHAKAR, S. An introduction to biometric recognition. *Circuits and Systems for Video Technology, IEEE Transactions on 14*, 1 (2004), 4–20.

[109] JIA, X., YANG, X., ZANG, Y., ZHANG, N., AND TIAN, J. A cross-device matching fingerprint database from multi-type sensors. In *Pattern Recognition (ICPR), 2012 21st International Conference on* (2012), IEEE, pp. 3001–3004.

[110] JIANG, X., LIU, M., AND KOT, A. C. Fingerprint retrieval for identification. *Information Forensics and Security, IEEE Transactions on 1*, 4 (2006), 532–542.

[111] JIN, Z., LIM, M.-H., TEOH, A. B. J., AND GOI, B.-M. A non-invertible randomized graph-based hamming embedding for generating cancelable fingerprint template. *Pattern Recognition Letters 42* (2014), 137–147.

[112] JIN, Z., TEOH, A. B. J., ONG, T. S., AND TEE, C. Fingerprint template protection with minutiae-based bit-string for security and privacy preserving. *Expert Systems with Applications 39*, 6 (2012), 6157–6167.

[113] JONIETZ, C., MONARI, E., WIDAK, H., AND QU, C. Towards mobile and touchless fingerprint verification. In *Advanced Video and Signal Based Surveillance (AVSS), 2015 12th IEEE International Conference on* (2015), IEEE, pp. 1–6.

[114] JUELS, A., AND SUDAN, M. A fuzzy vault scheme. *Designs, Codes and Cryptography 38*, 2 (2006), 237–257.

[115] KAIZHI, C., AND AIQUN, H. An enhancing fingerprint template protection method. In *Computational Intelligence and Communication Networks (CICN), 2013 5th International Conference on* (2013), IEEE, pp. 275–279.

[116] KARU, K., AND JAIN, A. K. Fingerprint classification. *Pattern recognition 29*, 3 (1996), 389–404.

[117] KHALIL, M. S. Reference point detection for camera-based fingerprint image based on wavelet transformation. *Biomedical engineering online 14*, 1 (2015), 40.

[118] KOO, W. M., AND KOT, A. Curvature-based singular points detection. In *Audio-and Video-Based Biometric Person Authentication* (2001), Springer, pp. 229–234.

[119] KRASNJAK, D., AND KRIVEC, V. Fingerprint classification using a homogeneity structure of fingerprint's orientation field and neural net. In *Image and Signal Processing and Analysis, 2005. ISPA 2005. Proceedings of the 4th International Symposium on* (2005), IEEE, pp. 7–11.

[120] KUMAR, D. G., SUDHA, G., AND REVATHI, B. An efficient space partitioning tree approach for indexing and retrieving fingerprint databases. In *Proceedings of the 2015 International Conference on Advanced Research in Computer Science Engineering & Technology (ICARCSET 2015)* (2015), ACM, p. 50.

[121] KURNIAWAN, F., KHALIL, M. S., AND KHAN, M. K. Core-point detection on camera-based fingerprint image. In *Biometrics and Security Technologies (ISBAST), 2013 International Symposium on* (2013), IEEE, pp. 241–246.

[122] LEE, C., CHOI, J.-Y., TOH, K.-A., AND LEE, S. Alignment-free cancelable fingerprint templates based on local minutiae information. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 37*, 4 (2007), 980–992.

[123] LEE, C., LEE, S., AND KIM, J. A study of touchless fingerprint recognition system. In *Structural, Syntactic, and Statistical Pattern Recognition*. Springer, 2006, pp. 358–365.

[124] LEE, C., LEE, S., KIM, J., AND KIM, S.-J. Preprocessing of a fingerprint image captured with a mobile camera. In *Advances in Biometrics*. Springer, 2006, pp. 348–355.

[125] LEE, H. C., RAMOTOWSKI, R., AND GAENSSLEN, R. *Advances in fingerprint technology*. CRC press, 2001.

[126] LEUNG, K., AND LEUNG, C. H. Improvement of fingerprint retrieval by a statistical classifier. *Information Forensics and Security, IEEE Transactions on 6*, 1 (2011), 59–69.

[127] LI, G., BUSCH, C., AND YANG, B. A novel approach used for measuring fingerprint orientation of arch fingerprint. In *Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2014 37th International Convention on* (2014), IEEE, pp. 1309–1314.

