Effective Solutions for Partial Fingerprint Indexing and Multi-Sensor Fingerprint Indexing

Author:
Zhou, Wei

Publication Date:
2016

DOI:
https://doi.org/10.26190/unsworks/19291

Effective Solutions for Partial Fingerprint Indexing and Multi-Sensor Fingerprint Indexing

Wei Zhou

A thesis submitted in fulfilment
of the requirements for the degree of
Doctor of Philosophy

School of Engineering and Information Technology
University of New South Wales
Canberra

March 2016

Abstract

A fingerprint is a type of oriented texture on a fingertip with locally smooth and intervening ridges and valleys. It is characterized to be both unique and permanent, which makes it become the most practical and widely used biometric technique since 1980s. Over the last decade, computer technology has facilitated both the capturing and processing of fingerprint data. Therefore, automated fingerprint identification and verification systems are widely used in commercial and security applications, such as access control and denial operations, and criminal identifications.

To narrow down the candidate list before matching, the theory of 'fingerprint indexing' was developed. Fingerprint indexing uses feature vectors to describe fingerprints. Through similarity preserving transformations, these feature vectors form a multidimensional feature space, where similar fingerprints characterized by similar features are arranged as neighbors in the feature space. For retrieval or identification, the query fingerprint is mapped into a point in the same feature space, and the neighboring fingerprints are compared one by one until a match is found.

A number of fingerprint indexing schemes have been proposed based on different fingerprint features. However, most of these existing fingerprint indexing schemes are not applicable to partial fingerprint identification. This is because partial fingerprint images have some parts missing and the missing parts are simply ignored (considered void). The resulting feature vector will therefore end up having too many void entries and will subsequently lose its similarity to the feature vector generated by the full fingerprint.

Therefore, it is extremely challenging to identify a partial fingerprint against a large database due to the inability to narrow down the candidate list for partial fingerprint verification. Furthermore, the traditional capture of fingerprints based

on the contact of the finger on a solid plane results in partial or degraded images due to improper finger placement, skin deformation, slippage and smearing, or sensor noise from wear and tear of surface coatings. The development of latest sensor technology allows us to acquire fingerprints with 3D fingerprint sensors. The difference resulted from multi-type sensors significantly affects the characteristics of the raw data, the extracted features and subsequently the indexing performance. It is also a challenging issue to exploit proper features or indexing algorithms for multi-sensor fingerprint indexing.

In this thesis, we aim to devise effective indexing schemes for partial fingerprint identification against very large scale databases. Furthermore, we have also acquired databases using multi-sensors and developed indexing techniques for traditional 2D images and the 3D images of fingerprints that have been generated by the new generation of touchless live scan devices. The main contributions of this thesis are listed as follows:

- For partial fingerprint indexing, we propose to combine both local features and global features. We design some novel features of minutiae triplets in addition to some commonly used features to constitute the local minutiae triplet features. Experiments carried out on FVC 2000 DB2a, FVC 2002 DB1a and NIST SD 14 demonstrate the performance improvement after adding the new features to the minutiae triplet feature set. We then propose to combine a reconstructed global feature and local minutiae triplet features to improve the performance of partial fingerprint indexing. Specifically, the minutiae triplet based indexing scheme and the FOMFE coefficients based indexing scheme are applied separately to generate two candidate lists, then a fuzzy-based fusion scheme is designed to generate the final candidate list for matching. Experiments carried out on the public database NIST SD 14 show that the proposed approach can improve the performance that has been achieved by individual partial fingerprint indexing algorithms before fusion.

- We have collected a multi-sensor fingerprint database to investigate the 3D fingerprint biometric comprehensively. It consists of 3D fingerprints as well as their corresponding 2D fingerprints captured by two commercial fingerprint scanners from 150 subjects in Australia. Also, we have tested the performance of 2D fingerprint verification, 3D fingerprint verification, and 2D to 3D fingerprint verification using different 3D images, such as raw 3D images, contrast enhanced raw 3D images, cropped raw 3D images, enhanced 3D images, and post-processed enhanced 3D images. The results show that more work is needed to improve the performance of 2D to 3D fingerprint verification and 3D fingerprint enhancement. In addition, the database has been released publicly for research purposes since 2015.

- For multi-sensor fingerprint indexing, we propose a finer hash bit selection method based on Locality-Sensitive Hashing (LSH) and Minutia Cylinder-code (MCC). That is, we divide the hash bit vectors, selected by LSH using a sliding window, into finer sub-vectors with certain fixed length, and then convert these sub-vectors into numerical approximations for MCC indexing. Also, we take into consideration another feature - the single maximum collision for indexing and fuse the candidate lists produced by both indexing methods to produce the final candidate list. Experimentations carried out on our collected multi-sensor fingerprint database show that the proposed indexing approach greatly improves the performance of fingerprint indexing. Evaluation was also conducted on some public benchmark databases for fingerprint indexing, and the results demonstrate that the new approach outperforms existing ones in almost all the cases.

List of Publications

Journal Papers

[1] W. Zhou, J. Hu, S. Wang, I. Petersen, and M. Bennamoun, "Performance evaluation of large 3d fingerprint databases," *Electronics Letters*, vol. 50, pp. 1060–1061, July 2014.

[2] W. Zhou, J. Hu, S. Wang, I. Petersen, and M. Bennamoun, "Partial fingerprint indexing: a combination of local and reconstructed global features," *Concurrency and Computation: Practice and Experience*, 2015.

[3] W. Zhou, J. Hu, and S. Wang, "Enhanced locality-sensitive hashing for multi-sensor fingerprint indexing." to be submitted to Information Forensics and Security, IEEE Transactions on, 2016.

Book Chapter

[1] W. Zhou, J. Hu, I. Petersen, S. Wang, and M. Bennamoun, "3d fingerprints-a survey," in *Biometric Security*, ch. 14, pp. 461–487, Cambridge Scholars Publishing, 2015.

Conference Papers

[1] W. Zhou, J. Hu, I. Petersen, and M. Bennamoun, *Network and System Security: 7th International Conference, NSS 2013, Madrid, Spain, June 3-4, 2013. Proceedings*, ch. Partial Fingerprint Reconstruction with Improved Smooth Extension, pp. 756–762. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.

[2] W. Zhou, J. Hu, I. Petersen, S. Wang, and M. Bennamoun, "Performance evaluation of 2d to 3d fingerprint recognition," in *Image and Signal Processing (CISP), 2013 6th International Congress on*, vol. 03, pp. 1736–1741, Dec 2013.

[3] W. Zhou, J. Hu, S. Wang, I. Petersen, and M. Bennamoun, *Network and System Security: 8th International Conference, NSS 2014, Xi'an, China, October 15-17, 2014, Proceedings*, ch. Fingerprint Indexing Based on Combination of Novel Minutiae Triplet Features, pp. 377–388. Cham: Springer International Publishing, 2014.

[4] W. Zhou, J. Hu, I. Petersen, S. Wang, and M. Bennamoun, "A benchmark 3d fingerprint database," in *Fuzzy Systems and Knowledge Discovery (FSKD), 2014 11th International Conference on*, pp. 935–940, Aug 2014.

Contents

Acknowledgements

I would like to express my appreciation to a number of people who constantly provided me with advice, help, support and understanding during my PhD study at UNSW Canberra.

Foremost, I would like to express my sincere gratitude to my supervisor Prof. Jiankun Hu for his continuous support, patience, motivation, enthusiasm, and immense knowledge during my PhD study. I would also take this opportunity to express my gratitude towards my external supervisors Dr. Song Wang, Prof. Mohammed Bennamoun, and Prof. Ian Petersen. Their valuable comments, suggestions and kind help significantly refined and improved my work.

I would also like to thank Mrs Elvira Berra, Mr. Craig Edwards and the administration group for their assistance and support that make my university life much easier and more colorful.

I am also thankful to my dear friends and colleagues for their useful suggestions and friendly encouragement during my PhD study: Miao Xie, Wencheng Yang, Jinwei Xu, Cai Li, Hu Chen, Baocheng Wu, Shanggang Lin, Liming Liu, Hongxu Wang, Qingyong Ren, Zhenhua Hao, Chao Fan, and so many more.

Last but not least, I would like to thank my parents, my brother, my sister-in-law, and my parents-in-law for their long distance support and love. The greatest debt I owe goes to my husband, Sheng Wen. Without their everlasting encouragement and understanding, this work would never have come into existence.

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

1.1 Overview

Fingerprint recognition has been the most practical and widely used biometric technique since 1980s. Over the last decade, computer technology has facilitated both capturing and processing of fingerprint data. Therefore, automated fingerprint identification and verification systems are widely used in commercial and security applications, such as access control and denial operations, criminal identifications, and the emerging bio-cryptography [1][2][3][4][5][6].

Fingerprint verification or authentication is the process used to verify whether a fingerprint sample matches a specific fingerprint stored in a database. It is a one-to-one matching. Fingerprint identification is the process used to identify an unknown person by searching a fingerprint database for a match of the input. It is the case of one-to-many comparisons which is commonly infeasible with large databases of fingerprints. Figure 1.1 illustrates the process of fingerprint identification.

Conventional fingerprint identifications from large databases are based on classification techniques whereby fingerprints are first classified into several classes to reduce the search space. The U.S. national standards ANSI/NIST [11] has classified 14 fingerprint pattern classes for automated fingerprint identification, such as arch, loop and whorl (the main patterns are shown in Figure 1.2). Such

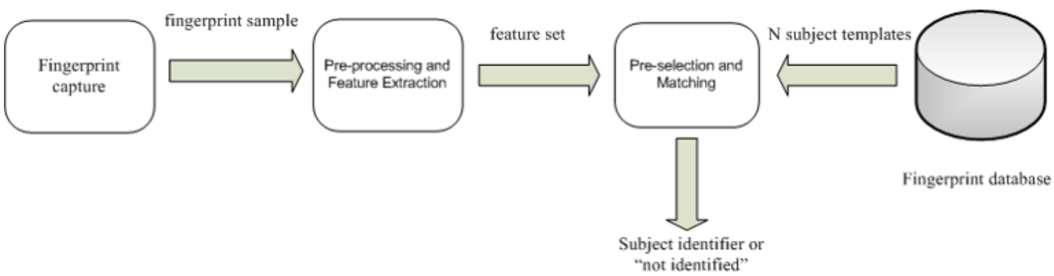


Figure 1.1: The process of fingerprint identification

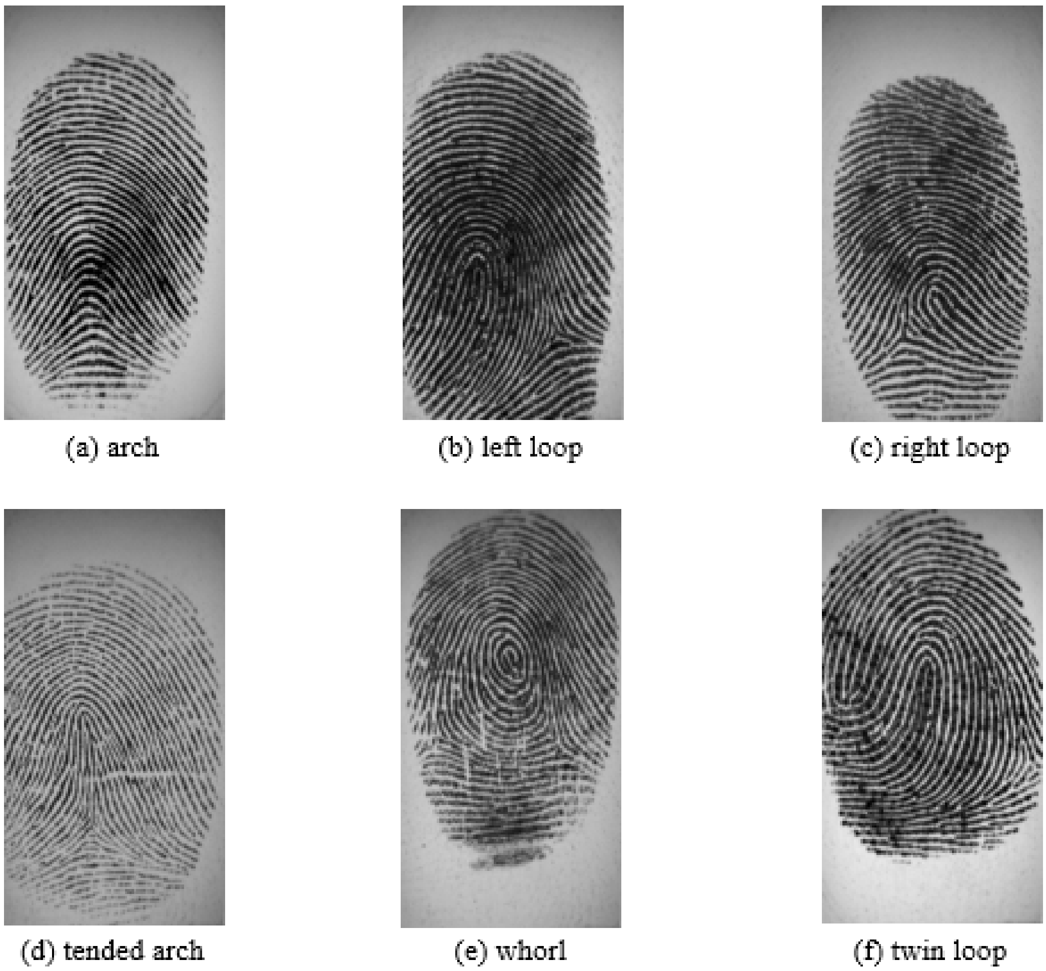


Figure 1.2: The main types of fingerprint pattern

exclusive classification based schemes have a fundamental drawback, namely that more than 90% of the fingerprints belong to only three classes (loops and whorl) due to the unevenly distribution of fingerprints. Hence the number of comparisons required to perform within one of the super classes can still be very big. Furthermore, due to the small inter-class variance and the large intra-class variance, we may have ambiguous fingerprints that are intrinsically difficult to classify even by human experts.

To address these problems, the theory of 'fingerprint indexing' (also called continuous classification) was developed, whereby instead of classifying fingerprints

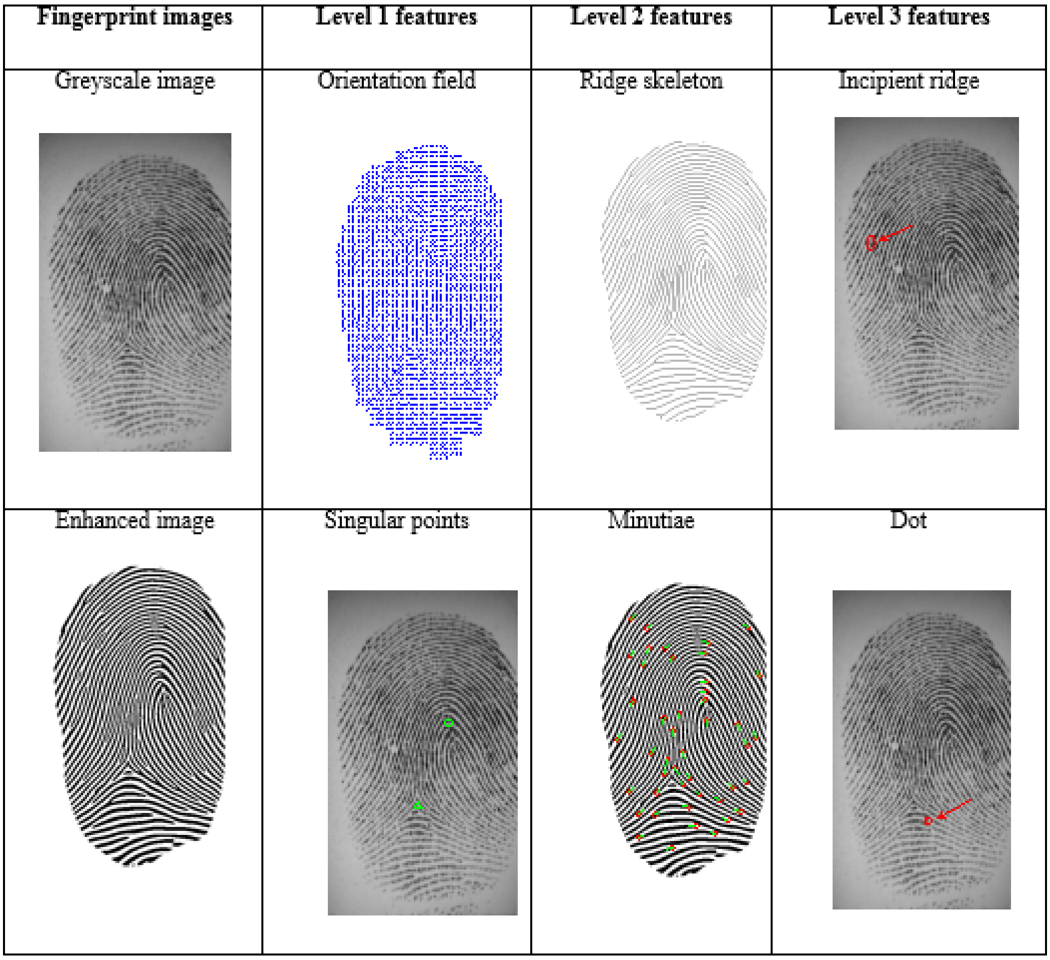


Figure 1.3: Fingerprint features

into limited and predefined classes, fingerprint indexing techniques use feature vectors to describe fingerprints. Fingerprint features are generally categorized into three levels (see Figure 1.3):

- Level 1: global features that are the macrodetails of the fingerprint such as friction ridge flows, singular points, and pattern type.
- Level 2: local features such as ridge skeletons, ridge bifurcations and endings (namely minutiae). Minutiae are generally stable and highly distinctive.
- Level 3: very-fine features including ridge contours, sweat pores, dots, and

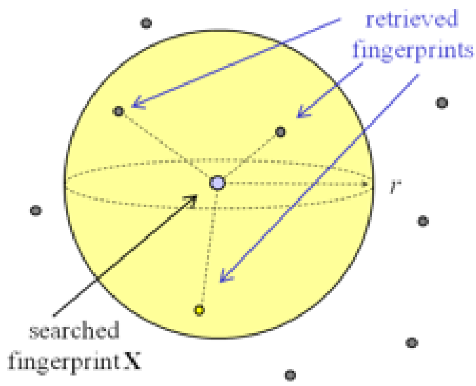


Figure 1.4: Fingerprint indexing

incipient ridges whose robust extraction needs high-resolution images (1,000 ppi) compared to the current FBI standard of 500 ppi.

Through similarity preserving transformations, the feature vectors form a feature space where similar fingerprints characterized by similar features are arranged as neighbours in a multidimensional feature space. For retrieval or identification, the query fingerprint is mapped into a point until a match is found (see Figure 1.4).

A number of fingerprint indexing schemes have been proposed. Lumini et al. [21] first proposed the idea of indexing for fingerprint identification using the orientation field. Cappelli et al. [14] used fingerprint prototype masks to generate feature vectors and studied several different strategies. Not only the level one ridge features are primarily used for fingerprint indexing, some other features have also been investigated. Bhanu and Tan [13] proposed to index fingerprints using minutiae triplets. Boer et al. [12] investigated the use of the orientation field, FingerCode and minutiae triplets as the input feature vectors and concluded that the orientation field performs the best if only a single type of feature were to be used. Wang et al. [8] proposed the FOMFE (a fingerprint orientation model based on 2D Fourier expansion) model based fingerprint indexing algorithm, which has achieved the fastest feature generation speed and an excellent searching performance. However, all these existing fingerprint indexing schemes are not applicable to partial fingerprint identification. This is because partial fingerprints have some parts missing and the missing parts are simply ignored (considered void). The resulting feature vector will therefore end up having too many void entries and will subsequently lose its similarity to the feature vector generated by the full fingerprint.

Therefore, it is extremely challenging to identify a partial fingerprint against a large database due to the inability of narrowing down the candidate list for partial fingerprint verification. Furthermore, the traditional capture of fingerprints based on the contact of the finger on paper results in partial or degraded images due to improper finger placement, skin deformation, slippage and smearing, or sensor noise from wear and tear of surface coatings. Contact-less biometric recognition performed using three-dimensional (3D) fingerprint models has the advantage of mitigating these problems. Therefore, recently there has been a major initiative supported by various government agencies for acquiring high quality 3D fingerprints using different techniques [7][8].

1.2 Problem Statement

Most recently Yi Wang et al. [9] introduced the idea of model-based partial fingerprint identification which has shown promising results. Encouraged by this preliminary progress, this research aims to devise effective indexing schemes for partial fingerprint identification against very large scale databases. Furthermore, we plan to acquire databases and develop identification techniques for the new generation of touchless live scan devices which generate 3D images of fingerprints.

The details concerning these problems are as follows:

i).Algorithm design for partial fingerprint indexing

Conventional approaches assume that the blank areas of a partial fingerprint provide no information and they solely rely on the available data of the partial segments [10]. Unfortunately in most cases the available partial segments may not contain enough ridge details to ensure the normal operational condition of the matching process. As a matter of fact, it is observed that the matching error usually increases as the number of detected local features decrease [11]. The FOMFE model [12] which uses the global feature for indexing cannot be directly applicable to partial fingerprint identification, because the resulting FOMFE model, when trained from a limited partial fingerprint segment, might not converge close enough to the final FOMFE model which results from the full fingerprint data training. We plan to incorporate local features such as minutiae that are available in the partial fingerprint segment to improve the indexing performance. We will also explore other possible features and advanced feature or decision fusion theory for partial fingerprint indexing.

ii).Acquisition of a multi-sensor fingerprint database

One barrier to experimental validation and comparison of algorithms in new biometric research areas is the lack of appropriate databases. To the best of our knowledge, in the domain of fingerprint biometric, there has been no 3D full fingerprint database with their corresponding 2D fingerprints publicly available. We therefore intend to build a 3D fingerprint database for algorithm validation and testing. The large size of the database will provide meaningful statistical analysis and a truthful assessment of the performance of the state-of-the-art algorithms in this area. Also, we plan to collect samples of corresponding 2D fingerprints. Our database can then serve as a standard database for developing identification techniques for 2D to 3D images of fingerprints. The resolution of these identification issues will require an innovative approach which will significantly advance research in the area of biometrics. It will also lead to the development of important commercial products.

iii).Algorithm design for multi-sensor fingerprint indexing

The difference resulted from multi-type (2D and 3D in this research) fingerprint sensors significantly affects the characteristics of the raw data, the extracted features and subsequently the indexing performance. Only limited research has been carried out on the scale of impact [13][14] or non-linear distortion [15][16] in multi-sensor matching. It is still a challenging issue to exploit proper features or indexing algorithms for multi-sensor fingerprint indexing.

1.3 Thesis Contributions

This section summarizes the major contributions of this research.

- For partial fingerprint indexing, we proposed to combine both local features and global features. We design some novel features of minutiae triplets in addition to some commonly used features to constitute the local minutiae triplet features. Experiments carried out on FVC 2000 DB2a, FVC 2002 DB1a and NIST SD 14 demonstrate the performance improvement after adding the new features to the minutiae triplet feature set. We then propose to combine the reconstructed global feature and local minutiae triplet features to improve the performance of partial fingerprint indexing.

Specifically, the minutiae triplet based indexing scheme and the FOMFE coefficients based indexing scheme are applied separately to generate two candidate lists, then a fuzzy-based fusion scheme is designed to generate the final candidate list for matching. Experiments carried out on the public database NIST SD 14 show that the proposed approach can improve the performance that has been achieved by individual partial fingerprint indexing algorithms before fusion.

- We have collected a 3D fingerprint database to investigate the 3D fingerprint biometric comprehensively. It consists of 3D fingerprints as well as their corresponding 2D fingerprints captured by two commercial fingerprint scanners from 150 subjects in Australia. We have also tested the performance of 2D fingerprint verification, 3D fingerprint verification, and 2D to 3D fingerprint verification using different 3D fingerprint images, such as unraveled 2D equivalent images and the enhanced ones. The results show that the cropped enhanced 3D images based on singular points can achieve the best performance regarding 2D to 3D fingerprint verification, and we choose those images as the standard 3D images for the subsequent experiments. In addition, we released the database publicly in 2015.

- For multi-sensor fingerprint indexing, we propose a finer hash bit selection method based on LSH. That is, we divide the hash bit vectors, selected by LSH using a sliding window, into finer sub-vectors with certain fixed length, and then convert these sub-vectors into numerical approximation for MCC indexing. We also take into consideration another feature - the single maximum collision for indexing and fuse the candidate lists produced by both indexing methods to produce the final candidate list. Experimentations carried out on our collected multi-sensor database (2D and 3D databases) show that the proposed indexing approach greatly improves the performance of fingerprint indexing. Evaluation was also conducted on some public benchmark databases for fingerprint indexing, and the results demonstrated that the new approach outperforms existing ones in almost all the cases.

1.4 Thesis Organization

The rest of this thesis is organized as follows. Chapter 2 presents the literature review on state-of-the-art fingerprint indexing approaches and 3D fingerprint technology. Chapter 3 presents the indexing approach for partial fingerprint. Chapters 4 describes the 2D and 3D fingerprint databases we have collected and analyzes the performance of 2D to 3D fingerprint recognition. The indexing approach for multi-sensor fingerprints is illustrated in Chapter 5, and Chapter 6 summarizes this research and suggests some important problems to be solved in the future.

Chapter 2

Literature Review

State-of-the-art fingerprint indexing methods can be classified into several categories based on the features used in these approaches, such as global features, local features, or the combination of these. This chapter first discusses the state-of-the-art fingerprint indexing techniques, followed by a comprehensive introduction to the latest 3D fingerprint technology.

2.1 Fingerprint Indexing Schemes

2.1.1 Fingerprint Indexing Based on Global Features

i).Orientation Field-Based Fingerprint Indexing

Lumini et al. [17] presented a new approach for continuous fingerprint classification for the first time. The basic principle of their approach was the characterization of the fingerprints with vectors in a multidimensional space. In particular, the directional image was calculated and registered in the same way and each feature vector was computed by using the Karhunen-Loeve (KL) transform which mapped 1680-element vectors in a lower dimensional space. They compared two different approaches for fingerprint classification and evaluated their capability for latent fingerprint retrieval in large databases. Two different methodologies for latent fingerprint retrieval were considered and implemented both with an exclusive and continuous classification. The results obtained showed that better performance could be achieved, in both the methodologies, through the continuous approach.

Cappelli et al. [18] introduced a new approach to automatic fingerprint classification. The directional image was partitioned into 'homogeneous' connected regions according to the fingerprint topology, thus giving a synthetic representation which could be exploited as a basis for the classification. A set of dynamic masks, together with an optimization criterion, were used to guide the partitioning. At the same time, the adaptation of the masks produced a numerical vector representing each fingerprint as a multidimensional point, which could be conceived as a continuous classification. Experimental results carried out on the most commonly used fingerprint databases showed that, with fingerprint retrieval based on continuous classification, their method gave the best performance and exhibited a very high robustness.

Jiang, Liu and Kot [19][20] proposed a fingerprint indexing scheme which used orientation field as the main retrieval feature, and the dominant ridge distance that had very low correlation with minutiae as an auxiliary feature. A new distance measure was also proposed that better quantified the similarity evaluation between two orientation fields than the conventional Euclidean and Manhattan distance measures. In addition, a variable search tolerance was introduced for more efficient retrieval. Furthermore, they proposed to partition the data base into clusters to avoid the exhaustive comparisons. This made the proposed method applicable to large databases and comparable to the widely studied exclusive classification in terms of the retrieval speed. Experiments and comparisons with other approaches on the NIST database 4 showed the feasibility of the proposed approaches. A problem of the proposed framework is its dependency on the accuracy and robustness of the reference point detection. In fact, a substantial portion of the retrieval errors is caused by the falsely or inconsistently detected reference point due to poor fingerprint quality.

Liu et al. [21] proposed an efficient fingerprint indexing algorithm using the registered orientation fields (OFs) as feature vectors to measure the similarity of fingerprints. The registration was performed based on a novel feature of the fingerprint, namely local axial symmetry (LAS), which could correct the location and direction estimation of a reference point. Experimental results showed this indexing scheme could remarkably reduce the workload of the fingerprint authentication system and its efficiency could be consistently promoted when the training set grew larger.

Li et al. [22] proposed a fingerprint indexing approach employing three kinds of symmetrical filters, which were the core type filter, the delta type filter and the parallel type filter, to map the orientation fields of different fingerprints into three different feature spaces. The magnitudes of the filters' response were extracted and a distance measure was formulated to compute the distance to each destination pattern for fingerprint indexing. The experimental results showed that the proposed method was effective in performing fingerprint indexing. Furthermore, the filters could be separated into two one-dimensional filters, which made the feature extraction process very fast.

Liu et al. [23] proposed an invariant representation of orientation fields based on a set of Polar Complex Moments (PCMs) for fingerprint indexing. PCMs were capable of describing fingerprint orientation patterns including singular regions and restoring spurious orientations in noisy fingerprints. Unlike most indexing schemes using the raw orientation data, a set of rotation moment invariants were derived from PCMs to form a compact feature vector, which was beneficial for clustering-based fingerprint indexing. Experiments on NIST DB4 and FVC 2002 Db1a demonstrated the effectiveness of the proposed invariant representation for fingerprint indexing. The PCMs-based features derived from orientation field have low correlation with the ridge frequency and minutiae features of fingerprint, which can be integrated to further improve the indexing performance.

In [12] Wang et al. proposed a fingerprint orientation model based on 2D

Fourier expansions (FOMFE) in the phase plane. The FOMFE did not need any prior knowledge of singular points. It was able to describe the overall ridge topology easily, including the Singular Point regions, even for noisy fingerprints. The FOMFE provided a detail description for orientation features, which enabled its profitable use in feature-related applications such as fingerprint indexing. FOMFE coefficients were exploited to generate the feature vector for indexing, whereas most indexing schemes used raw orientation data. Experiments conducted on a public databases showed that the proposed FOMFE could remarkably increase the accuracy of fingerprint feature extraction and thus that of fingerprint matching. Furthermore, the FOMFE had a less-computational cost and could work very effectively on large fingerprint databases.

ii).Singular Point-Based Fingerprint Indexing

Liu et al. [24] proposed an indexing approach based on SP correlation. SP detection and direction estimation were achieved simultaneously by applying a T-shape model to directional field (DF). The T-shape model revealed the intrinsic nature of SPs including cores and deltas which broadly existed in fingerprint images but were seldom utilized in fingerprint indexing. Specifically, one main-axis with corresponding confidence to a core, and three main-axes to a delta were obtained. As a part of the T-shape model, the summation of all main-axes was used to measure the candidate SPs confidence. Then the Minimum Average Correlation Energy (MACE) filter, a kind of distortion-tolerant filter, was used to synthesize templates and perform correlation computation to give the similarity measurement. Further indexing was obtained by sorting the similarity between the query image and all stored templates. Due to the feature detection process being SP-based, this approach cannot deal with fingerprints that have no SPs, such as arch type ones.

2.1.2 Fingerprint Indexing Based on Local Features

i).Minutia-Based Fingerprint Indexing

Germain et al. [25], for the first time, proposed to use redundant combinations of three minutiae points when forming indices in their Flash Framework. The full index consists of nine components: the length of each side, the ridge count between each pair, and the angles measured with respect to the fiducial side. Later, Bebis et al. [26] proposed to use minutiae triangles of the Delaunay triangulation for indexing. Given a minutiae triangle, they computed three invariants which were then used to form a 3-dimensional index. The invariants were based on the sides and angles of the minutiae triangle: $\frac{l_1}{l_3}, \frac{l_2}{l_3}$, and $\cos(A)$, where l_1, l_2 and l_3 are the three sides of the triangle with the constraint $l_1 \leq l_2 \leq l_3$, and A is the angle between the smallest two sides l_1 and l_2. Then, the invariants were quantized as in preprocessing. The resulting index was used to retrieve from the database all the entries stored at the same index table location. To account for noise, they also retrieved entries stored in a small neighborhood (i.e., a circle of radius 2) around the indexed location. Experiments on their own collected ink database showed that indexing could be implemented in a low-dimensional space and achieve good performance.

Bir Bhanu et al. [27] presented a model-based approach, wherein novel features of triangles formed by the triplets of minutiae were used as the basic representation unit. The triangle features that were used are: its angles, handedness, type, direction, and maximum side. Geometric constraints based on other characteristics of minutiae were used to eliminate false correspondences. Experimental results on live-scan fingerprint images of varying quality and NIST special database 4 (NIST 4) have shown that the indexing approach efficiently narrows down the number of candidate hypotheses in the presence of translation, rotation, scale, shear, occlusion, and clutter. The performance of the above approach when evaluated against another prominent indexing approach has shown that the adopted model-based approach is better for both the live scan database and the ink-based database NIST 4.

Jea et al. [28] proposed to index partial fingerprints before matching using secondary features of minutiae. For each central minutia, k nearest minutiae around it were selected, and the secondary feature was formed by the central minutia and any two of its neighboring minutiae. Therefore, there were C_k^2 secondary features for each minutia. Then, secondary features were put into different bins according to their neighbors properties and positions that were relative to the central minutiae. They divided the space around a central minutia into several, say 8, quadrants, and each of the two neighboring minutiae would belong to one of the quadrants. When matching was performed, only the feature distances of the secondary features within the same bin were examined. This improved the matching speed significantly. The average number of times of matching was reduced from 600 times to 47 times per feature.

X. Liang et al. [29] proposed a more accurate fingerprint indexing algorithm to efficiently retrieve the top N possible matching candidates from a huge database. This method is based on minutia neighbourhood structure (this structure contains richer minutia information) and a more stable triangulation algorithm (low-order Delaunay triangles, consisting of order 0 and 1 Delaunay triangles), which are both insensitive to fingerprint distortion. The indexing features include minutia detail and attributes of low-order Delaunay triangle (its handedness, angles, maximum edge, and related angles between orientation field and edges). Experiments on databases FVC2002 and FVC2004 show that the proposed algorithm considerably narrows down the search space in fingerprint databases and is stable for various fingerprints. When compared with other indexing approaches, the results obtained by them have shown better performance, especially on fingerprints with distortion.

Arun Ross et al. [30] extended the indexing framework based on minutiae triplets by utilizing ridge curve parameters in conjunction with minutiae information to enhance indexing performance. A 9-dimensional index space model based on minutiae triplets of Delaunay triangulation and the associated ridge curves was built: 3 features derived from the sides and maximum angle of the triplet, and the other 6 features generated from fitting a quadratic curve to the ridges associated with each triplet: each ridge curve was represented as a second order curve parameterized by 3 coefficients. The ratio of these coefficients constructed 6 features. They demonstrated that the proposed technique facilitated the indexing of fingerprint images acquired using different sensors. Experiments on the publicly available FVC database confirmed the utility of the proposed approach in indexing fingerprints.

Singh et al. [31] used level-2 minutiae features and level-3 pore features as the parameters for fingerprint indexing. In their approach they formed a 'Delaunay triangle' using the minutia information as a first-step to generate minutiae triplets. The indexing parameters were then computed, which included both minutiae and pore information: average angle in minutiae triplet, triangle orientation, triplet density, longest side in minutiae triplet, Min-Max distance between minutiae points and k-nearest neighbor pores, and average distance of k-nearest neighbor pores. The identification performance was further improved by incorporating a Dempster Shafer theory based score fusion algorithm. Experimental results on a high resolution fingerprint database showed that the proposed algorithm improved the identification performance by at least 10% when compared to previous fingerprint identification algorithms, which used level-2 fingerprint features only.

Iloanusi et al. [32] proposed a new structure named 'minutiae quadruplet' for the purpose of indexing. A minutia quadruplet was a quadrilateral formed from a set of 4 minutiae points. Seven features from a minutia quadruplet were proposed for indexing fingerprints: the first two features were the differences of two opposite angles in the quadruplet, the second pair of features were the diagonals of the quadruplet, the third pair of features were the heights of the inner parallelogram, whose vertices were the midpoints of the sides of the quadruplet, and the last feature was a composite global feature that combined the sides and the areas of the quadruplet and the parallelogram. Minutiae quadruplets allowed the use of features that were less sensitive to deformation as compared to minutiae triplets. Experiments on fingerprints with spurious minutiae points and fingerprints with missing minutiae showed that the technique was reasonably robust. The retrieval strategy was computationally cheaper and the proposed algorithm had fewer requirements for storage.

Cappelli et al. [33] proposed a hash-based indexing method based on minutia cylinder-code (MCC) to speed up fingerprint identification in large databases. MCC encoded the neighboring information of each minutia point into a fixed length bit vector. Then the bit vector was indexed by means of Locality-Sensitive Hashing (LSH). The similarity between two templates was counted by the number of collisions of each pair of binary vectors. Experimentations carried out over all the benchmark databases showed that in spite of the smaller set of features used (minutia position and direction only), the proposed approach outperformed the existing ones in almost all of the cases. Details on MCC and LSH will be introduced in Chapter 5.

Yuan et al. [34] proposed a novel fingerprint retrieval approach based on minutiae triplet. In their approach, two techniques were designed to compensate for the relatively weak discriminative power of minutiae triplet. The first was to derive the number of matched minutiae polygons from matching information of minutiae triplets, which offered additional information to evaluate the similarity of two minutiae sets. The second one was to attach a one-bit flag to each of the feature indices to enhance the matching precision. The features they employed were: three sides of the triangle, three differences of orientations between two consecutive vertices, and the handedness of the triangle. Experimental results on FVC 2000, 2002, 2004 databases and NIST SD27 database showed that the proposed approach efficiently narrowed down the number of candidate fingerprint images, and outperformed the state-of-the-art algorithms. Furthermore, it showed the good performance on very difficult latent-to-roll filtering.

Similar to MCC, Vij et al. [35] developed a minutiae based feature representation that was provably invariant to affine deformations and hence made it applicable directly to minutiae based templates, but the representation avoided the use of minutiae orientations to make the method applicable to the widest variety of existing templates. The atomic unit of their representation was a fixed-length descriptor for a minutia that captured its distinctive neighborhood pattern in an affine-invariant fashion. For each minutia, the nearest n neighbors of it were first calculated, then for each combination, m ($m < n$) points were arranged in clockwise order. For every 4 consecutive points A, B, C, D in the clock-wised m points, four features were calculated: the ratio of the areas of the triangles formed by minutiae triplets A, B, C and A, B, D, the ratio of the lengths of the largest side of the triangles formed by minutiae triplets A, B, C and A, B, D, the ratios of the median and minimum angles of the triangles formed by minutiae triplets A, B, C and A, B, D. A weighted combination of these four features was computed to get one final invariant value that described the local arrangement of these m points. Experimental results showed that the algorithm efficiently narrowed down the size of the database to be searched and was robust under missing or spurious minutiae.

In [36][37], the authors proposed an indexing algorithm that used a new representation of fingerprints, which was based on an extension of the Delaunay triangulation, combined with a strategy that allowed to discard bad quality triplets. Specifically, they used Delaunay triangles with higher order than 1 to build a set of triangles that collected more geometric information. In the indexing stage, a feature vector from each extended Delaunay triplet was extracted: triangle sign, relative directions of the minutiae with respect to their opposite side, and ridge counters between each segments. Exhaustive experimentation conducted on some of the FVC and NIST datasets showed the high performance of their algorithm regarding other solutions reported in literature.