[128] LI, G., YANG, B., AND BUSCH, C. Qualifying fingerprint samples captured by smartphone cameras in real-life scenarios. https://brage.bibsys.no/xmlui/handle/11250/2388306. ISBN: 978-82-8340-040-3.

[129] LI, G., YANG, B., AND BUSCH, C. A score-level fusion fingerprint indexing approach based on minutiae vicinity and minutia cylinder-code. In *Biometrics and Forensics (IWBF), 2014 International Workshop on* (2014), IEEE, pp. 1–6.

[130] LI, G., YANG, B., AND BUSCH, C. A fingerprint indexing scheme with robustness against sample translation and rotation. In *Biometrics Special Interest Group (BIOSIG), 2015 International Conference of the* (2015), IEEE, pp. 1–8.

[131] LI, G., YANG, B., AND BUSCH, C. A novel fingerprint indexing approach focusing on minutia location and direction. In *Identity, Security and Behavior Analysis (ISBA), 2015 IEEE International Conference on* (2015), IEEE, pp. 1–6.

[132] LI, G., YANG, B., OLSEN, M. A., AND BUSCH, C. Quality assessment for fingerprints collected by smartphone cameras. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on* (2013), IEEE, pp. 146–153.

[133] LI, G., YANG, B., RAGHAVENDRA, R., AND BUSCH, C. Testing mobile phone camera based fingerprint recognition under real-life scenarios. *NISK 2012* (2012).

[134] LI, G., YANG, B., RATHGEB, C., AND BUSCH, C. Towards generating protected fingerprint templates based on bloom filters. In *Biometrics and Forensics (IWBF), 2015 International Workshop on* (2015), IEEE, pp. 1–6.

[135] LI, J., YAU, W.-Y., AND WANG, H. Fingerprint indexing based on symmetrical measurement. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on* (2006), vol. 1, IEEE, pp. 1038–1041.

[136] LI, J., YAU, W.-Y., AND WANG, H. Fingerprint indexing based on symmetrical measurement. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on* (2006), vol. 1, IEEE, pp. 1038–1041.

[137] LIANG, X., ASANO, T., AND BISHNU, A. Distorted fingerprint indexing using minutia detail and delaunay triangle. In *Voronoi Diagrams in Science and Engineering, 2006. ISVD'06. 3rd International Symposium on* (2006), IEEE, pp. 217–223.

[138] LIANG, X., BISHNU, A., AND ASANO, T. A robust fingerprint indexing scheme using minutia neighborhood structure and low-order delaunay triangles. *Information Forensics and Security, IEEE Transactions on 2*, 4 (2007), 721–733.

[139] LIU, M., JIANG, X., AND KOT, A. C. Fingerprint reference-point detection. *EURASIP J. Adv. Sig. Proc. 2005*, 4 (2005), 498–509.

[140] LIU, M., JIANG, X., AND KOT, A. C. Fingerprint retrieval by complex filter responses. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on* (2006), vol. 1, IEEE, pp. 1042–1042.

[141] LIU, M., AND YAP, P.-T. Invariant representation of orientation fields for fingerprint indexing. *Pattern Recognition 45*, 7 (2012), 2532–2542.

[142] LIU, T., ZHANG, C., AND HAO, P. Fingerprint reference point detection based on local axial symmetry. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on* (2006), vol. 1, IEEE, pp. 1050–1053.

[143] MALTONI, D., MAIO, D., JAIN, A. K., AND PRABHAKAR, S. *Handbook of fingerprint recognition*. springer, 2009.

[144] MARTIN A. OLSEN, J. W., AND BUSCH, C. Finger image quality based on singular point localization. SPIE.

[145] MIHAILESCU, P. The fuzzy vault for fingerprints is vulnerable to brute force attack. *arXiv preprint arXiv:0708.2974* (2007).

[146] MIRMOHAMADSADEGHI, L., AND DRYGAJLO, A. A template privacy protection scheme for fingerprint minutiae descriptors. In *Biometrics Special Interest Group (BIOSIG), 2013 International Conference of the* (2013), IEEE, pp. 1–8.

[147] MOHAMMADI, S., AND FARAJZADEH, A. Fingerprint reference point detection using orientation field and curvature measurements. In *Intelligent Computing and Intelligent Systems, 2009. ICIS 2009. IEEE International Conference on* (2009), vol. 4, IEEE.