Wang et al. [38] proposed a theoretical framework for systematically learning compact binary hash codes and developed an integrative approach to hash-based fingerprint indexing. Specifically, they built on the popular minutia cylinder-code (MCC) and were inspired by observing that the MCC bit-based representation was bit-correlated. Accordingly, they applied the theory of Markov random field to model bit correlations in MCC. This enabled them to learn hash bits from a generalized linear model whose maximum likelihood estimates could be conveniently obtained using established algorithms. They further designed a hierarchical fingerprint indexing scheme for binary hash codes, which combined the merits of LSH and geometric hashing. In geometric hashing, they created an integer array to construct a geometric dictionary. In particular, they implemented the dictionary as a hash table container with the created integers as access keys. The access keys of each dictionary encoded the global geometric configuration from the view of a particular basis point. The matching of geometric dictionaries mimicked geometric hashing in a micro way. The proposed Geo-LSH indexing approach could effectively achieve superior identification accuracy and was more robust in the presence of noise and missing points.

ii).Ridge-Based Fingerprint Indexing

Feng et al. [39] proposed an invariant-based fingerprint indexing scheme, wherein the invariants were computed from ridges. In this approach, the role that a minutia played was the reference of surrounding ridges. A minutia and surrounding ridges were combined to form a substructure. For two substructures in a fingerprint, using one of them as the base, lots of invariants could be constructed to describe the relations between them. Compared to minutiae triplets based indexing algorithm, this algorithm carried more information. Promising results were obtained on public database FVC2002 DB1_A.

2.1.3 Fingerprint Indexing Based on Hybrid Features

De Boer et al. [40] proposed a fingerprint indexing approach based on three multiple fingerprint features, namely the registered directional field estimate, Finger Code and minutiae triplets, and showed that indexing schemes that were based on these features, were able to search a database more effectively than a simple linear scan. The indexing scheme was constructed based on advanced methods of combining these features. They also developed a new indexing scheme based on combining these features. It was shown that the new scheme resulted in a considerably better performance than the schemes that were based on the individual features or on more naive methods of combining the features, thus allowing much larger fingerprint databases to be searched. The result compared to a simple linear search, allowed the size of databases to be 100 times larger, while maintaining the same FAR (False Acceptance Rate) and FRR (False Rejection Rate).

Feng and Jain [41][42] proposed a multi-staged filtering system to reduce the search space while retrieving the potential candidates for large-scale latent fingerprint matching. In addition to minutiae, other features are incrementally used based on the required time in manual feature marking. Those features include reference points, ridge quality map, ridge flow map, ridge wavelength map, and skeleton. Experiments on NIST SD 27 against a background of 10,258 rolled prints achieved a hit rate of 97.3% at a penetration rate of 39%. However, this filtering scheme depends on the singular points in the partial fingerprint segment, which might not exist.

Cappelli et al. [43] proposed a novel fingerprint retrieval method that combined level-1 (local ridge-line orientations and frequencies) and level-2 (minutiae positions and angles) features. For level-2 features, the Minutia Cylinder-Code representation (MCC) was adopted to obtain a set of fixed-length invariant features from minutiae. Various score-level and rank-level fusion approaches were tested to combine the output of the level-1 and level-2 similarity measures. Systematic experiments were carried out over six publicly available datasets, comparing the new retrieval approach with eighteen published retrieval methods and seventeen exclusive classification techniques. The results were even better than was expected: the proposed Hybrid approach largely outperformed state-of-the-art methods on all the datasets considered, including methods based on the same set of features. The search speed was also very high and compared favorably with other published methods.

Paulino et al. [44] proposed to use a fusion of level 1 and level 2 features to improve the indexing performance. Firstly, orientation field in the neighborhood of each minutia was encoded into a rotation- and translation-invariant fixed length bit vector. The bit vectors were then indexed by means of Locality-Sensitive Hashing (LSH). Then, conventional minutiae triplet based indexing was boosted by incorporating rotation constraints. Orientation field indexing and triplet indexing were fused with fingerprint indexing technique based on the Minutia Cylinder-Code (MCC) representation, and this fusion was further boosted by combining singular points and ridge period filtering. Experimental results carried out on 258 latents in NIST SD27 against a large background database (267K rolled prints) showed that the proposed approach outperformed state-of-the-art fingerprint indexing techniques reported in the literature.

2.1.4 Fingerprint Indexing Based on Matching Scores and Variable Score Threshold

Gyaourova and Ross [45] utilized match scores to index and retrieve fingerprint images. The method relied on comparing a fingerprint with a small set of reference images and using the resulting match scores to derive an index code. The proposed method can be applied to any biometric database irrespective of the biometric trait or matcher being used. Furthermore, it generates a compact code based on the evidence of a single impression of a finger. Thus, the proposed method has modest storage requirements.

Cappelli et al. [46] studied candidate list reduction criteria for fingerprint indexing approaches. Two new reduction criteria (variable threshold on score difference, and variable threshold on score ratio) were proposed and compared to the traditional ones (fixed threshold, and top ranking). An ideal criterion was also experimented with, to better understand the possible room for improvement. Systematic experiments were carried out on five publicly available datasets, using two state-of-the-art indexers based on complementary features (orientation and MCC). The results were even better than expected. The reduction criteria proposed, although quite simple, allowed a significant reduction of the candidate list in a closed-set scenario, with great improvement of the indexing performance.

2.1.5 Fingerprint Indexing Based on Other Features

Shuai et al. [47] proposed a fingerprint indexing and retrieval approach using scale invariant feature transformation (SIFT), a widely used approach especially for generic image retrieval. SIFT provides a large number of features over a wide range of locations and scales, while the number of minutiae points appearing in a plain fingerprint image impression is limited to a small number. Furthermore, the number of SIFT feature points can be regulated by a set of parameters such as the number of scales and octaves. Moreover, most of the minutiae points can also be detected by SIFT interest point detector. A composite set of features to form multiple impressions for the fingerprint representation was used to cope up with the uncertainty of acquisition (e.g. partialness, distortion). In the process of construction of the index, the use of the locality-sensitive hashing (LSH) allowed to perform similarity probes, only by examining a small fraction of the database. Experiments conducted on database FVC 2000 and FVC 2002 showed

Table 2.1: Performance Comparison of Some Well-known Indexing Approaches on Certain Benchmark Databases – Hit Rate (%)@PR=10%

	Features	Databases					
		NIST			FVC		
		DB4	DB14	SD27	2000 db2	2002 db1	2004 db1
Global	OF Clustering [19]	89.5	-	-	92.5	-	-
	FOMFE [12]	-	98	-	-	99.9	-
	PCM of OF [23]	88	-	-	-	85	-
Local	Minutiae Triplets [27]	85.5	-	-	-	-	-
	Minutiae Triplets [29]	-	-	-	-	-	99
	Minutiae Quadruplets [32]	-	-	-	-	-	98
	MCC [33]	-	95	-	-	99	-
	Minutiae Triplets [34]	-	-	-	-	-	80.7
	MCC Bit Correlation [38]	-	97	-	-	99	-
Hybrid	Type + SP + OF [41]	-	-	97.3 @PR=39%	-	-	-
	MCC+Minutiae [43]	-	98.7	-	-	100	-
	Minutiae Triplets + MCC + OF Descriptor Indexing [44]	-	-	81.8	-	-	-
Others	SIFT [47]	98	-	-	-	-	-

the effectiveness of the proposed fingerprint indexing approach.

Hartloff et al. [48] proposed a natural combination of fuzzy vault and indexing fingerprints by storing information about paths to attain both security and fast indexing. In particular, from the hashing perspective, they first converted the minutia points into a set of paths and then applied the fuzzy vault to this latter set. The path information stored were distances and certain angles, which were invariant under translation and rotation. The experiments showed the acceptable performance of the proposed method with respect to indexing, as well as matching accuracy. In addition, it was shown that the algorithm had proper security characteristics.

Jayaraman et al. [49] concentrated on the geometric properties of the principal components of features to construct triplet based indexing technique. For each model in the database, Speeded Up Robust Feature (SURF) was used to extract feature points. Then triangles were formed by the triplets of feature points. The triangle features used were its angles. In the proposed indexing technique, multiple entries of the same feature were eliminated. It reduced both computational and memory costs significantly. The geometric properties of principal components of features were found to be robust to handle translation and rotation effects.

2.1.6 Summary

We have surveyed the state-of-the-art fingerprint indexing schemes in this section. The performance of some well-known approaches to certain public benchmark databases is summarized in Table 2.1. Some other available techniques to narrow down the candidate list for large databases are based on exclusive classification. We do not discuss them in detail but provide the following oft-cited examples: geometric framework [50], eigenfeatures of ridge direction patterns [51], multichannel[52], machine learning approaches [53], singularities [54][55].

2.2 3D Fingerprint Technology

Fingerprint acquisition, for several decades, has evolved from ink (rolled or plain) to capacitive, ultrasonic, pyroelectric, thermal, and optoelectronic approaches. Among these capture approaches, contact based methods detect the geometric difference between contact and non-contact parts (e.g. ridges and valleys) of the fingertips on a device. The optical approach, on the other hand, captures the texture information of the fingerprint under examination.

Recent developments in fingerprint acquisition technology have resulted in touchless 3D (three-dimensional) live scan, which uses one digital camera and several mirrors or more than one camera that surround the finger for acquisition of a 3D fingerprint. Touchless biometric recognition performed using 3D fingerprint models has the advantages of reducing problems related to the deformations of the skin, dust on the sensor, and spoofing of latent fingerprints. Moreover, the fingerprint area usable for the recognition is wider than the one captured by traditional contact-based acquisition techniques. Therefore, the new generation of touchless live scan devices that generate 3D representation of fingerprints has been introduced to the market. A 3D single and ten fingerprint system that uses shape from shading and stereovision based technique to obtain 3D fingerprints in a non-contact fashion was developed by TBS North America [7]. Flashscan3D LLC [8] and the University of Kentucky have developed a non-contact, 3D finger scanning system, which can capture the 3D ridge-valley details of the fingertips.

To be compatible with existing 2D fingerprinting technology, there have been many attempts to extend the traditional fingerprint identification methods to 3D fingerprint identification. However, it is necessary to unroll the 3D fingerprint images into 2D equivalent ones before matching. Available unrolling algorithms can be divided into two categories - parametric and non-parametric - according to whether a model is assumed for the finger surface or not. Parametric unrolling algorithms assume that the finger surface can be represented as a parametric surface, e.g., cylinder, tube or sphere. Unlike parametric methods, non-parametric methods do not assume any models for the finger surfaces, instead, they directly compute the corresponding pixels in the 2D equivalent fingerprint image from the points in the 3D fingerprint model.

In this section, we investigated the 3D fingerprint technology comprehensively, including the advantages of this new technology, various acquisition techniques of 3D fingerprint images, the compatibility between 3D fingerprints and 2D fingerprints, and the possible research in the near future.

2.2.1 Comparison with 2D Fingerprint Technology

i).Disadvantages of 2D Fingerprint Technology

Traditional 2D fingerprinting technologies rely upon either applying ink (or other substances) to the finger tip skin and then pressing or rolling the finger onto a paper surface or touching or rolling the finger onto a glass (silicon, polymer, proprietary) surface (platen) of a special device. In both cases, the finger is placed on a hard or semi-hard surface, resulting in some disadvantages of the 2D fingerprint scanning [56]:

- obligatory maintenance of a clean sensor or prism surface;
- uncontrollability and non-uniformity of the finger pressure on the device;
- permanent or semi-permanent change of the finger ridge structure due to injuries or heavy manual labors;
- residues from the previous fingerprint capture;
- data distortion under different illumination, environmental, and finger skin conditions;
- extra scanning time and motion artifacts incurred in technologies that require finger rolling.

ii).Advantages of 3D Fingerprinting

3D touchless fingerprint acquisition is a remote sensing technology to capture the ridge-valley pattern which provides essential information for recognition. Compared with conventional fingerprinting, the advantages of 3D fingerprint scanning and processing technology include:

- **automaticity**: 3D fingerprint devices can function independently of an operator since the finger is aligned with real-time visual feedback, which gives the user real-time feedback for correct placement of the finger. The operator does not need to interact with the user unless there is a special circumstance such as a physical deformity. Therefore, quality of the print is no longer tied to the skills of the operator manipulating the acquisition. Besides, enhanced segmentation can be done for multi-fingers capture [57].

- **Image quality**: Better image quality is achieved because there is no contact of the print with the scanner to distort the image. Simultaneous acquisition of both texture and ridge depth information [58] of fingers produces higher quality fingerprint images, which can result in improved fingerprint matching accuracy.

- **Speed**: 3D fingerprint scanners can achieve fast scanning (less than 1 second). Some devices can scan ten prints simultaneously and allow for use in high volume environments.

- **Stability**: 3D fingerprint devices can function consistently regardless of dry, oily, or damaged fingertip surfaces, therefore, the failure to acquire rates are very low.

- **Compatibility**: 3D fingerprints are flattened to produce 2D equivalent fingerprints, which are consistent and compatible with existing databases and matching programs.

- **Security**: 3D fingerprinting is robust to clutter and fraud (e.g. latex overlays) because of the difficulties in faking 3D fingerprints. Besides, it can reduce risk of transfer of microorganisms and communicable diseases.

- **Low cost**: The use of off-the-shelf commodity cameras and projectors, whose performance is market driven, can help build low cost acquisition systems. Besides, no cleaning is required, which can eliminate costs and downtime associated with cleaning the platen of conventional contact based scanners between users.

iii).Disadvantages of 3D Fingerprinting

Despite the advantages, 3D fingerprinting is a new technology, and there are some drawbacks to it:

- the image resolution is not constant within the image and decreases from the center to the image extremities.

- the contrast between the ridges and the valleys is low in fingerprint images.

- defocus and motion blurriness are acquired sometimes.

2.2.2 3D Fingerprint Acquisition Technology

A 3D fingerprint acquisition system is a combination of projector(s), camera(s) and/or mirrors with calibrated positions. According to the number of cameras used in the system and the illumination pattern, we classify the acquisition technology into several categories:

i).Single Digital Camera

1. **Single Image** B.Y. Hiew et al. [59] proposed to use a digital camera to acquire the fingerprint images with the size of 640*480. The captured raw images will be normalized, segmented, enhanced and followed by the core point detection. After the core point detection, the image is cropped again into the size of 200*200 with the core point as the center. The normalized images will then be proposed by the Gabor filters to extract features. Chulhan et al. [60] introduced a hardware approach that used a camera and the wavelengths of light. Also, they proposed a strong view difference image rejection method using the distance between the core and the center axis of the finger in order to overcome the 3D to 2D image mapping problem.

   Apart from the above 3D fingerprint image acquisition method using a single image, Ruggero et al. [61] proposed to simulate contactless fingerprint acquisitions performed in different light conditions by using different hardware setups and image processing techniques. The method starts from a simulated fingerprint image or a real fingerprint image captured using a contact-based sensor. Well-known algorithms designed for fingerprint recognition systems are applied to the input image in order to extract the distinctive pattern of the ridges. Then, realistic effects such as noise, pores, and incipient ridges are introduced. The next step is the estimation of the 3D structure of the ridges, which is then superimposed on a parametric model of the finger shape, computed considering experimental measurements of the average finger curvature. In order to improve the realism of the simulated data, the lens focus blur is simulated. The model is then completed with the estimation of a realistic color pattern, obtained by applying a low-pass filter to a real contactless fingerprint image, and by adding the properties of reflectance that match the ones of the human skin. Finally, a virtual light source is used to illuminate the scene and make the details of the ridges visible.

Disadvantages: Such acquisition methods cannot get the 3D model of the fingerprint, some parts of the fingerprint region are in focus but some parts are out of focus, and the effective region of the fingerprint is very limited.

2. **Multi-Images** Gil et al. [62] proposed to use a linescan camera and a mechanical motion system to acquire the equivalent of a rolled fingerprint collected by contact means. The system captures four high resolution images at different depths using polarization rotation and birefringence at frame rate and with no moving parts. Then depth from focus is used to generate a coarse 3D data file. The captured images are registered and combined into a single high-resolution image with a resolution of 500 PPI (points per inch). Finally the 3D data is used to create the equivalent of a rolled fingerprint for comparison with standard fingerprint databases.

   Pang et al. [63] used a photometric stereo 3D reconstruction system to obtain 3D fingerprint data. The system comprises of a camera with a resolution of 659×493 pixels and seven LED lamps mounted around it. By synchronizing the camera and lamps, seven fingerprint images under various lighting conditions can be captured within 0.2 second. Using the calibrated lighting directions and image intensities in seven images, the surface normal at each image point can be estimated by solving a nonlinear equation. Finally, 3D models of fingerprints are obtained through surface normal integration.

Ajay et al. [64] developed a low-cost 3D fingerprint acquisition system using a single camera. Several finger images are acquired using a contactless imaging setup and the average/expected distance between the camera and the finger is around 10cm. Seven illumination sequence and the image acquisition is synchronized and controlled by a computer using a very low-cost imaging interface. The position of LEDs on the acquired images is calibrated. Each of these images is downsampled (after edge detection, boundary scanning, down) to extract 500×350 pixels region of interest (ROI). Once the ROI images are extracted, 3D fingerprint surface is reconstructed using the shape from shading technique.

ii).Two Cameras

Ruggero et al. [65] presented a novel methodology being able to obtain a 3D reconstruction of the fingertip in less constrained conditions. The method is based on a single two-view acquisition of the fingertip with the aid of a fixed projected pattern. The finger is placed according to the depth of focus of the cameras, and in the overlapping field of views. The proposed methodology can be applied to a single acquisition composed by two frames, captured using a synchronization trigger. The projected pattern is used in order to extract a set of reference points in the two images, which are rapidly matched by using the geometric information related to the pattern itself. The finger model is then reconstructed by using the information related to a previous calibration of the cameras. A novel algorithm is then used in order to remove the light pattern from the captured images, and one input image is wrapped on the resulting 3D model, obtaining a 3D pattern with a limited distortion of the ridges. Finally, an enhancement method is applied to the texture of the 3D model in order to improve the visibility of the distinctive characteristics of the fingertip.

Yao et al. [66] presented a theoretical study to reconstruct a set of 3D minutiae from two planar minutiae images captured by mobile devices. First, two fingerprint images were obtained by using two cameras with known relative positions under the assumption that the images were obtained by cameras via orthogonal projection and the minutiae did not contain angle information. Then two planar minutiae sets were extracted from these two images to reconstruct 3D minutia points.

iii).The Surround Imager

Geppy et al. [67] in TBS North America developed a 3D fingerprint acquisition technology named the surround imager (SI). The device is a cluster of 3 or 5 cameras located on a semicircle and pointing to its center, where the finger has to be placed in a correct position so that it is completely contained in the field-of-views of the cameras at the same time during the acquisition. Moreover, the device contains a set of several green LED arrays and the large size has also been chosen to dissipate the heat generated by the light system.

The Surround Imager provides a negative polarity representation of the fingerprint, i.e. the ridges appears to be brighter than the valleys. The image obtained by the device contains also the structure of the valleys. The 3D reconstruction procedure is based on stereovision and photogrammetry algorithms. Thus, the exact position and orientation of each camera (camera calibration) with respect to a given reference system is needed for further processing. The calibration is done off-line, using a 3D target on which points with known positions are marked. To facilitate the integration of the Surround Imager into existing systems, a 2D version of the reconstructed fingerprint is also provided after the reconstruction.

The computed 3D finger geometry can be used to virtually roll the fingerprint onto a plane, obtaining a complete rolled-equivalent fingerprint of the acquired finger.

iv).Structured Light Illumination (SLI)

The idea of SLI is to project a structured pattern of light onto the target surface and extract the depth by the amount of deviation that the reflected light pattern undergoes. Flashscan3D LLC. and the University of Kentucky [57][58][68] [69] have developed the following non-contact 3D scanning systems that employ structured light illumination (SLI).

1. **SLI single Point Of View (POV)** In the SLI single POV approach, the scanner, which can simultaneously acquire 3D scans of all the five fingers and the palm in high speed and fidelity, consists of a commercial off-the-shelf projector to project the SLI patterns and a high resolution camera to capture the shape deformed SLI patterns reflected from the target being s-canned.

   The algorithm for fingerprint scanning is phase measuring profilometry (PMP), which originates from the classical optical interferometry techniques and can make a 3D scan of the human finger with sufficiently high resolution so as to record 3D ridge depth information. Post processing of these scans is performed later to virtually extract the finger and palm surfaces, and create 2D flat equivalent images.

2. **SLI Subwindowing** In the SLI Sub-window technique, the scanner uses a custom LED line source with a static SLI pattern and cameras operating in subwindow mode rather than full-frame for increased frame rates.

The hardware [70] consists of a simple projection system with an LED based illumination module and a photographic slide with encoded sine wave patterns. The projection system effectively projects a static image pattern on a target surface. A small region of interest (ROI) of the pixel resolution in the camera sensor is chosen. The ROI is called an image slice. Additionally, the exposure time of the camera is set very low which limits the amount of light available per frame but helps in capturing the 2D image slices at a very high frame rate. Using the sub-window based approach, the 2D image slices are captured at a much higher frame rate with the finger moving across the projector and camera's fields of view in a swipe like motion. The number of image slices captured N, is based on the camera's frame rate and the speed at which the finger moves in the scan volume. The image slices are stitched using an image registration algorithm to create an image mosaic of the full fingerprint.

Full-hand scanners using SLI Subwindowing were also developed by Flash-scan 3D LLC.. For this type of scanner, a total of four cameras capture image slices for each finger. Multiple image-slices are captured by each camera at a very high frame rate to span the length of a full hand (from tip of the middle finger to the bottom portion of the palm). For a 3D full-hand scan, a subject moves his/her hand in a vertical direction. The scanner has a built-in proximity detector to turn on the LED line source and project a static SLI pattern on to the target. Each camera only captures a portion of the hand and all the cameras are hardware synced to capture the image slices at the same time. Multiple image-slices for each camera are stitched to create a mosaic of texture image of the hand portion in addition to generating the phase map for 3D depth computation using the projected

SLI pattern.

2.2.3 Compatibility with 2D Fingerprints

There are two ways to develop an Automatic Fingerprint Identification System (AFIS) using 3D fingerprints: (1) 3D image based and (2) 2D flat equivalent image based. The former requires to develop new feature extraction and matching methods. The latter can make use of the existing algorithms for 2D fingerprint processing after 3D fingerprint scans are unraveled into 2D flat equivalent ones.

The flattening approaches can be roughly classified into parametric and non-parametric methods. While parametric methods try to project the 3D object onto a parametric model, e.g., a cylinder, and then flatten the model, nonparametric methods apply the flattening directly to the 3D object.

i).Parametric Methods

1. **Cylindrical Model**

   Yi Chen et al. [71] [69] used a cylinder as the parametric model. Since a cylindrical model is the closest model to the finger shape, it is a reasonable choice for parametric unwrapping of 3D fingerprints. The transformation in this method is often straightforward. The texture of the fingerprint is projected onto the cylinder which surrounded the finger, and then the 2D fingerprint is obtained by flattening the cylinder. Each point (x, y, z) in the fingerprint is transformed to the cylindrical coordinate (θ, z), where $\theta = tan^{-1}(x/y)$.

   Shortcomings: It does not preserve the relative distance between the points on the fingerprint surface, which introduce a horizontal distortion to the flattened fingerprint.

2. **Tube Model**

Several algorithms to unravel 3D fingerprints into 2D equivalent images using a tube model [68] [69] [62] [65] have been developed. The finger is similar to a cylinder but tends to taper in radius toward the fingertip. So the tubular fit algorithm fits a series of consecutive circles to the 3D fingerprint cross section along its length. The fingerprint points are then associated with a radius, angular value and their original Y coordinate based on each consecutive circular cross section. Knowing the radius and angle of each point allows the print to be rolled in a way mimicking the rolled print process.

3. **Fit Sphere Model**

Another algorithm, the fit-sphere algorithm, was proposed [72] to reduce the computational cost. The fit-sphere model relies upon best fitting a sphere to the fingerprint scan where the original 3D data in Cartesian coordinates is converted to the spherical coordinate (θ, ϕ, ρ). Then, fingerprint ridges will be extracted from depth by applying a bandpass filter to the ρ dimension, where the low-frequency, smooth contours of the finger surface as well as high-frequency , noise fluctuations will be removed. That is, the 3D fingerprint surface was mapped onto a plane with minimal distortion.

Shortcomings: The curvature is not an exact fit to a typical finger, so there is some projection error. While the algorithm does mimic flat fingerprint acquisition, the spherical fit algorithm does not mimic the rolled print process.

ii).Non-Parametric Methods

1. **Spring Algorithm**

   The spring algorithm [73] first extracts the smooth surface of the 3D fingerprint by smoothing the ridge and valleys by a weighted, non-linear, least square algorithm. The weights are obtained by a Gaussian function. Then the smoothed 3D surface is transformed to the 2D unrolled surface using the springs algorithm [74]. The texture of the fingerprint (ridges and valleys) is calculated by taking a difference between the original 3D surface and the smoothed 3D surface. Therefore, the final, unrolled, 2D fingerprint is obtained by putting the texture onto the unrolled surface which is extracted by the springs algorithm.

   Sara et al. [75] also adopted the spring algorithm to convert the 3D fingerprint surface into a 2D unrolled surface, however, the texture (ridges and valleys) of fingerprint is computed by curvature analysis, particularly, the points lying on ridge lines on the surface are extracted by Gaussian and mean curvature.

   Shortcomings: There are several challenges to the Spring algorithm, such as the distortion effects of the finger tip that do not mimic rolled or flattened prints. Besides, the Spring algorithm is numerically intensive.

2. **Direct Sampling**

   In this method [71], the unwrapping directly applies to the fingerprint without projecting it to a special model. The approach locally unfolds the finger surface. In fact, a 3D fingerprint is divided into thin horizontal parallel sections and each section is unfolded separately. Linear interpolation is used to obtain more slices between the main slices which results in a more smooth fingerprint. Finally, points are regenerated using linear interpolation for each horizontal slice to map the slice from 3D to 2D. The regenerating of the point for unwrapping starts from the center and goes to the nail side. The non-parametric method generates better results than the parametric method since it preserves the relative distance between minutiae in the fingerprint.

Zhao et al. [76] took distortion into consideration when converting 3D fingerprints into 2D equivalent fingerprints using direct sampling and proposed a distortion model. The distortion model aims to simulate non-uniform sampling rates caused by the nonuniform pressure across a plain fingerprint. For simplicity, two assumptions on plain fingerprint acquisition are made: 1) The finger moves towards the fingerprint sensor along the direction perpendicular to the acquisition plane of the sensor. The point on the finger surface which touches the acquisition plane first is defined as the center of the obtained fingerprint. 2) No traction or torsion is applied to the finger once it gets in contact with the acquisition plane. Under these assumptions, the pressure reaches the maximum at the center and gradually decreases as we approach the boundary of the fingerprint. Correspondingly, the sampling interval gradually increases from the center to the boundary.

3. **Valley-ridge Lines Extraction**

Xufang et al. [63] developed an approach for directly extracting valley-ridge lines from point-cloud-based 3D fingerprint models. First, the moving least-squares (MLS) method was applied to fit a local paraboloid surface and to represent the local point cloud area. On the basis of the fitting surface, the 3D fingerprint surfaces curvature and curvature tensors were calculated. By referring to the curvatures, potential valley-ridge points were detected. Through statistical means, those points were projected to the most likely valley-ridge lines. Then, by growing the polylines that approximate the projected points and removing the perturbations between the sampled points, the 3D valley-ridge lines were obtained.

Advantages: This approach can directly extract the features of valley-ridge lines without employing unwrapping, which converts 3D models to 2D but introduces distortions.

2.2.4 Feature Extraction of 3D Fingerprints

Different from 2D fingerprints, 3D fingerprint models introduce some new features, such as minutiae in 3D space. Therefore, feature detection and representation are crucial issues in 3D fingerprinting techniques.

i).Finger Surface Code

The shape index (SI) can be used to describe 3D surface using curvature information. On 3D fingerprint surface, the SIs are concentrated in numeric values representing fingerprint valley (0.25) and ridge (0.75) regions. The surface index is therefore likely to be largely distributed in this zone. Therefore the encoding scheme splits the fingerprint surface into five zones: cup, rut, saddle, ridge, cap. The direction of the dominant principle curvature is portioned into six directions. Rut and ridge zones are further divided. The resulting feature representation has 15 different values and therefore 4-bits can store resulting binary code for each pixel. This binarized representation of a 3D fingerprint surface is referred to as Finger Surface Code [64]. The matching score between two Finger Surface Codes can be computed using their normalized Hamming distance.

ii).3D Minutiae

The 2D fingerprint templates (x, y, θ) typically include position of the minutiae (x, y) and the angle θ representing the orientation of the minutiae in 2D space. This representation can be extended to include new (extended) features which can more accurately localize such minutiae in 3D space. The 3D feature z can represent the height of the vertex on the reconstructed 3D fingerprint surface at position (x, y) while the ϕ can represent the minutiae orientation in spherical coordinates with unit length 1. Such extended minutiae templates can more effectively localize the minutiae in 3D space and referred as 3D minutiae (x, y, z, θ, ϕ) [67][64].

iii).Ridge-valley Structure

Besides the coarse 3D representation of the fingerprint shape, the Surround Imager [67] provides also a finer 3D description of the ridge-valley structure. The entire 3D ridge-valley structure captured with a specific illumination can be well represented by the image gray-levels, mapping each image pixel into a 3D space $\{x, y, I(x, y)\}$, where $I(x, y)$ represents the value of the gray-level of the fingerprint image I at position (x, y).

2.2.5 Summary

Non-contact 3D fingerprint technology is gradually replacing traditional fingerprint acquisition and recognition in many applications. Recent research on 3D fingerprint biometric focuses on the acquisition of 3D fingerprint models, unwrapping 3D fingerprints into 2D equivalent ones and using existing algorithms for 2D to 3D fingerprint recognition. This section presents a comprehensive study of this new technology, mainly the acquisition of 3D fingerprints, the comparison and compatibility of traditional 2D fingerprints and 3D fingerprints.

Chapter 3

Partial Fingerprint Indexing: A Combination of Local and Reconstructed Global Features

3.1 Motivation and Contributions

A number of fingerprint indexing schemes based on all levels of features have been proposed for both full fingerprint [17][18][27][40][12] and partial fingerprint [41][42][34][44] indexing since 1997. However, the indexing techniques for full fingerprint are not applicable to partial fingerprint identification because the missing parts of a partial fingerprint are simply ignored (considered void). The resulting feature vector will therefore end up having too many void entries and will subsequently lose its similarity to the feature vector generated by the full fingerprint. The filtering or indexing schemes for partial fingerprint either depend on the singular points which are hardly found in the partial fingerprint segment, or involve excessive computation on the minutiae information.

Most recently, Wang and Hu [9] applied their prior work, namely the FOMFE model [12] to address partial fingerprint identification from another angle. Instead of extracting level 2 or level 3 features from the partial segment, the authors proposed an analytical approach to reconstructing the global orientation field (OF) by exploiting the global topological features. Specifically, they have developed algorithms to extend the partial ridge flows smoothly into the unknown segment while preserving the fidelity. This approach has shown very promising results in reducing the size of candidate lists for matching when applied in fingerprint indexing, and what is more, the information of singular points is not necessary. Motivated by this progress, in this chapter we propose to combine the estimated global feature and local minutiae information for partial fingerprint indexing. Specifically, local minutiae triplets are utilized to complement FOMFE coefficients based partial fingerprint indexing. For indexing based on minutiae triplets, certain new features as well as some commonly used features are combined to form the feature vectors for indexing. For indexing based on the global feature, orientation fields of partial fingerprints are reconstructed by FOMFE model and Smooth Extension, then the coefficients of the FOMFE model are used to form the feature vectors directly. The minutiae triplet based indexing scheme and FOMFE coefficients based indexing scheme are applied separately to generate two candidate lists for further processing. Before the generation of the final candidate list, a training process is conducted on a small portion of the query fingerprints to decide which candidate list tends to be more reliable. Based on the reliability of the candidate list and the order of the candidates in both lists, a set of fuzzy rules are derived to guide the fusion of the two candidate lists for generating the final candidate list. We have conducted a series of experiments on several public databases to evaluate the performance of the proposed scheme. For partial fingerprint indexing, the last 2000 F images in NIST SD 14 are chosen as the template fingerprints, and their corresponding S images are eroded to generate partial query fingerprints. Experimental results on FVC 2000 DB2a and 2002 DB1a show that the minutiae triplet based indexing can individually achieve better performance than state-of-the-art methods for full fingerprint indexing, meanwhile, the minutiae triplet based indexing on partial fingerprint database can be comparable to that on full fingerprint indexing if the parameters are chosen properly. Experimental results on SD 14 show that the proposed fusion method can improve the performance that has been achieved by individual partial fingerprint indexing algorithms before fusion.

The main contribution of this work is threefold: (1) the formation of feature vectors through noncollinear minutiae triangles enabled by incorporation of local minutiae triplets can effectively index partial fingerprints; (2) the fusion scheme designed for generating the final candidate list brings about an improvement in indexing performance, evidenced by a reduction in the penetration rate and search space; (3) there is no dedicated partial fingerprint database available for testing various algorithms. Although public latent fingerprint database is available [77], it is not suitable for testing certain characteristics, e.g., non-existence of singular points. Also latent fingerprint database contains many elements such as background noise and non-relevant textures where preliminary processing such as fingerprint image separation is involved. It is infeasible to use such data for partial fingerprint indexing testing as the impact of the preliminary processing is inseparable. Our database has addressed this issue and will be made publicly available to this research community.

The rest of this chapter is organized as follows. Section 3.2 elaborates on the generation of the global feature, namely FOMFE coefficients, and local minutiae triplet feature. The indexing scheme based on these features is proposed in Section 3.3. Section 3.4 describes the fuzzy based fusion approach for generating the final candidate list. Experiments on a public fingerprint databases are demonstrated in Section 3.5 and Section 3.6 concludes the whole work.

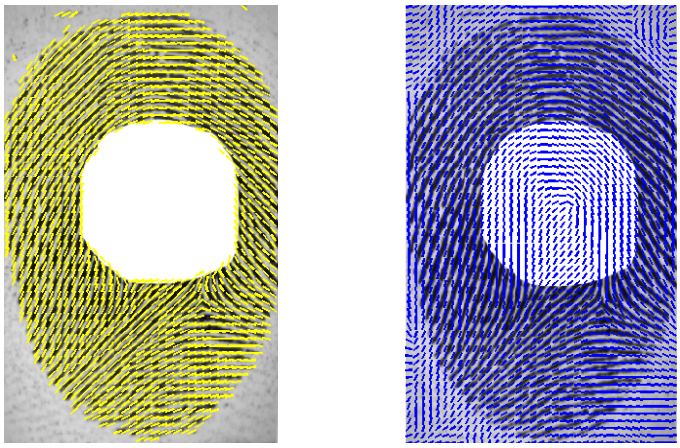
3.2 Feature Set

In this chapter, we develop a new indexing scheme which utilizes both global and local features, namely the FOMFE coefficients and minutiae triplets, respectively.

3.2.1 Global Feature – FOMFE Coefficients

i).Raw Orientation Field

An Orientation Field (OF) consists of regularly spaced grids whose elements represent the local average directions of fingerprint ridges. Therefore, it can



(a) Raw orientation field on a partial fingerprint image

(b) Reconstructed orientation field on the partial fingerprint image

Figure 3.1: Orientation field before and after smooth extension

reveal the intrinsic features of ridge topology and is a rich information resource for further fingerprint feature extraction and processing.

To obtain reliable fingerprint orientation fields, the most popular approach is to use the gradients of gray intensity in the fingerprint image. In the proposed method, the raw OF is evaluated by an improved gradient-based method [78]. The gray-scale image is first divided into blocks with an equal size of $N_B \times N_B$ pixels, then the dominant orientation angle θ in each block is computed by a weighted averaging scheme from four neighboring blocks. This approach has been proven to be more robust against noise compared with other gradient-based methods [78].

Orientation field data are often represented in cosine and sine terms and are widely used as features for fingerprint indexing. The resulting phase representation exhibits intrinsic periodic characteristics, which does not satisfy the general requirement of a Gaussian distribution in the dataset that is needed for similarity preserving transformation such as Karhunen-Loeve (KL) transform. In [12], the proposed FOMFE model can provide a compact and comprehensive description about the overall ridge topology.

ii).FOMFE Model [12]

By doubling orientation angles (θ), the orientation field of a fingerprint is transformed into a vector field. In the FOMFE model, two bivariate trigonometric polynomials are used to approximate the functions $f = (f_c, f_s)^T$. In a defined two dimensional site S: $(-l \leq x \leq l,\ -h \leq y \leq h)$, each phase function is represented as

$$f(x,y) = \sum_{i=0}^{2k}\sum_{j=0}^{2k} \varsigma_{ij}\psi_{ij}(x,y) + \varepsilon(x,y) \tag{3.1}$$

wherein $k < +\infty \in N$ is the polynomial order, ς_{ij} are the real – valued model coefficients, and ψ_{ij} are the expansion functions. In fact, Eq. (1) is a truncated 2D Fourier expansion, with $\Psi = \{\psi_{ij}(x,y)\}$ being the Fourier expansion set defined in S. According to the inverse Fourier transform,

$$\varsigma_{ij} = \int_S \widetilde{\psi}_{ij}(x,y) f(x,y) \tag{3.2}$$

wherein $\widetilde{\Psi}$ is the dual function of Ψ, $\widetilde{\Psi} = \{\widetilde{\psi}_{ij}(x,y)\}$. In practice $f(x,y)$ is replaced by the input phase data $d(x,y)$ sampled at point (x,y), and $d(x,y) = (d_c, d_s)^T$, $d_c = cosin(2\theta)$, $d_s = sin(2\theta)$.

The model coefficients ς_{ij} have been suggested as representation features for indexing, which can avoid the problem of Gaussian distribution. Compared to raw orientation field based approaches, the FOMFE model based fingerprint indexing scheme demonstrated a significant penetration rate improvement and a much faster speed in generating the feature space [12].

iii).Smooth Extension [9]

To reconstruct the global feature of a partial fingerprint, the whole site S is divided into two segments, the partial segment Ω and the unknown region χ, that is, $\chi \cup \Omega = S$. Define $\beta_S = \{\varsigma_{ij}|S\}$ as the model coefficient set for the global phase portrait to be reconstructed, $\beta_\Omega = \{\varsigma_{ij}|\Omega\}$ as that evaluated from the partial segment Ω, and $\beta_\chi = \{\varsigma_{ij}|\chi\}$ as that evaluated from the unknown region χ. Then, Eq. (3.2) is partitioned as

$$\begin{aligned}\beta_S &= \sum_S \widetilde{\psi}_{ij} d(x,y) \\ &= \beta_\Omega + \beta_\chi \\ &= \sum_\Omega \widetilde{\psi}_{ij} d(x,y) + \sum_\chi \widetilde{\psi}_{ij} d(x,y)\end{aligned} \tag{3.3}$$

The problem of partial fingerprint reconstruction is now converted to how to retrieve the global representation β_S from the partial evaluation β_Ω subject to all possible constrains imposed on β_χ.