[148] MOUJAHDI, C., BEBIS, G., GHOUZALI, S., AND RZIZA, M. Fingerprint shell: Secure representation of fingerprint template. *Pattern Recognition Letters 45* (2014), 189–196.

[149] MUELLER, R., AND SANCHEZ-REILLO, R. An approach to biometric identity management using low cost equipment. In *Intelligent Information Hiding and Multimedia Signal Processing, 2009. IIH-MSP'09* (2009), pp. 1096–1100.

[150] MUKHERJEE, R. *Indexing techniques for fingerprint and iris databases*. ProQuest, 2007.

[151] MUÑOZ-BRISEÑO, A., GAGO-ALONSO, A., AND HERNÁNDEZ-PALANCAR, J. Fingerprint indexing with bad quality areas. *Expert Systems with Applications 40*, 5 (2013), 1839–1846.

[152] MUNOZ-BRISENO, A., GAGO-ALONSO, A., AND HERNÁNDEZ-PALANCAR, J. Using reference point as feature for fingerprint indexing. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*. Springer, 2014, pp. 367–374.

[153] MYERS, JEROME L.;WELL, A. D. Research design and statistical analysis (2nd ed.), 2003.

[154] NANDAKUMAR, K., AND JAIN, A. K. Multibiometric template security using fuzzy vault. In *Biometrics: Theory, Applications and Systems, 2008. BTAS 2008. 2nd IEEE International Conference on* (2008), IEEE, pp. 1–6.

[155] NANDAKUMAR, K., AND JAIN, A. K. Biometric template protection: Bridging the performance gap between theory and practice. *Signal Processing Magazine, IEEE 32*, 5 (2015), 88–100.

[156] NANDAKUMAR, K., JAIN, A. K., AND PANKANTI, S. Fingerprint-based fuzzy vault: Implementation and performance. *Information Forensics and Security, IEEE Transactions on 2*, 4 (2007), 744–757.

[157] OLSEN, M., XU, H., AND BUSCH, C. Gabor filters as candidate quality measure for NFIQ 2.0. In *Biometrics (ICB), 2012 5th IAPR International Conference on* (2012), IEEE, pp. 158–163.

[158] OLSEN, M. A., ŠMIDA, V., AND BUSCH, C. Finger image quality assessment features–definitions and evaluation. *IET Biometrics* (2015).

[159] PHROMSUTHIRAK, K., AND AREEKUL, V. Fingerprint quality assessment using frequency and orientation subbands of block-based fourier transform. In *Biometrics (ICB), 2013 International Conference on* (2013), IEEE, pp. 1–7.

[160] PIURI, V., AND SCOTTI, F. Fingerprint biometrics via low-cost sensors and webcams. In *Biometrics: Theory, Applications and Systems, 2008. BTAS 2008. 2nd IEEE International Conference on* (2008), IEEE, pp. 1–6.

[161] POONGUZHALI, N., AND EZHILARASAN, M. An indexing technique based on feature level fusion of fingerprint features. *International Journal on Signal and Image Processing 4*, 3 (2013), 34.

[162] POONGUZHALI, N., AND EZHILARASAN, M. A novel fingerprint indexing technique based on level-1 and level-2 features.

[163] PRASAD, M. V., ANUGU, J. R., AND RAO, C. Fingerprint template protection using multiple spiral curves. In *Proceedings of 3rd International Conference on Advanced Computing, Networking and Informatics* (2016), Springer, pp. 593–601.

[164] PRASAD, M. V., AND KUMAR, C. S. Fingerprint template protection using multiline neighboring relation. *Expert Systems with Applications 41*, 14 (2014), 6114–6122.

[165] QUINN, G. W., AND GROTHER, P. The one-to-many multi-modal fusion challenge. In *Biometrics (ICB), 2012 5th IAPR International Conference on*.

[166] RAM, S., BISCHOF, H., AND BIRCHBAUER, J. Curvature preserving fingerprint ridge orientation smoothing using legendre polynomials. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on* (2008), IEEE, pp. 1–8.

[167] RAMALHO, M., CORREIA, P. L., SANTOS, T. M., AND SOARES, L. D. Feature vector binarization: A quantization-based approach to a secure biometric system.