Assume that there are M phase samples d_Ω in the partial fingerprint region, $\Omega \subset S$. Rewrite Eq. (3.1) in matrix notation, and the phase data in Ω can be estimated by the following equation.

$$\hat{d}_\Omega = \Psi_\Omega \beta_S \tag{3.4}$$

Accordingly,

$$\beta_S = \widetilde{\Psi}_\Omega^T \hat{d}_\Omega + N(\Psi_\Omega) \tag{3.5}$$

wherein $N(\Psi)$ is the null space [9] of data set Ψ.

In practice, the authors [9] proposed to extend fingerprint ridge flow trends progressively into the unknown region χ. Assume Δ is a small thin band encompassing the known region $\Omega^{(0)} = \Omega$, and $\beta^{(0)} = \beta_\Omega$ is the coefficient estimation on partial segment Ω. According to Eq. (3.4), the phase estimation of the expanded small area is

$$\hat{d}(x, y) = \Psi(x, y)\beta^{(0)}, \quad \forall(x, y) \in \Delta \tag{3.6}$$

Written in matrix notation, Eq. (3.6) is equivalent to $\hat{d}_\Delta(x, y) = \Psi_\Delta \beta^{(0)}$. After the first expansion, $\Omega^{(1)} = \Omega^{(0)} \cup \Delta$. Since the Fourier expansion set $\Psi(x, y)$ is readily evaluated at every point in the phase plane regardless of the position, it is independent of the phase structure information in $\Omega^{(1)}$.

To further expand the partial fingerprint into the unknown region χ, the phase structure in the new available segment $\Omega^{(m+1)}$ is updated upon the m-th expansion, that is,

$$\beta^{(m+1)} = \beta^{(m)} + \widetilde{\Psi}_\Delta^T \hat{d}_\Delta, \quad m = 0, 1, 2, \cdots, t-1 \tag{3.7}$$

By iterating the computation of Eq. (3.6) and Eq. (3.7) $t - 1$ times, the unknown region χ is finally filled with phase estimates. Figure 3.1 is a demonstration of the orientation field on a partial fingerprint image before and after smooth extension.

After Smooth Extension, the orientations of the missing part can be estimated step by step, and the final Fourier coefficients can be used as the feature vector for continuous fingerprint indexing. However, the FOMFE model using the global feature for the indexing method cannot be directly applied to partial fingerprint identification, because the resulting FOMFE model when trained from a limited partial fingerprint segment might not converge close enough to the final FOMFE model which comes from the full fingerprint data training. To address this issue in a bid to improve the indexing performance, we decide to incorporate local minutiae that are available in the partial fingerprint segment.

3.2.2 Local Feature – Minutiae Triplets

The features of a minutia extracted from a fingerprint image usually include its coordinates (x, y), local ridge orientation θ and minutia type (ridge bifurcation or ending denoted by 1 or 0). In our approach, a commercial fingerprint verification software VeriFinger SDK [79] was adopted to extract minutiae information for both full fingerprint images and partial fingerprint images, because the adaptive image filtration algorithm VeriFinger hosts can eliminate noises, ridge ruptures and stuck ridges for reliable minutiae extraction even from poor quality fingerprints with a fast processing speed. What is more, the resulting minutiae are sorted according to their y coordinates in an ascending order, which will benefit the construction of minutiae triplets.

i).Feature Vector of Minutiae Triangles

Since Delauney triangles [36] would change greatly if several minutiae are missing, we employ the following feature set derived from each noncollinear minutiae triangle for partial fingerprint indexing.

- **Triangle handedness:** Suppose P_1, P_2, P_3 are the three minutiae to form a triangle and their y coordinates are in an ascending order. We choose P_1 as the first vertex and use (x_i, y_i) to denote the coordinates of minutiae $P_i, i = 1, 2, 3$. Define $\phi = (x_2 - x_1) \times (y_3 - y_1) - (y_2 - y_1) \times (x_3 - x_1)$. If

$\phi > 0$, P_1, P_2, P_3 are in counter-clockwise order, then we set the vertices as $\{P_1, P_2, P_3\}$; otherwise, we order the vertices as $\{P_1, P_3, P_2\}$. By this means, we make sure that the vertices of all triangles are arranged in the counter-clockwise direction.

- **Lengths of each side:** Suppose P_1, P_2, P_3 are already in counter-clockwise order. Let $Z_i = x_i + jy_i$ be the complex number ($j = \sqrt{-1}$) corresponding to the coordinate (x_i, y_i) of $P_i, i = 1, 2, 3$. Define $Z_{21} = Z_2 - Z_1$, $Z_{32} = Z_3 - Z_2$, and $Z_{13} = Z_1 - Z_3$. The length of each side is defined as $\{L_1, L_2, L_3\}$, wherein $L_1 = |Z_{21}|$, $L_2 = |Z_{32}|$, and $L_3 = |Z_{13}|$.

- **Triangle type:** Suppose P_1, P_2, P_3 are already in counter-clockwise order. Let $\gamma = 4\gamma_1 + 2\gamma_2 + \gamma_3$, where γ_i is the type of minutiae $P_i, i = 1, 2, 3$. If P_i is a bifurcation point, $\gamma_i = 1$, or else $\gamma_i = 0$. So $\gamma \in \{0, 1, 2, 3, 4, 5, 6, 7\}$.

- **Triangle position:** Suppose P_1, P_2, P_3 are already in counter-clockwise order and the fingerprint image is aligned roughly. We divide the segment into 4 equal-sized blocks. Similar to quadrant partition, we let 1 denote the upper right block, 2 denote the upper left block, 3 denote the lower left block and 4 denote the lower right block. Let $\rho_i, i = 1, 2, 3$ be the block type of minutiae $P_i, i = 1, 2, 3$, $\rho_i \in \{1, 2, 3, 4\}$. Define $\varrho = 100\rho_1 + 10\rho_2 + \rho_3$ as the triangle position, then the number of triangle positions is 4^3.

- **Orientation differences:** Suppose P_1, P_2, P_3 are already in counter-clockwise order. Let θ_i be the local orientation of minutiae $P_i, i = 1, 2, 3$. We represent orientation difference between each pair of adjacent vertices as $\alpha_i, i = 1, 2, 3$, wherein $\alpha_1 = \theta_2 - \theta_1$, $\alpha_2 = \theta_3 - \theta_2$, and $\alpha_3 = \theta_1 - \theta_3$.

The final feature set of a minutiae triangle is in the form of an eight tuple $\{L_1, L_2, L_3, \gamma, \varrho, \alpha_1, \alpha_2, \alpha_3\}$. Among these features, L_1, L_2, L_3 and γ are the commonly used features of minutiae triplets for indexing [27][34], and $\alpha_1, \alpha_2, \alpha_3$ and ϱ are the newly designed ones since they are simple, discriminative and easy to obtain even with the singular areas missing.

ii).Geometric Constraints

To reduce the number of false correspondences obtained from querying the indexing space, some constrains on length and orientation difference are introduced.

- **Relative length difference:** Assume the length of each side of a triangle formed by minutiae triplet does not change much in different impressions of the same finger. Let L and $L^{'}$ be L_1, L_2, or L_3 in a query image and a template image, respectively. We constrain $|L - L^{'}| < \delta_L$.

- **Relative rotation:** Assume the orientation difference does not change much in different impressions of the same finger. Let α and $\alpha^{'}$ be α_1, α_2, or α_3 in a query image and a template image, respectively. We constrain $|\alpha - \alpha^{'}| < \delta_O$.

3.3 Indexing Schemes

In a real fingerprint identification system that adopts continuous classification for preselection, whether a match for a query fingerprint is found or not in the database is determined by the results of the matching algorithm. Therefore, the system's inherent matching errors (false accept and false reject) will affect the indexing system's error rate (of finding a match). However, our purpose is to assess the performance of different indexing schemes, so we can exclude the impact of different systems' inherent matching errors and investigate the indexing performance only. Specifically, since the query sample images and the template images have already been paired up with the image indices in public databases, we can get the position of the mated fingerprint in the candidate list by checking their image indices only.

3.3.1 FOMFE Coefficients based Indexing

To obtain a raw OF with lower dimensions, we choose an orientation block size N_B for our indexing approach. Suppose the resolution of the fingerprint image is $W \times H$. Thus there will be $(W/N_B) \times (H/N_B)$ blocks in a raw OF. In each block, the phase portrait is governed by the same two trigonometric polynomials, whose coefficients are concatenated to form the feature vector. Therefore the length of the coefficient-based feature vector has nothing to do with the resolution of the fingerprint image, but the trigonometric polynomial order k. The length of the coefficient-based feature vector is represented by $(2 \times k + 1)^2 \times 2$, which is several times shorter than the length of an OF-based feature vector.

To generate the feature space, the FOMFE model is applied to every template fingerprint image in the database and all the coefficient vectors form a feature matrix. Different from template images, each sample image has to go through smooth extension at first, and the final coefficient vector is combined with the template feature matrix to form the total feature space.

To reduce dimensionality, Karhunen Loeve (KL) transform is conducted. New feature vectors are truncated so that the feature vector length (FVL) can be reduced. The similarity between two fingerprints is measured by the Euclidean distance between them in the new feature space. Finally, all the indices of the template fingerprints will be sorted in the candidate list in ascending order ac-

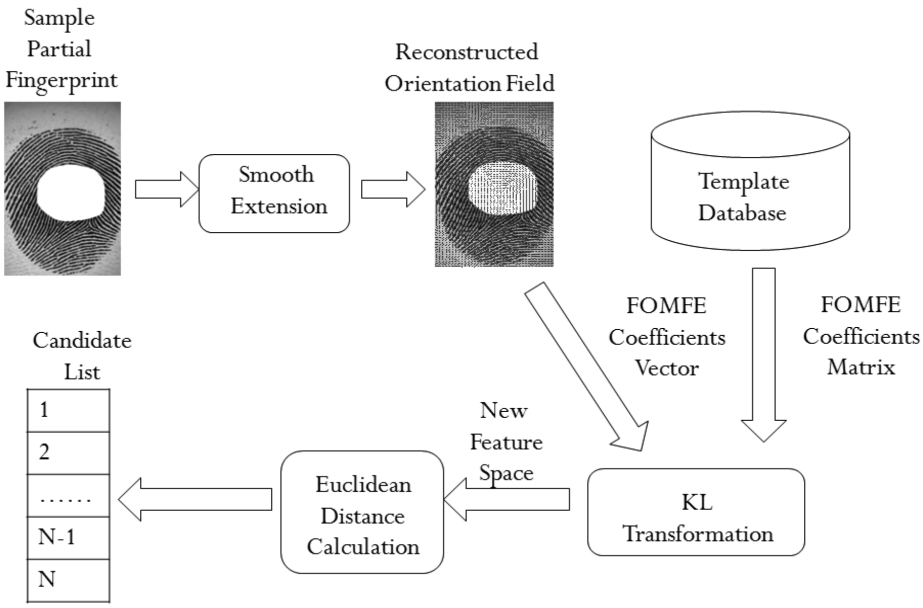


Figure 3.2: Procedure of FOMFE coefficients based indexing

cording to their distance to the query sample fingerprint image. The procedure of FOMFE coefficients based indexing is shown in Fig 3.2.

3.3.2 Minutiae Triangle based Indexing

To reduce computational complexity, minutiae that are too close to their neighbors are filtered out since triangles with a short side are too sensitive to distortion. As mentioned before, certain distortion in the sides of triangles should be allowed, so we adopt quantization to implement feature space clustering. During the registration process, each triangle in a template image is characterized by an eight-tuple vector, which means each fingerprint is viewed as a collection of points distributed in the index space with each point characterizing an eight-dimensional feature vector. Then, we quantize the triangles by the lengths of their three sides. Suppose the maximum side of all the triangles in the database is L_{max}, then the indexing space is partitioned into $(L_{max}/\delta_L)^3$ clusters. Each of the points is

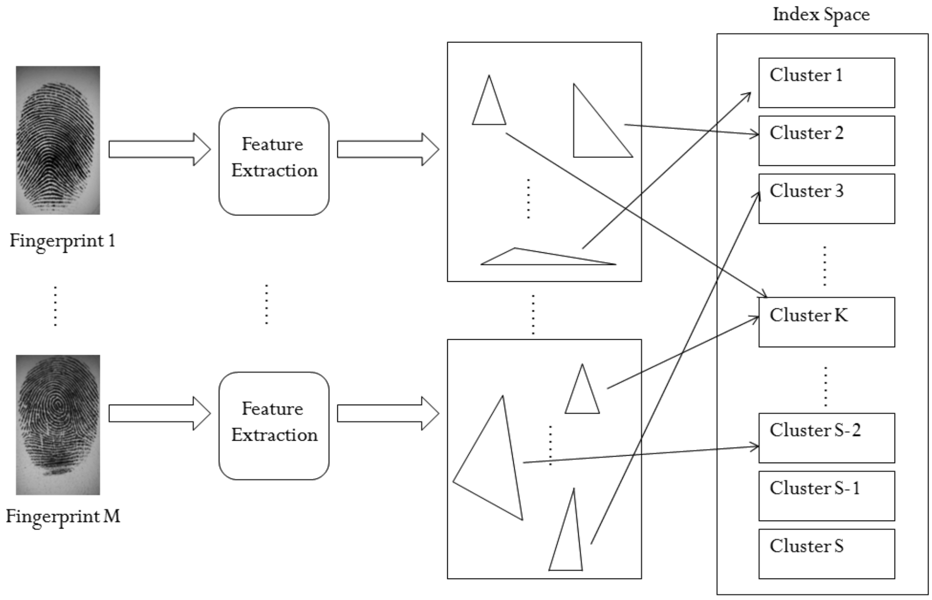


Figure 3.3: Procedure of registration of minutiae triangle based indexing

assigned to one of the pre-defined clusters based on the quantization rule. This process is repeated for every template fingerprint in the database. Thus, a cluster in the index space will have a list of fingerprint indices that have at least one point assigned to that cluster. Besides, the cluster also stores the remaining features in the eight-tuple vector for further processing except for the lengths of each side, they are $\{\gamma, \varrho, \alpha_1, \alpha_2, \alpha_3\}$.

During the query process, when a query sample fingerprint q is presented, it is first represented as a set of points with eight-dimensional feature vectors. Next, these points are mapped to individual clusters in the index space. A set of possible matching indices corresponding to a small number of clusters are then determined. After that, each point of the query fingerprint is further compared with the possible matching points in the clusters, and those points that satisfy the following requirements will be chosen:

- The triangle types γ and $\gamma^{'}$ are the same.

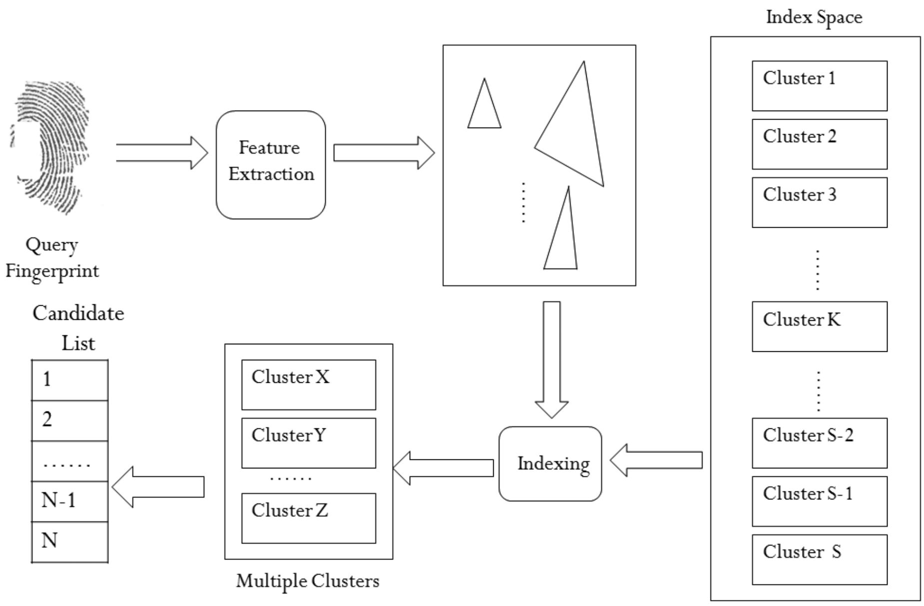


Figure 3.4: Procedure of query of minutiae triangle based indexing

- The triangle positions ϱ and ϱ' are the same.
- $|\alpha - \alpha'| < \delta_O$.

Finally, the qualified indices are sorted in the candidate list by their occurring frequency in descending order. The procedures involved in the registration and query processes of minutiae triangle based indexing are shown in Figure 3.3 and Figure 3.4, respectively.

3.4 Candidate Lists Fusion

In general, given a query fingerprint, the candidate list provided by fingerprint indexing should be as short as possible, but it should contain, with a high probability, all fingerprints similar to the query. In previous research, two reduction criteria were commonly explored: fixed threshold (only fingerprints with indexing score higher than a fixed threshold are selected) and top ranking (a fixed number of candidates with highest scores are retained). In some studies, more complex criteria to produce the candidate list have been proposed, but only with the aim of combining multiple preselection techniques.

Actually, for each candidate list, the order only reflects the general similarity, not the absolute probability of how similar the query fingerprint is to each candidate fingerprint in the database. In our approach, we treat all the template fingerprints as candidates and apply a fuzzy-based fusion scheme to the two candidate lists to generate a new candidate list for matching, because fuzzy logic [80] deals with reasoning that is approximate rather than fixed and exact. Using non-numeric linguistic variables, such as 'near', 'far', fuzzy logic can facilitate the expression of our fusion scheme.

The proposed fuzzy-based candidate list fusion scheme is composed of two stages: (a) the training stage for generating fuzzy rules and (b) the testing stage. Assume all the template images are evenly distributed in the database. In the training stage, a small portion of the query samples are indexed using the FOMFE coefficients based scheme and the minutiae triangle based scheme, respectively, then the average indexing performance (the penetration rate) is evaluated. If the penetration rate of the FOMFE coefficients based indexing is lower than the minutiae triangle based indexing, which means that the candidate list generated by the FOMFE coefficients based indexing is more reliable, then the candidate list generated by the FOMFE coefficients based indexing in the testing stage tends to be reliable too, and its top K candidates can keep the original order in the new candidate list; otherwise, the candidate list generated by the minutiae triangle based indexing tends to be reliable in both the training and the testing stages, and its top K candidates can be the top K candidates in the new candidate list. The fuzzy set in the proposed method is defined as follows:

- Three fuzzy sets, namely Low, Medium and High, are used to describe the reliability of the candidate list generated by FOMFE coefficients based indexing, which is denoted by R_F.

- Three fuzzy sets, namely Top, Middle and Bottom, are adopted to depict the position of each template at the two input candidate lists and the output candidate list. For an arbitrary query fingerprint q, two candidate lists will be generated. For any two template fingerprints q_i and q_j, their positions in both candidate lists are represented by P_{i-F}, P_{j-F},P_{i-T},P_{j-T}, respectively, and their positions in the new candidate lists are represented by P_{i-C}, P_{j-C}, respectively.

In the testing stage, for each query fingerprint, the order of all the template fingerprints in the new candidate list will be decided by the following fuzzy rules:

- If R_F is Medium, then $K = 0$;

- For a template fingerprint q_i, if P_{i-F} and P_{i-T} are Top, then P_{i-C} is Top;

- For a template fingerprint q_i, if P_{i-F} and P_{i-T} are Bottom, then P_{i-C} is Bottom;

- For a template fingerprint q_i, if P_{i-F} is Top and $P_{i-F} \leq K$ and R_F is High, then $P_{i-C} = P_{i-F}$;

- For a template fingerprint q_i, if P_{i-T} is Top and $P_{i-T} \leq K$ and R_F is Low, then $P_{i-C} = P_{i-T}$;

- For a template fingerprint q_i, if P_{i-F} is Top and $P_{i-F} > K$ and R_F is High and P_{i-T} is Top, then P_{i-C} is Top;

- For a template fingerprint q_i, if P_{i-F} is Top and $P_{i-F} > K$ and R_F is High and P_{i-T} is Bottom, then P_{i-C} is Bottom;

- For a template fingerprint q_i, if P_{i-T} is Top and $P_{i-T} > K$ and R_F is Low and P_{i-F} is Top, then P_{i-C} is Top;

- For a template fingerprint q_i, if P_{i-T} is Top and $P_{i-T} > K$ and R_F is Low and P_{i-F} is Bottom, then P_{i-C} is Bottom;

- For two template fingerprints q_i and q_j, if P_{i-F}, P_{j-F},P_{i-T} and P_{j-T} are Middle and $P_{i-F} < P_{j-F}$ and $P_{i-T} < P_{j-T}$, then $P_{i-C} < P_{j-C}$.