[168] RATHA, N. K., CHIKKERUR, S., CONNELL, J. H., AND BOLLE, R. M. Generating cancelable fingerprint templates. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 29*, 4 (2007), 561–572.

[169] RATHA, N. K., CONNELL, J. H., AND BOLLE, R. M. Enhancing security and privacy in biometrics-based authentication systems. *IBM systems Journal 40*, 3 (2001), 614–634.

[170] RATHA, N. K., KARU, K., CHEN, S., AND JAIN, A. K. A real-time matching system for large fingerprint databases. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 18*, 8 (1996), 799–813.

[171] RATHGEB, C., AND UHL, A. A survey on biometric cryptosystems and cancelable biometrics. *EURASIP Journal on Information Security 2011*, 1 (2011), 1–25.

[172] ROSS, A., AND MUKHERJEE, R. Augmenting ridge curves with minutiae triplets for fingerprint indexing. In *Proceedings of SPIE Conference on Biometric Technology for Human Identification IV* (2007), vol. 6539, p. 65390C.

[173] RUUD, M., JONATHAN, H., SHARATH, P., NALINI, K., AND ANDREW, W. Guide to biometrics, 2004.

[174] SANDHYA, M., AND PRASAD, M. V. k-nearest neighborhood structure (k-nns) based alignment-free method for fingerprint template protection. In *Biometrics (ICB), 2015 International Conference on* (2015), IEEE, pp. 386–393.

[175] SANDHYA, M., AND PRASAD, M. V. Cancelable fingerprint cryptosystem based on convolution coding. In *Advances in Signal Processing and Intelligent Recognition Systems*. Springer, 2016, pp. 145–157.

[176] SANDHYA, M., PRASAD, M. V., AND CHILLARIGE, R. R. Generating cancellable fingerprint templates based on delaunay triangle feature set construction. *IET Biometrics* (2015).

[177] SANKARAN, A., MALHOTRA, A., MITTAL, A., VATSA, M., AND SINGH, R. On smartphone camera based fingerphoto authentication. In *Biometrics Theory, Applications and Systems (BTAS), 2015 IEEE 7th International Conference on* (2015), IEEE, pp. 1–7.

[178] SANKARAN, A., VATSA, M., AND SINGH, R. Multisensor optical and latent fingerprint database. *Access, IEEE 3* (2015), 653–665.

[179] SANTOS, T., LOURENÇO, G., SOARES, L. D., AND CORREIA, P. L. Enhancing biometrics security.

[180] SHAKED, D., AND TASTL, I. Sharpness measure: Towards automatic image enhancement. In *Image Processing, 2005. ICIP 2005. IEEE International Conference on* (2005), vol. 1, IEEE, pp. I–937.

[181] SHEN, L., KOT, A., AND KOO, W. Quality measures of fingerprint images. In *Audio- and Video-based Biometric Person Authentication* (2001), Springer, pp. 266–271.

[182] SHUAI, X., ZHANG, C., AND HAO, P. Fingerprint indexing based on composite set of reduced sift features. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on* (2008), IEEE.

[183] SINGH, R., VATSA, M., AND NOORE, A. Fingerprint indexing using minutiae and pore features. In *IPCV* (2009), pp. 870–875.

[184] SRINIVASAN, V., AND MURTHY, N. Detection of singular points in fingerprint images. *Pattern Recognition 25*, 2 (1992), 139–153.

[185] STEIN, C., BOUATOU, V., AND BUSCH, C. Video-based fingerphoto recognition with anti-spoofing techniques with smartphone cameras. In *Biometrics Special Interest Group (BIOSIG), 2013 International Conference of the* (2013), IEEE, pp. 1–12.

[186] STEIN, C., NICKEL, C., AND BUSCH, C. Fingerphoto recognition with smartphone cameras. In *Biometrics Special Interest Group (BIOSIG), 2012 BIOSIG-Proceedings of the International Conference of the* (2012), IEEE, pp. 1–12.

[187] TABASSI, E., WILSON, C., AND WATSON, C. Nist fingerprint image quality. *NIST Res. Rep. NISTIR7151* (2004).

[188] TABASSI, E., AND WILSON, C. L. A novel approach to fingerprint image quality. In *Image Processing, 2005. ICIP 2005. IEEE International Conference on* (2005), vol. 2, IEEE, pp. II–37.