- For two template fingerprints q_i and q_j, if P_{i-F}, P_{j-F},P_{i-T} and P_{j-T} are Middle and $P_{i-F} < P_{j-F}$ and $P_{i-T} > P_{j-T}$ and $P_{i-T} > P_{j-F}$, then $P_{i-C} > P_{j-C}$.

- For two template fingerprints q_i and q_j, if P_{i-F}, P_{j-F},P_{i-T} and P_{j-T} are Middle and $P_{i-F} < P_{j-F}$ and $P_{i-T} > P_{j-T}$ and $P_{i-T} < P_{j-F}$, then $P_{i-C} < P_{j-C}$.

- For two template fingerprints q_i and q_j, if P_{i-F}, P_{j-F},P_{i-T} and P_{j-T} are Middle and $P_{i-F} > P_{j-F}$ and $P_{i-T} > P_{j-T}$, then $P_{i-C} > P_{j-C}$.

As shown in Figure 3.5, after applying the fuzzy rules on both candidate lists, the new candidate list is generated as the final list.

3.5 Experiments

To evaluate the proposed partial fingerprint indexing approach, statistical experiments have been carried out on several public databases. Section 3.5.1 describes the databases and the tools used in our experiments. Section 3.5.2 demonstrates

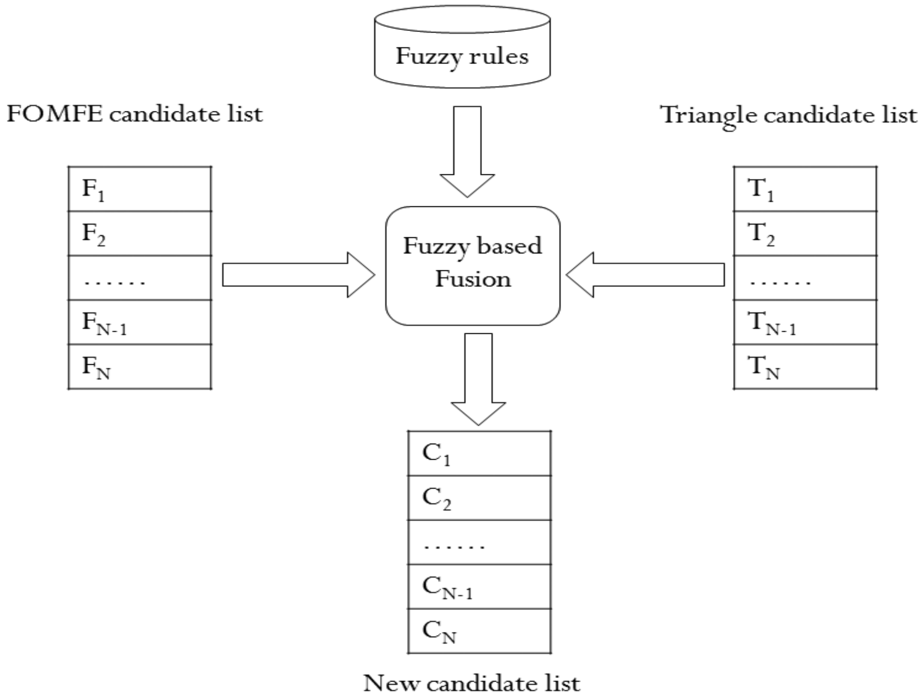


Figure 3.5: Fuzzy based fusion on the candidate lists

the experimental results of minutiae triplets based indexing on both full and partial fingerprint database. Section 3.5.3 demonstrates the experimental results of the proposed fusion indexing scheme on partial fingerprint database. Section 3.5.4 is the computational complexity analysis of the indexing scheme.

3.5.1 Database and Tools

Most of the published techniques for full fingerprint indexing have been evaluated on FVC 2000 DB2a and FVC 2002 DB1a. FVC 2000 DB2a contains 800 fingerprints from 100 subjects (8 impressions per subject) captured using a capacitive fingerprint scanner. FVC 2002 DB1a also contains 800 fingerprints from 100 fingers (8 impressions per finger), but it was captured using an optical fingerprint scanner. In our experiment, we chose the first impression of each subject (100 in total) to form the template database and the rest as the query samples (700 in total) for both FVC 2000 DB2a and FVC 2002 DB1a.

Related works on latent fingerprint matching or indexing have used NIST special database 27 (SD 27) as the query image set, because SD 27 is the only public database available containing mated latent and rolled fingerprints. However, feature extraction in latent fingerprint images is manually done [42][44][34] and is still a challenging problem due to the heavy background noise. We would have to go through a segmentation process first if we want to get the region of interest (ROI) of a latent fingerprint image, which is beyond the scope of this work. Since the objective of our study is partial fingerprint indexing, we used another public database NIST special database 14 (SD 14) [81] in our experiments.

NIST SD 14 is the de facto benchmark database for fingerprint classification and indexing tests. It consists of 54000 ink-rolled prints scanned from fingerprint cards. There are two impressions recorded for each finger, namely, the F (First) prints ranging from F00001 to F27000 and the S (Second) prints ranging from S00001 to S27000. Among the records, most are rolled full prints. The scanned resolution is 500 dpi and the fingerprint image size is 832×768. In our experiments, we chose the last 2000 F prints to constitute the template database and the last 2000 S images as the query samples. Because a large portion of the fingerprint images include impressions of part of the second joints, we first segmented both the template and sample images to remove peripheral regions and make the remainder frame lie in a north-south direction as much as possible. The image size after segmentation is 480×512 pixels.

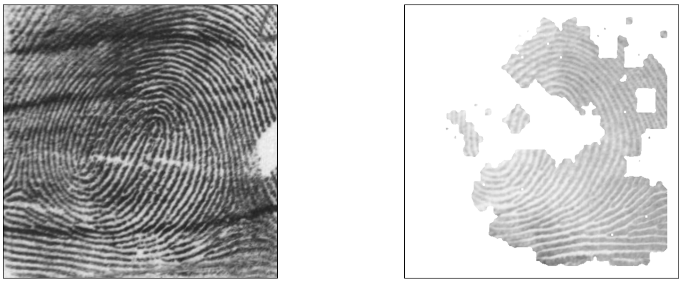
For each sample image, we used a routine of NIST Biometric Image Software [82], namely Mindtct, to obtain a quality map marking reliability of local fingerprint image areas at different levels. Mindtct first extracts the low contrast map, low flow map, and the high curve map which point to different low quality regions of the image. The information in these maps is further integrated into one general map, which contains 5 levels of quality. The quality assigned to a specific block is determined based on its proximity to blocks flagged in these various maps. We extracted an image foreground with the highest quality level and produced a partial fingerprint segment by keeping only the high quality areas. Figure 3.6 shows a typical example of such partial fingerprint images generated in the test and its mated full fingerprint. Therefore, partial fingerprints generated in our experiment do not contain any singularity, and even singularity regions are usually removed. In this way, we can generate a sample image set composed of partial fingerprints.

The performance of the fingerprint indexing scheme is evaluated by reporting the hit rate (HR) at certain penetration rates (PR). We tested the proposed indexing scheme using different parameter settings, including the block size N_B, Fourier extension order k, distortion scale of the triangle sides δ_L, distortion scale of the orientation difference δ_O, the number of fixed candidates K in the list, among which the block size N_B and the Fourier order k were fixed to 8 and 5, respectively, the other parameters were chosen with different values. Detailed explanation and values of these parameters are listed in Table 3.1.

The whole indexing scheme was implemented in Matlab on a workstation PC with the following configurations: Intel(R) Core(TM)i7 3.4GHz, 16GB memory, 64-bit Operating System.

Table 3.1: Major Parameters Used

Symbol	Explanation	Values
N_B	Block size of the orientation field	8
k	Order of the FOMFE coefficients	5
δ_L	Difference of the length of sides	4, 5
δ_O	Difference of the orientation difference	15, 30,60
K	Number of fixed candidates	100, 200, 300



(a) Full fingerprint image

(b) Partial fingerprint image

Figure 3.6: A typical partial fingerprint image in our experiment and its corresponding full image

3.5.2 Evaluation of the Minutiae Triplet based Indexing Scheme

i).Performance on Full Fingerprint Databases

Table 3.2 and Table 3.3 show the performance of the minutiae triplet based indexing approach on FVC 2000 DB2a and FVC 2002 DB1a, respectively, wherein the best performance at a certain penetration rate is highlighted in bold. We can see from these tables that even if the penetration rate is very low (e.g. 1%), the hit rate is high (above 80%). Different choice of δ_L and δ_O results in different performance. For FVC 2000 DB2a, the best choice of δ_L and δ_O is 6 and 15, respectively, and for FVC 2002 DB1a, the best choice of δ_L and δ_O is 5 and 15, respectively.

Fig. 3.7(a) shows the performance comparison of the minutiae triplet based

Table 3.2: Performance Evaluation on FVC 2000 DB2a – Hit Rate

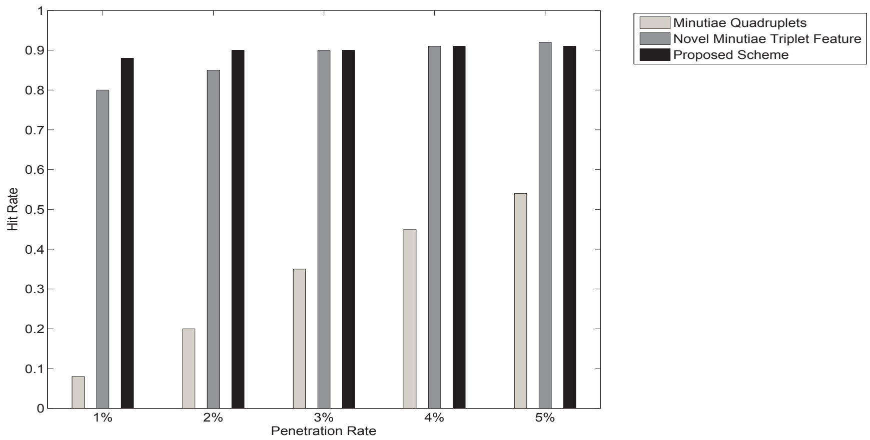
δ_L	δ_O	HR (%)						
		$PR = 1\%$	$PR = 2\%$	$PR = 3\%$	$PR = 4\%$	$PR = 5\%$	$PR = 10\%$	$PR = 20\%$
4	15	85	87	88	89	89	90	92
	30	84	86	87	87	88	90	92
	60	81	85	86	87	87	89	92
5	15	86	88	89	90	**91**	**92**	**94**
	30	85	86	88	89	89	91	**94**
	60	82	84	86	87	88	91	93
6	15	**88**	**90**	**90**	**91**	**91**	**92**	**94**
	30	86	88	89	89	90	**92**	**94**
	60	82	86	88	88	89	91	93

Table 3.3: Performance Evaluation on FVC 2002 DB1a – Hit Rate

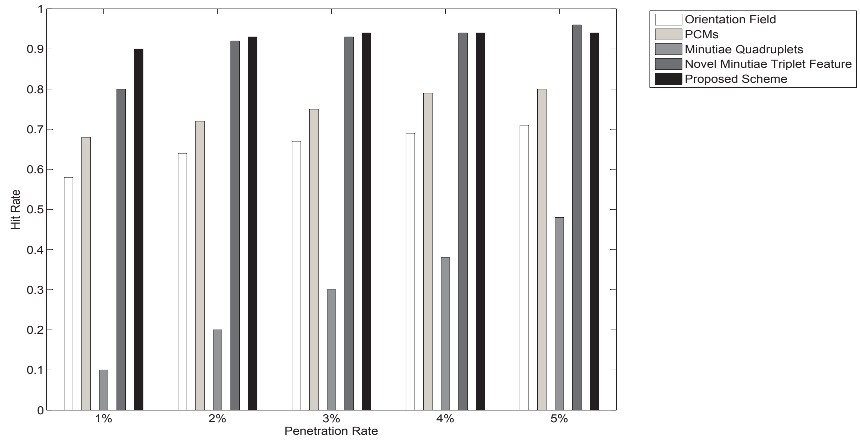
δ_L	δ_O	HR (%)						
		$PR = 1\%$	$PR = 2\%$	$PR = 3\%$	$PR = 4\%$	$PR = 5\%$	$PR = 10\%$	$PR = 20\%$
4	15	89	91	92	92	92	93	95
	30	88	90	91	91	92	93	94
	60	84	87	87	88	89	92	93
5	15	90	**93**	**94**	**94**	**94**	**95**	**96**
	30	89	91	91	92	93	**95**	**96**
	60	85	88	88	89	91	92	94
6	15	**91**	92	93	93	**94**	**95**	95
	30	88	90	91	92	92	93	95
	60	84	86	87	88	89	91	94

indexing in our approach on FVC 2000 DB2a with other methods, including minutiae quadruplets based indexing [32] and indexing with novel minutiae triplet feature [34]. Fig. 3.7(b) shows the performance comparison of the proposed minutiae triplet based indexing on FVC 2002 DB1a with other techniques based on orientation field [19], PCMs [23], minutiae quadruplets [32] and novel minutiae triplet feature [34]. We can see that the proposed indexing scheme outperforms other state-of-the-art methods, especially when the penetration is very low (1% and 2%).

Table 3.4 and Table 3.5 show the results on FVC 2002 DB1a and FVC 2000 DB2a for comparisons with other methods using another measurement, respec-



(a) FVC 2000 DB2a



(b) FVC 2002 DB1a

Figure 3.7: Performance comparison of different indexing schemes on FVC databases

tively, that is the average penetration rate when the hit rate is 100%. We can see from both tables that the proposed minutiae triplet based indexing scheme can achieve much better performance than other method evaluated using the same measurement.

ii).Performance on Partial Fingerprint Database

Fig. 3.8 illustrates the performance improvement of minutiae triplet based fingerprint indexing on NIST SD 14 when the new features are used incrementally.

Table 3.4: Average penetration rate on FVC 2002 DB1a when hit rate is 100%

Minutiae Triplets [27]	38.1%
Low-order Delaunay Triangle [29]	18.1%
Minutiae Quadruplets [32]	11.8%
Novel Minutiae Triplets [34]	9.9%
Proposed Scheme	**3.51%**

Table 3.5: Average penetration rate on FVC 2000 DB2a when hit rate is 100%

SIFT Features [47]	Minutiae Quadruplets [32]	Novel Minutiae Triplets [34]	Proposed Scheme
91%	26%	22%	**5.24%**

In this experiment, the last 2000 F prints ($F25001 \sim F27000$) constitute the template database and 1000 S prints ($S25001 \sim S26000$) are divided into 10 groups as the query samples. Parameters δ_L and δ_O were set to be 4 and 15, respectively. As is shown in Fig. 3.8, the penetration rate decreased by at least 1/3 when the triangle position was used as an extended feature, and the the penetration rate further decreased by at least 1/2 when the orientation difference was used as another extended feature.

Table 3.6 is the performance of the minutiae triplet based indexing approach on NIST SD 14 with different choice of δ_L and δ_O. In this experiment, the last 2000 F prints ($F25001 \sim F27000$) form the template database and the last 2000 S prints ($S25001 \sim S27000$) are used as the query samples. We can see that when δ_L and δ_O are 6 and 30 respectively, the performance is the best (nearly 10%) in this test.

As mentioned before, existing techniques on partial fingerprint indexing approaches were evaluated on NIST SD 27, which need human involvement to extract features. However, the partial sample images used in our experiments are generated from full fingerprint images by erosion and are not used elsewhere, so there is no related comparable work. According to the indexing performance of

Table 3.6: Performance Evaluation on NIST SD 14 – Penetration Rate

δ_L	Penetration Rate (%)		
	$\delta_O = 15$	$\delta_O = 30$	$\delta_O = 60$
4	15	13.28	14.63
5	12.25	11.48	13.29
6	10.25	10.12	10.27

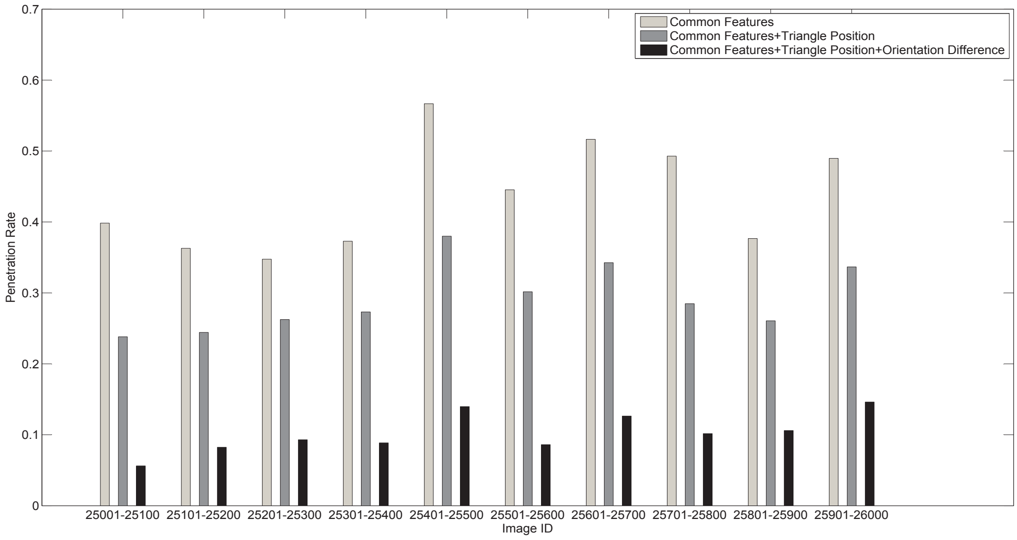


Figure 3.8: Performance improvement of using new features incrementally on NIST SD 14

other methods on full fingerprint databases in Table 3.4 and Table 3.5, we can see that 10% penetration rate is fairly good for partial fingerprint indexing.

3.5.3 Performance of the Proposed Fusion Scheme

Without loss of generality, we choose the penetration rate as the performance measure of our fusion indexing scheme, wherein PR_F, PR_T and PR_C denote the penetration rate of FOMFE coefficients based indexing, minutiae triangle based indexing and the indexing after fusion, respectively.

The performance of the proposed fusion approach on NIST SD 14 is shown in Table 3.7. We can see that the candidate list generated by minutiae triangle

Table 3.7: Performance Evaluation on NIST SD 14 – Penetration Rate

δ_L	δ_O	PR_F	PR_T	PR_C		
				$K = 100$	$K = 200$	$K = 300$
4	15	16.61	15	**12.54**	**12.48**	**12.51**
	30		13.28	**11.38**	**11.33**	**11.53**
	60		14.63	**12.77**	**12.62**	**12.72**
5	15	16.61	12.25	**10.45**	**10.35**	**10.48**
	30		11.48	**10.29**	**10.21**	**10.28**
	60		13.29	**11.92**	**11.71**	**11.71**

PR_F: penetration rate of FOMFE coefficients based indexing only
PR_T: penetration rate of minutiae triangle based indexing only
PR_C: penetration rate after the fusion scheme

based indexing is more reliable, so we use the top K of the candidates in this list as the top candidates in the new candidate list in all the experiments. No matter what values the parameters δ_L, δ_O and K are assigned, the penetration rate after fusion is smaller than that of either FOMFE coefficients based indexing or minutiae triangle based indexing only. For example, when $\delta_L = 4$ and $\delta_O = 15$, the proposed fusion method can improve the penetration rate by around 2.5%, which means that the search space is further reduced by nearly 17% compared to the penetration rate of using minutiae triplet based indexing (15%).

Figure 3.9 and Figure 3.10 illustrate the histograms of the penetration rate when $\delta_L = 4$ and $\delta_L = 5$, respectively. We can see that the number of fixed candidates K in minutiae triangle based list does not influence the final performance much, and the performance is better when $K = 200$ in all parameter settings.

3.5.4 Computational Complexity Analysis

According to [12], the computational efficiency of the FOMFE can be greatly improved when it runs on a large fingerprint database, because $\Psi = \{\psi_{ij}(x, y)\}$ in Eq. (3.1) is only related to the coordinate variables, which means Ψ will be the same for various inputs if x and y are the same. In this way, Ψ becomes a common template that in fact only needs to be calculated once for all inputs. Since x and y are generated on lattice indices, coarse OF inputs with the same dimensions can share a common template of Ψ. Therefore, QR factorization can be brought forward to the beginning even before the modeling process. The required terms are then simply passed to the subsequent routines for evaluating the coefficients and reconstructing the OF individually.

Before Smooth Extension, the FOMFE is used to refine the coarse OF of the partial fingerprint and generate the initial coefficients for Smooth Extension. The process consists of two parts: coefficients estimation and OF reconstruction. The computational complexity of the coefficients estimation is $O(2M_\Omega K^2)$ [12], where M_Ω is the number of valid blocks from the coarse OF and $K^2 = (2k+1)^2$ is the number of coefficients in the FOMFE. It is obvious that a larger block size N_B results in a smaller M_Ω and, thus, a faster computation. The cost of reconstruction is $O(M_\Omega K^2)$ for evaluating the K^2 basis functions at each valid block in the OF [12]. Therefore, the total theoretical cost of the FOMFE before Smooth Extension is $O(2M_\Omega K^2) + O(M_\Omega K^2)$.

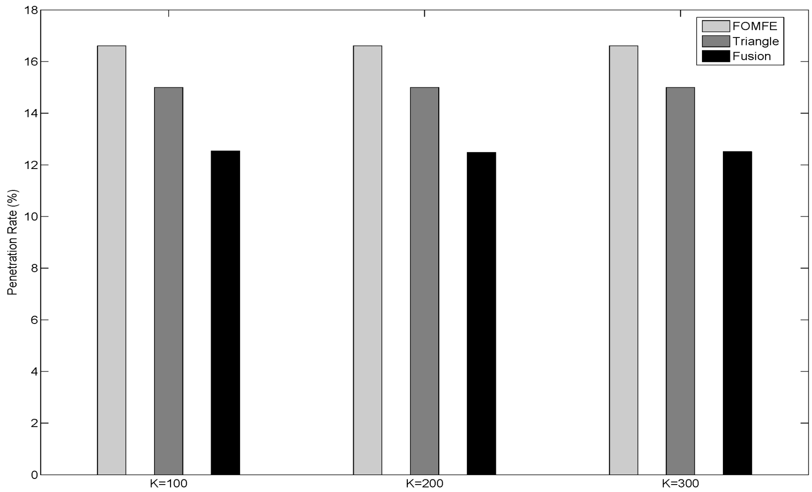
Similarly, the computational complexity can also be estimated for Smooth Extension. During Smooth Extension, the orientations of unknown blocks M_χ are estimated step by step. For each step m in the Smooth Extension, the cost for coefficients estimation is related to the total number of blocks already reconstructed in previous $m-1$ steps and the blocks in the m-th extended band, that is $O(2M_{\Omega^{(m)}}K^2)$. So the total cost for coefficients estimation during Smooth Extension is $\sum_{m=1}^{t-1} O(2M_{\Omega^{(m)}}K^2)$. For OF reconstruction, no matter how many steps it takes to fill the unknown region χ with phase estimates, the total cost for reconstruction is $O(M_\chi K^2)$. Therefore, the total cost for generating the global feature set is $O(2M_\Omega K^2) + O(M_\Omega K^2) + \sum_{m=1}^{t-1} O(2M_{\Omega^{(m)}} K^2) + O(M_\chi K^2)$. Since $\Omega^{(0)} = \Omega$ and $M = M_\Omega + M_\chi$ is the total number of blocks in the partial fingerprint image, the total cost can be represented as $\sum_{m=0}^{t-1} O(2M_{\Omega^{(m)}} K^2) + O(MK^2)$. After Smooth Extension, the final coefficients are generated. The indexing using FOMFE coefficients has been proved to be much faster than that of using raw OF [12].

For indexing based on local minutiae triplet features, we use features from all minutiae triplets instead of those of Delauney triangles because Delauney triangles are not stable in partial fingerprint images. Therefore, it is unavoidable that the number of triangles considered is very large. Fortunately, the number of minutiae in a partial fingerprint image is less than that in a full fingerprint image, and we have adopted some strategies to reduce the average percentage of hypotheses that need to be considered for indexing: (1) before the construction of triangles, minutiae with short distances to their neighbors are filtered out; (2) we use quantization to retrieve points only in related bins, not to compare all the points in the feature space for each query; (3) the new features are easy to calculate and can filter out most of the points in each bin on average. Compared to the well-known full minutiae based indexing approach in [27], our indexing scheme based on minutiae triplets is much less complex.

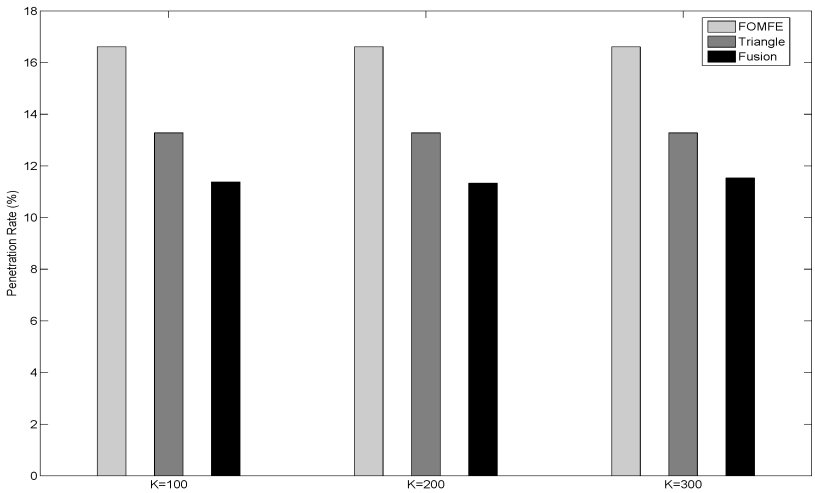
The fusion scheme to generate the final candidate list is only related to the number of templates N in the database and the number of iterations is N, so the computational cost is $O(N)$. In addition, the FOMFE coefficients based indexing and the minutiae triplets based indexing are independent, so they can be conducted in parallel, which can further speed up the retrieval process.

3.6 Summary

In this chapter, we proposed to combine a reconstructed global feature, namely FOMFE coefficients, with local minutiae triplet based features for partial fingerprint indexing. At first, the minutiae triplet based indexing scheme and FOMFE coefficients based indexing scheme are applied separately to generating two candidate lists: reconstructed FOMFE coefficients are used to form the global feature space directly; newly designed feature vectors on minutiae triplets are adopted to form the local feature space. Before getting the final candidate list, a training process is conducted on a small portion of the query fingerprints to decide which candidate list tends to be more reliable. Based on the reliability, we derive a set of fuzzy rules to guide the fusion of the two candidate lists for generating the final candidate list. Evaluation on several public databases show that the performance of minutiae triplet based indexing is improved significantly after new features are considered, and can be comparable to that on full fingerprint indexing under certain parameter settings. Experiments on the last 2000 F and S fingerprint images in NIST SD 14 show that the penetration rate will decrease after applying the fusion scheme, which means that the searching space will be further reduced before matching.

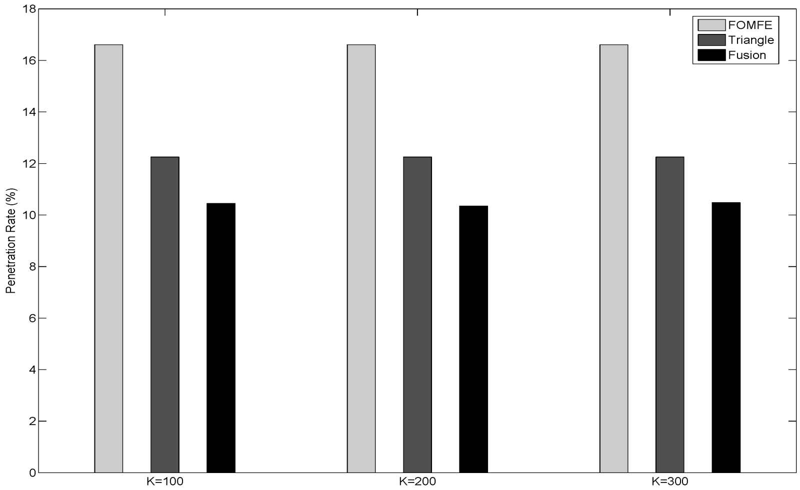


(a) $\delta_L = 4$, $\delta_O = 15$

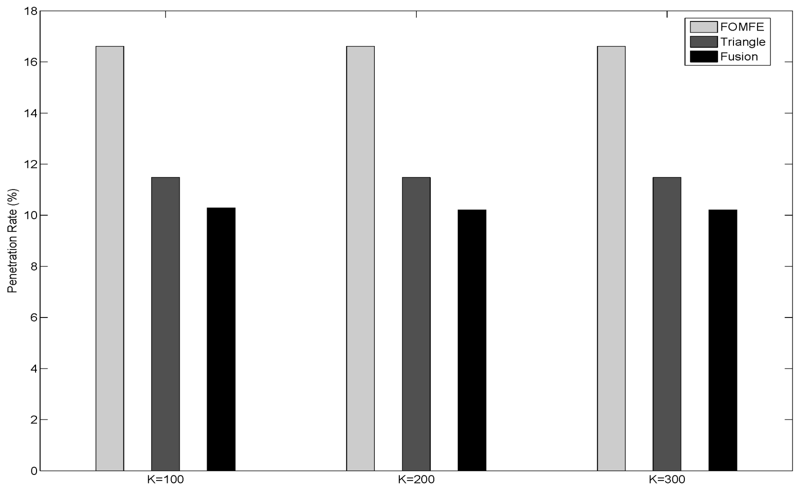


(b) $\delta_L = 4$, $\delta_O = 30$

Figure 3.9: Penetration Rate: block size $N_B = 8$, Fourier extension order $k = 5$, distortion scale of the triangle sides $\delta_L = 4$



(a) $\delta_L = 5$, $\delta_O = 15$



(b) $\delta_L = 5$, $\delta_O = 30$

Figure 3.10: Penetration Rate: block size $N_B = 8$, Fourier extension order $k = 5$, distortion scale of the triangle sides $\delta_L = 5$

Chapter 4

Multi-Sensor Fingerprint Database Collection and Evaluation

4.1 Motivation and Contributions

Conventionally, fingerprints were captured using contact-based methods, e.g., ink, thermal, optical, capacitive, ultrasonic, etc. In these cases, the subjects have to press or roll their fingers against a solid surface with force to get 2D fingerprint images. As a consequence, this capturing scheme often introduces degraded images due to skin deformation, nonuniform pressure and residue left on the sensor surface. To overcome these problems, touchless fingerprint imaging technology has been proposed, particularly, the 3D fingerprint acquisition techniques have drawn more and more attention recently. TBS North America [7] has developed a 3D fingerprint system that uses shape from shading and stereovision based technique to obtain 3D fingerprints in a non-contact fashion. Another representative system was developed by Flashscan3D LLC [8] and the University of Kentucky. Their system uses structured light illumination (SLI) technique and can capture the 3D ridge-valley details of the fingertips. All these 3D fingerprint sensors produce unraveled 2D equivalent fingerprints at last to be compatible with legacy 2D fingerprints.

Despite the development of 3D fingerprint technology, preliminary experiments have been carried out with limited individually collected samples [69][83], instead of a publicly available 3D fingerprint database. This is a great barrier to experimental validation and comparison of algorithms in 3D fingerprint biometric research area. We therefore built a multi-sensor fingerprint database with 3D fingerprints as well as their corresponding 2D fingerprints from 150 volunteers (subjects). The large size of the database we have established will provide meaningful statistical analysis and a truthful assessment of the performance of the state-of-the-art algorithms in this area. Besides, our database can serve as a standard database for developing identification techniques for 2D to 3D fingerprint images. The resolution of these identification issues will require an innovative approach which will significantly advance research in the area of biometrics. It will also lead to the improvement and development of important commercial products.

The rest of this chapter is organized as follows: Section 4.2 is a brief description of the database, including the acquisition protocol, the problems encountered during the acquisition process, the naming scheme of the database and the validation steps. Evaluations on this database using original 3D fingerprint images and enhanced 3D fingerprint images are demonstrated in Section 4.3 and Section 4.4, respectively. Section 4.5 concludes the whole work in this chapter.

4.2 Collection of 2D and 3D Fingerprint Database

4.2.1 Acquisition Protocol

The acquisition of the 2D and 3D fingerprint database was carried out in three universities in Australia: the University of New South Wales at Canberra, Latrobe University at Melbourne, and Deakin University at Melbourne. The institution in charge of coordinating the acquisition process was the University of New South Wales at Canberra.

A total of 150 subjects were recruited from students and staff with balanced demographic characteristics regarding age, gender, nationality similar to BMDB [84]: 45% of the subjects were between 18 and 25 years of age, 45% between 25 and 35, and the remaining 10% of the subjects were above 35 years old; the gender distribution was balanced with only a 10% difference between male and female subjects; 45% of the subjects were East Asians, 45% were Indians or

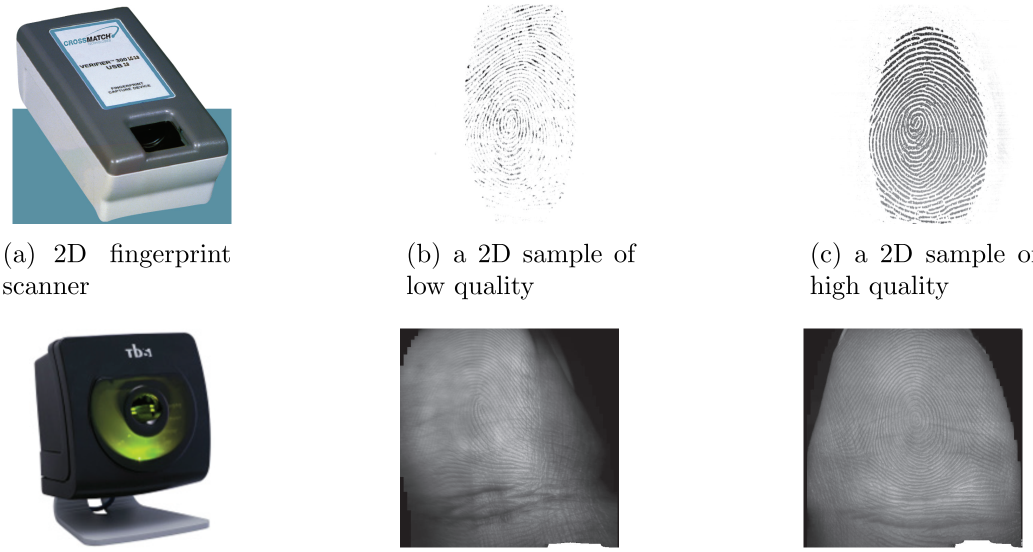
Bangladeshis, and the remaining 10% of the subjects were Caucasians.

To protect the privacy of all subjects, the name of each subject was not recorded; instead a random number was assigned as the only ID for each subject. All the relevant non-biometric data of each subject was stored in an independent text file, which will not be published with the database.

Personal information and captured fingerprint data are personal data, for which we have got approval from the Human Research Ethics Committee (HREC) at the University of New South Wales. At the start of the acquisition a Participant Information Statement and Consent Form was signed by each subject. In the Participant Information Statement and Consent Form, the subjects were properly informed about how personal information would be used, that the fingerprint data would be released in public anonymously, and that it was unlikely that they would be identifiable in the future. The acquisition procedure could start only when this consent form was fully understood and signed by the subject.

Two samples (BMP format) of ten fingers of each volunteer were collected using a 3D fingerprint scanner; four samples (BMP format) of ten fingers of each volunteer were collected using a 2D fingerprint scanner. The devices involved and some representative examples are shown in Fig. 4.1: Fig. 4.1(a) is the 2D fingerprint scanner; Fig. 4.1(b) is an example of its output image in low quality due to the dry finger surface; Fig. 4.1(c) is an example of 2D fingerprints which has clear ridge lines; Fig. 4.1(d) is the 3D fingerprint scanner; Fig. 4.1(e) is a typical 3D fingerprint of low quality because the contrast of the left part of the fingerprint is very low; and Fig. 4.1(f) is an example of 3D fingerprints in high quality.

The scenario for the acquisition was an office-like environment with neutral illumination which had no preponderant focuses. The acquisition was carried



(a) 2D fingerprint scanner

(b) a 2D sample of low quality

(c) a 2D sample of high quality

(d) 3D fingerprint scanner

(e) a 3D sample of low quality

(f) a 3D sample of high quality

Figure 4.1: Devices used and corresponding captured samples

out using a workstation PC which ran the acquisition software and two scanners were connected to the PC via the USB interface 2.0. The donors sat in a chair in front of the sensor. Acquisition was managed by a supervisor, who sat in another chair next to the donor and was in charge of the following activities: giving necessary instructions to the donors so that the acquisition protocol will be followed; manipulating the operation of the software and inputting relevant data for each donor; manually verifying the samples to decide whether to discard or store them. The main features of the devices and software are listed in Table 4.1.

(a) a noisy 2D sample (b) a noisy 3D sample

Figure 4.2: Noisy examples captured during acquisition

Table 4.1: Acquisition devices and their main features

Modality	Model	Main Features
3D Fingerprints	TBS S120E	Software: TBS 3DCaptureSuite_v10018 DPI: around 1000 Image size: $1024 * 1280$ Storage size: 1.25MB
2D Fingerprints	CROSSMATCH Verifier 300 LC2.0	Software: FingerprintVerificationSDK 1.0 DPI: around 500 Image size: $640 * 480$ Storage size: 304KB

4.2.2 Naming and Validation of Acquired Data

i).Problems encountered during the acquisition

During the acquisition, several problems encountered are listed as follows:

- The residual left on the 2D fingerprint scanner affected the next capture, especially when the former finger was wet. An example of the captured 2D fingerprints is shown in Fig. 4.2(a). Besides, some 2D fingerprints were very hard to capture because their corresponding fingers were too dry.

- The 3D fingerprint scanner was too sensitive: even if the finger pose was not right, the device still accepted it leading to unacceptable scans sometimes. Therefore, certain volunteers had to try many times to get the right scans.

- Some volunteers have short ring fingers and little fingers, so their corre-

Table 4.2: Finger ID

Fingers	**ID**
Right Thumb	1
Right Index	2
Right Middle Finger	3
Right Ring Finger	4
Right Little Finger	5
Left Thumb	6
Left Index	7
Left Middle Finger	8
Left Ring Finger	9
Left Little Finger	10

sponding 3D fingerprints could not be captured.