[189] TAMS, B., MERKLE, J., RATHGEB, C., WAGNER, J., KORTE, U., AND BUSCH, C. Improved fuzzy vault scheme for alignment-free fingerprint features. In *Biometrics Special Interest Group (BIOSIG), 2015 International Conference of the* (2015), IEEE, pp. 1–12.

[190] TEOH, A., AND YUANG, C. T. Cancelable biometrics realization with multispace random projections. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 37*, 5 (2007), 1096–1106.

[191] TEOH, A. B. J., AND KIM, J. Secure biometric template protection in fuzzy commitment scheme. *IEICE Electronics Express 4*, 23 (2007), 724–730.

[192] TIWARI, K., AND GUPTA, P. A touch-less fingerphoto recognition system for mobile hand-held devices. In *Biometrics (ICB), 2015 International Conference on* (2015), IEEE, pp. 151–156.

[193] TOMAZ, F., CANDEIAS, T., AND SHAHBAZKIA, H. Fast and accurate skin segmentation in color images. In *Computer and Robot Vision, 2004. Proceedings. First Canadian Conference on* (2004), IEEE, pp. 180–187.

[194] TRAURING, M. Automatic comparison of finger-ridge patterns. *Nature 197* (1963), 938–940.

[195] TSAI, W.-H., AND MA, C.-H. Fingerprint clustering for forensic data indexing. In *Computer Software and Applications Conference Workshops (COMPSACW), 2013 IEEE 37th Annual* (2013), IEEE, pp. 5–10.

[196] TULYAKOV, S., FAROOQ, F., AND GOVINDARAJU, V. Symmetric hash functions for fingerprint minutiae. In *Pattern Recognition and Image Analysis*. Springer, 2005, pp. 30–38.

[197] UMARANI, J., VISWANATHAN, J., GUPTA, A. K., AND GUPTA, P. Minutiae based geometric hashing for fingerprint database. In *Emerging Intelligent Computing Technology and Applications*. 2012, pp. 422–427.

[198] VAN, T. H., AND LE, H. T. An efficient algorithm for fingerprint reference-point detection. In *Computing and Communication Technologies, 2009. RIVF'09. International Conference on* (2009), IEEE.

[199] VIJ, A., AND NAMBOODIRI, A. Fingerprint indexing based on local arrangements of minutiae neighborhoods. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on* (2012), IEEE, pp. 71–76.

[200] WANG, S., AND HU, J. Alignment-free cancelable fingerprint template design: A densely infinite-to-one mapping (ditom) approach. *Pattern Recognition 45*, 12 (2012), 4129–4137.

[201] WANG, Y., HU, J., AND PHILLIPS, D. A fingerprint orientation model based on 2d fourier expansion (fomfe) and its application to singular-point detection and fingerprint indexing. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 29*, 4 (2007), 573–585.

[202] WANG, Y., YUEN, P. C., AND CHEUNG, Y.-M. Hashing fingerprints for identity de-duplication. In *Information Forensics and Security (WIFS), 2013 IEEE International Workshop on* (2013), IEEE, pp. 49–54.

[203] WANG, Y. A., AND HU, J. Global ridge orientation modeling for partial fingerprint identification. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 33*, 1 (2011), 72–87.

[204] WENG, D., YIN, Y., AND YANG, D. Singular points detection based on multiresolution in fingerprint images. *Neurocomputing 74*, 17 (2011), 3376–3388.

[205] WILSON, C., CANDELA, G., AND WATSON, C. Neural network fingerprint classification. *Journal of Artificial Neural Networks 1*, 2 (1994), 203–228.

[206] WONG, W. J., TEOH, A. B., KHO, Y. H., AND WONG, M. D. Kernel pca enabled bit-string representation for minutiae-based cancellable fingerprint template. *Pattern Recognition 51* (2016), 197–208.

[207] WONG, W. J., TEOH, A. B., WONG, M. D., AND KHO, Y. H. Enhanced multi-line code for minutiae-based fingerprint template protection. *Pattern Recognition Letters 34*, 11 (2013), 1221–1229.

[208] WONG, W.-J., WONG, M.-L. D., AND KHO, Y.-H. Multi-line code: A low complexity revocable fingerprint template for cancelable biometrics. *Journal of Central South University 20*, 5 (2013), 1292–1297.

[209] WOODWARD, J. D., ORLANS, N. M., AND HIGGINS, P. T. *Biometrics:[identity assurance in the information age]*. McGraw-Hill/Osborne New York, 2003.

[210] XIE, S. J., YANG, J., YOON, S., PARK, D., AND SHIN, J. Fingerprint quality analysis and estimation for fingerprint matching. *State of the art in Biometrics. Intech, Vienna. ISBN* (2011), 978–953.

[211] XIE, S. J., YOON, S., SHIN, J., AND PARK, D. S. Effective fingerprint quality estimation for diverse capture sensors. *Sensors 10*, 9 (2010), 7896–7912.

[212] XU, J., AND HU, J. Multi-constrained orientation field modeling and its application for fingerprint indexing. In *Network and System Security*. Springer, 2015, pp. 176–187.

[213] YANG, B., AND BUSCH, C. Parameterized geometric alignment for minutiae-based fingerprint template protection. In *Biometrics: Theory, Applications, and Systems, 2009. BTAS'09. IEEE 3rd International Conference on* (2009), IEEE, pp. 1–6.

[214] YANG, B., BUSCH, C., BOURS, P., AND GAFUROV, D. Robust minutiae hash for fingerprint template protection. In *IS&T/SPIE Electronic Imaging* (2010), International Society for Optics and Photonics, pp. 75410R–75410R.

[215] YANG, B., BUSCH, C., GAFUROV, D., AND BOURS, P. Renewable minutiae templates with tunable size and security. In *Pattern Recognition (ICPR), 2010 20th International Conference on* (2010), IEEE, pp. 878–881.

[216] YANG, B., HARTUNG, D., SIMOENS, K., AND BUSCH, C. Dynamic random projection for biometric template protection. In *Biometrics: Theory Applications and Systems (BTAS), 2010 Fourth IEEE International Conference on* (2010), IEEE, pp. 1–7.

[217] YANG, B., LI, G., AND BUSCH, C. Qualifying fingerprint samples captured by smartphone cameras. In *Image Processing (ICIP), 2013 20th IEEE International Conference on* (2013), IEEE, pp. 4161–4165.

[218] YANG, B., LI, X., AND BUSCH, C. Collecting fingerprints for recognition using mobile phone cameras. *Electronic Imaging 8304* (2012), 83040L.

[219] YANG, W., HU, J., AND WANG, S. A delaunay quadrangle-based fingerprint authentication system with template protection using topology code for local registration and security enhancement. *Information Forensics and Security, IEEE Transactions on 9*, 7 (2014), 1179–1192.

[220] YANG, W., HU, J., WANG, S., AND STOJMENOVIC, M. An alignment-free fingerprint bio-cryptosystem based on modified voronoi neighbor structures. *Pattern Recognition 47*, 3 (2014), 1309–1320.

[221] YANG, W., HU, J., WANG, S., AND YANG, J. Cancelable fingerprint templates with delaunay triangle-based local structures. In *Cyberspace Safety and Security*. Springer, 2013, pp. 81–91.

[222] YANG, X., AND HAN, F. Mobile fingerprint for identity authentication in mobile commerce applications. In *Proceedings of the 12th International Conference on Advances in Mobile Computing and Multimedia* (2014), ACM, pp. 157–160.

[223] YUAN, B., SU, F., AND CAI, A. Fingerprint retrieval approach based on novel minutiae triplet features. In *Biometrics: Theory, Applications and Systems (BTAS), 2012 IEEE Fifth International Conference on* (2012), IEEE, pp. 170–175.

[224] ZHANG, J., AND WANG, J. An overview of principal curves. In *Chinese journal of computer* (2003), pp. 129–146.

[225] ZHENG, R., ZHANG, C., HE, S., AND HAO, P. A novel composite framework for large-scale fingerprint database indexing and fast retrieval. In *Hand-Based Biometrics (ICHB), 2011 International Conference on* (2011), IEEE, pp. 1–6.

[226] ZHOU, J., AND GU, J. A model-based method for the computation of fingerprints' orientation field. *Image Processing, IEEE Transactions on 13*, 6 (2004), 821–835.

[227] ZHOU, W., HU, J., WANG, S., PETERSEN, I., AND BENNAMOUN, M. Partial fingerprint indexing: a combination of local and reconstructed global features. *Concurrency and Computation: Practice and Experience* (2015).