- The 3D fingerprint scanner introduced some noise when the room temperature was too high. An example of the captured 3D fingerprints is shown in Fig. 4.2(b). It can be noticed that the peripheral region of the fingerprint is not shown, so some post-processing techniques are needed to remove the noise.

ii).Naming of the database

The proposed database consists of two sub-databases, one for 2D fingerprints, the other for 3D fingerprints. The naming for both sub-databases are the same, that is *Subject ID_Finger ID_Capture Order*. The Finger ID is listed in Table 4.2.

There are in total 150 volunteers, so the Subject ID ranges from 1 to 150. For each subject, the Finger ID ranges from 1 to 10 according to Table II. The Capture Order can be 1 and 2 for 3D fingerprints, and 1,2,3, and 4 for 2D fingerprints. For example, the fingerprint named 102_3_2 in 3D sub-database represents the second capture of the right middle finger of subject 102.

iii).Validation of the database

Although the database was carefully collected by human supervisors, there were still possible errors caused by software or humans. In order to ensure that the database was conformed to the acquisition protocol, all the acquired samples were manually verified by a human supervisor. The samples non-compliant with the acquisition protocol were either corrected or removed. Note that valid low-quality samples and invalid samples are different: low-quality samples are acceptable as long as the acquisition protocol was followed (e.g., dry or wet 2D fingerprint images), and these samples were not removed from the database since they represent real-world samples that can be found in the normal use of a fingerprint biometric system. On the other hand, invalid samples are those that do not comply with the specifications given for the database (e.g., blurred 3D fingerprint images).

The first stage of validation was carried out during the acquisition process itself. Human investigators were in charge of validating every captured sample and recapturing if it did not meet the specified quality standards. After completion of the acquisition, a second validation step was carried out again manually by an investigator:

- Checking the names of all the samples using a read image routine, for example, image named 30-2-4 would be replaced by name 30_2_4.
- Replacing a blank 2D fingerprint image (the fingerprint was not saved successfully) by a sample of the same finger but of a different capture order.
- Marking missing 3D fingerprint images (ring or little fingers are too short to be scanned by the 3D fingerprint scanner) and recording it down in the instruction document.

4.3 Verification Experiments On Raw 3D Images

The output of the 3D fingerprint sensor are unraveled 2D equivalent fingerprint images, we call it original raw 3D images. To evaluate the recognition performance, we have conducted three groups of testing on original raw images in Section 4.3.1: (i) one for 2D fingerprint verification, (ii) one for 3D fingerprint verification, and (iii) one for 2D to 3D fingerprint verification. We used a commercial fingerprint identification software VeriFinger SDK [79] to extract all the minutiae from each fingerprint image and get the match score for each pair of testing fingerprints. Two different protocols are used to report the verification performance of the established database:

- The original FVC protocol [85]: each template is compared against the remaining ones of the same finger to obtain the False Non Match Rate (FNMR). The first template of each finger is compared against the first template of the remaining fingers in the data set, to determine the False Match Rate (FMR).

- The modified 1vs1 protocol: the first template of each finger is compared against the second one of the same finger to obtain the FNMR. The first template of each finger is compared against the second template of the remaining fingers in the data set, to determine the FMR.

Moreover, to enhance the contrast of the original raw 3D fingerprint images and reduce the peripheral noise in these images, we carried out two sets of comparison experiments on enhanced original raw 3D fingerprint images in Section 4.3.2.

4.3.1 Verification Experiments Using Original Raw Images

i).2D fingerprint verification

In this test, only part of the 2D fingerprint database was chosen, in particular, the first capture and the second capture of six fingers (right thumb, right index, right middle finger, left thumb, left index and left middle finger) of each subject constituted the testing set. In the original FVC protocol, there are in total 900 (6×150) comparisons that should be genuine match, 150 each for right thumb, right index, right middle finger, left thumb, left index, and left middle finger, and there are in total 67050 ($6 \times 150 \times 149/2$) comparisons that should be false match, 11175 each for right thumb, right index, right middle finger, left thumb, left index, and left middle finger. In the modified 1vs1 protocol, similar to that in the original protocol, there are in total 900 (6×150) comparisons that should be genuine match, and there are in total 134100 ($6 \times 150 \times 149$) comparisons that should be false match, 22350 each for right thumb, right index, right middle finger, left thumb, left index, and left middle finger.

ii).3D fingerprint verification

Similar to the 2D testing, all the 3D captures (the first and the second) of six fingers (right thumb, right index, right middle finger, left thumb, left index and left middle finger) of each subject were chosen in this test. In the original FVC protocol, there are in total 900 (6×150) comparisons that should be genuine match, and there are in total 67050 ($6 \times 150 \times 149/2$) comparisons that should be false match. In the modified 1vs1 protocol, there are in total 900 (6×150) comparisons that should be genuine match, and there are in total 134100 ($6 \times$

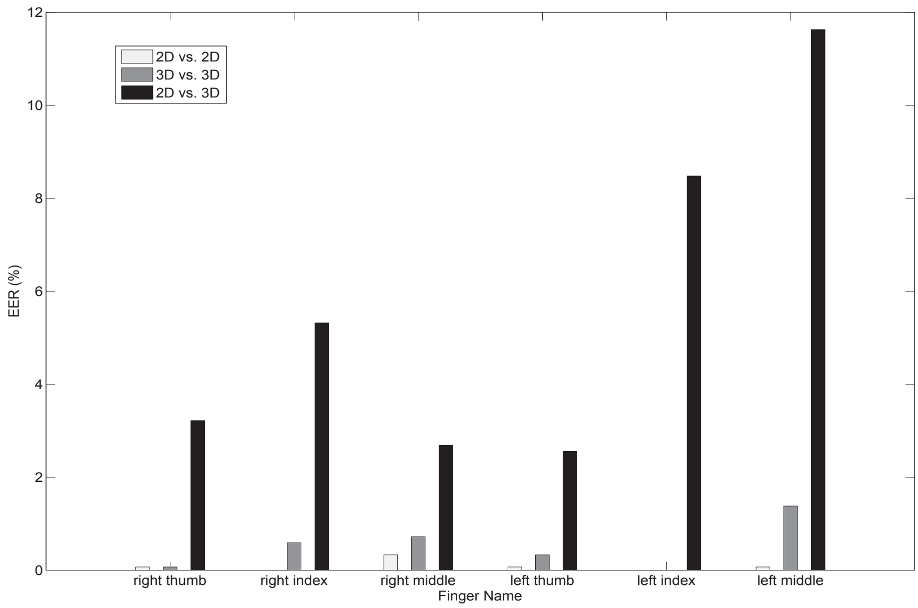


Figure 4.3: Verification performance using the original FVC protocol

150×149) comparisons that should be false match.

iii).2D to 3D fingerprint verification

In this test, the first capture of six fingers (right thumb, right index, right middle finger, left thumb, left index and left middle finger) of each subject in both 2D and 3D fingerprint databases were chosen, so the original FVC protocol is not suitable for this test. In the modified 1vs1 protocol, there are in total 900 (6×150) comparisons that should be genuine match, 150 each for right thumb, right index, right middle finger, left thumb, left index, and left middle finger, and there are in total 134100 ($6 \times 150 \times 149$) comparisons that should be false match, 22350 each for right thumb, right index, right middle finger, left thumb, left index, and left middle finger.

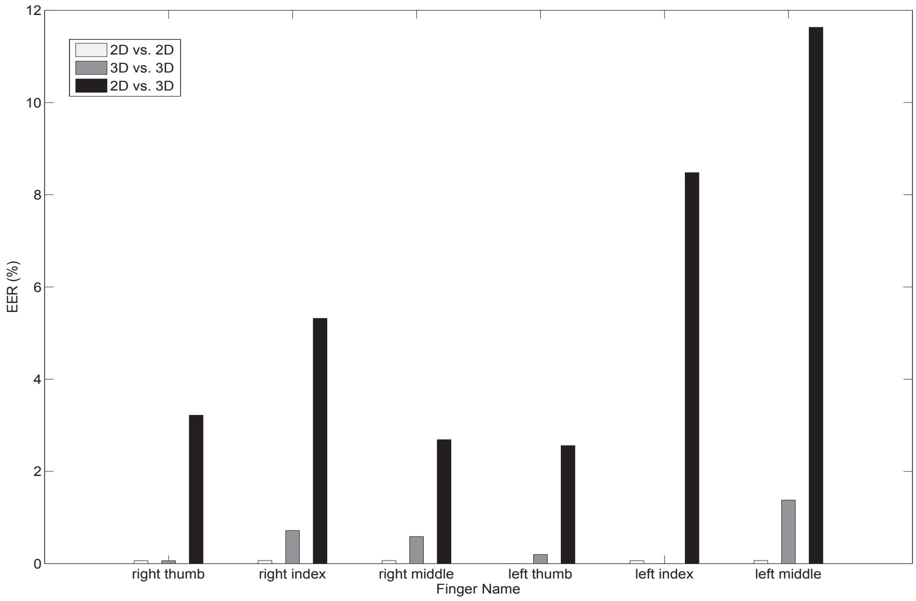


Figure 4.4: Verification performance using the modified 1vs1 protocol

iv).Results

We adopted the Equal Error Rate (EER) as a measure of our verification performance. EER is the value where FMR and FNMR are equal and is the best single description of the error rate of an algorithm. The lower the EER, the better the algorithm.

Fig. 4.3 and Fig. 4.4 show the Equal Error Rate for each group in the above three tests using the original FVC protocol and the modified 1vs1 protocol, respectively. We can see from both figures that the average EER is less than 0.1% for 2D fingerprint verification, around 0.5% for 3D fingerprint verification, and around 5.67% for 2D to 3D fingerprint verification. It is obvious that, for each group, the verification performance of 2D to 2D fingerprints is better than or comparable to that of 3D to 3D fingerprints, and the performance of 2D to 3D fingerprint verification is the worst. In other words, the matching rate is not satisfactory for 2D to 3D fingerprint identification when using the raw 3D fingerprint images and the commercial fingerprint identification software VeriFinger. Therefore, post-processing algorithms for 3D fingerprint images as well as more sophisticated 3D matching algorithms will be very important to improve the performance of 2D to 3D fingerprint identification.

4.3.2 Verification Experiments Using Post-processed Original Raw Images

i).Testing on contrast enhanced images

As is mentioned before, the contrast of the 3D fingerprint images is low compared to contact-based 2D fingerprint images, we therefore enhanced the contrast of the 3D fingerprint images by transforming the contrast values using contrast-limited adaptive histogram equalization (CLAHE) [86]. Using CLAHE, the contrast enhancement can be limited in order to avoid amplifying the noise which might be present in the image. Then we tested the performance of 3D to 3D fingerprint verification and 2D to 3D fingerprint verification using the enhanced 3D fingerprints.

ii).Testing on cropped images

Since the fingerprint images produced by the 3D scanning device is larger than those of the traditional 2D sensor, we cropped all the raw 3D fingerprint images by removing the peripheral regions. The size of the cropped 3D images is around 480×560. Then we tested the performance of 3D to 3D fingerprint verification and 2D to 3D fingerprint verification using the cropped 3D fingerprints.

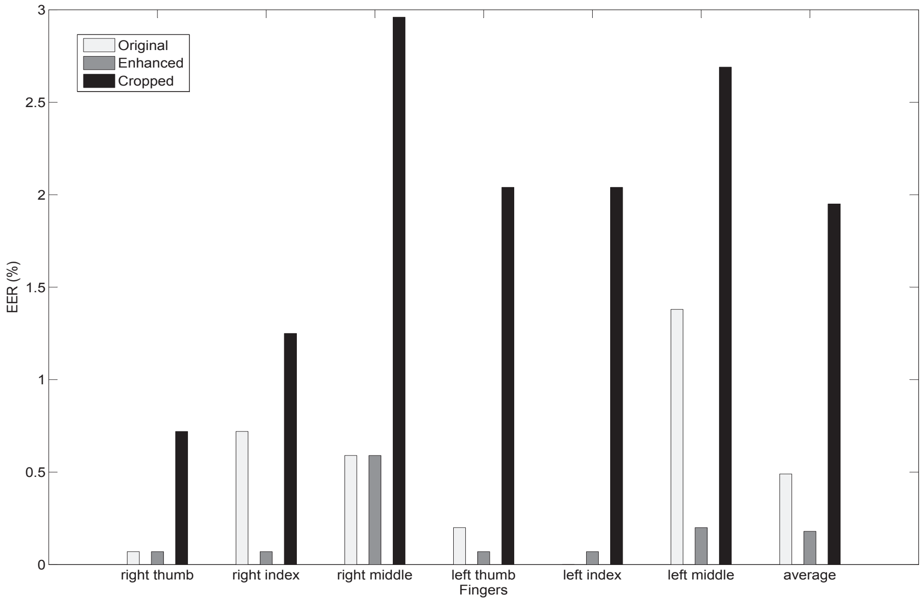


Figure 4.5: Performance of 3D to 3D fingerprint verification

iii).Results

Fig. 4.5 shows the performance comparison of 3D to 3D fingerprint verification using original raw 3D images, contrast enhanced raw 3D images and cropped raw 3D images, respectively. We can see that the performance for all fingerprints except the left index fingerprints is improved after contrast enhancement, especially the right index fingerprints, left thumb fingerprints and left middle fingerprints, and the average EER for 3D to 3D fingerprint verification using contrast enhanced images is comparable to that of the 2D to 2D fingerprint verification. However, as shown in Fig. 4.5, cropping 3D images cannot improve the performance of 3D to 3D fingerprint verification, because the EER for 3D to 3D fingerprint verification using cropped 3D images are much larger than that of using the original raw 3D images.

Fig. 4.6 shows the performance comparison of 2D to 3D fingerprint verification using original raw 3D images, contrast enhanced 3D images and cropped 3D

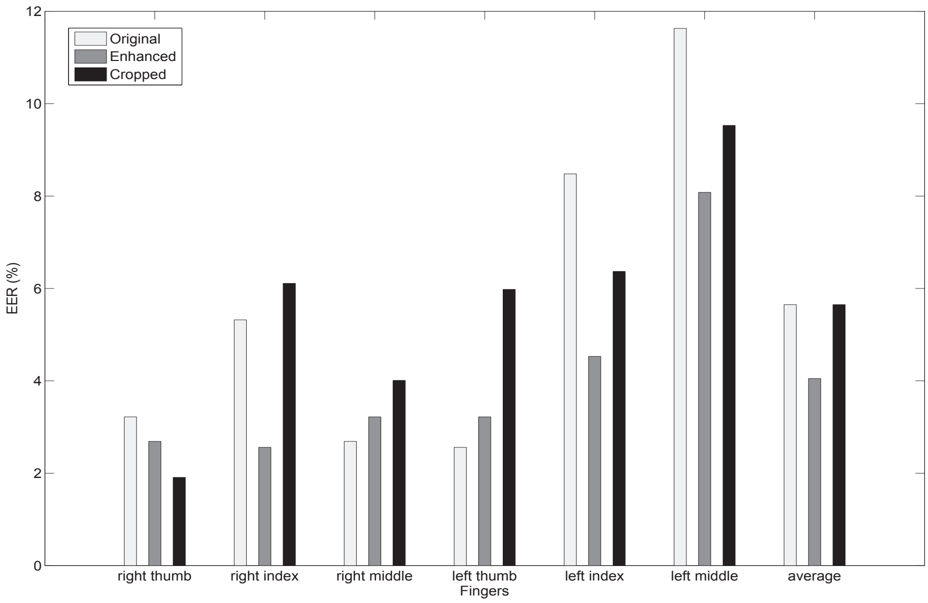


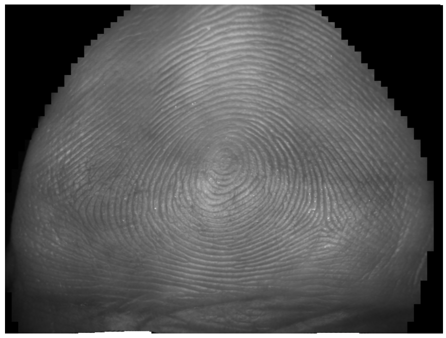
Figure 4.6: Performance of 2D to 3D fingerprint verification

images, respectively. It is obvious that both contrast enhancement and cropping cannot improve the performance of 2D to 3D fingerprint verification too much, and compared to the performance of 2D to 2D fingerprint verification and 3D to 3D fingerprint verification, the performance of 2D to 3D fingerprint verification using VeriFinger is far from satisfactory.

4.4 Verification Experiments On Enhanced 3D Images

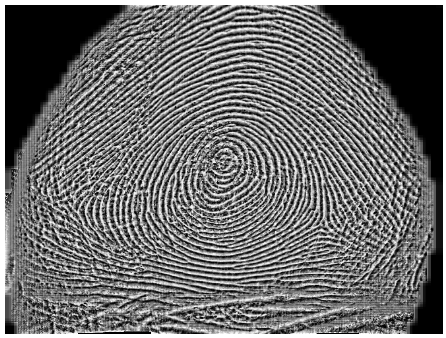
The raw 3D (unraveled 2D equivalent) fingerprint images were further processed with algorithms 'TH6' and 'R414' provided by TBS 3DCaptureSuite. Fig. 4.7 and Fig. 4.8 show the typical fingerprint images of good quality and bad quality (many creases captured by the 3D scanner) in the database, respectively, wherein Subfigure (a) is the plain 2D fingerprint captured by the traditional scanner,

(a) 2D fingerprint image



(b) 2D equivalent fingerprint image generated by 3D scanner

(c) Post-processed image in (b) by algorithm HT6



(d) Post-processed image in (b) by algorithm R414

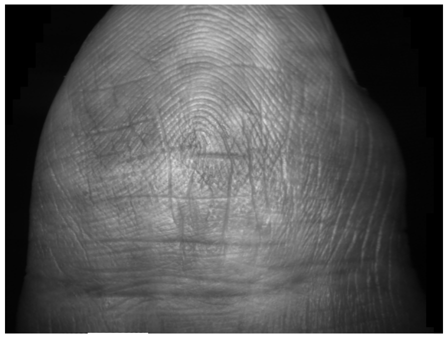
Figure 4.7: Representative good quality images in the database

Subfigure (b) is the unraveled 2D equivalent fingerprint image produced by the 3D scanner, Subfigure (c) and (d) are the post-processed images of Subfigure (b) by algorithm 'TH6' and 'R414', respectively.

4.4.1 Comparison Experiments Between Different Post-process Algorithms

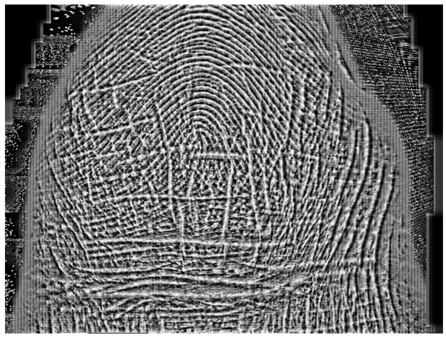
To test the performance of different enhancement algorithms, we have conducted several experiments on a subset of the fingerprint database: fingerprints from 5 subjects, one of good quality, one of bad quality, and the others are of general

(a) 2D fingerprint image



(b) 2D equivalent fingerprint image generated by 3D scanner

(c) Post-processed image in (b) by algorithm HT6



(d) Post-processed image in (b) by algorithm R414

Figure 4.8: Representative bad quality images in the database

quality.

Therefore, there are in total 100 ($5 \times 10 \times 2$) 2D plain fingerprints, 100 unraveled 2D equivalent fingerprint images, 100 post-processed images using algorithm 'TH6', and 100 post-processed images using algorithm 'R414'.

To investigate the performance of 2D to 3D fingerprint recognition, we divided the database into several groups by subjects or by fingers and conducted a series of experiments. The recognition tool we used is also the Neurotechnology's commercial software Verifinger [79], which can also help extract and present the main features of fingerprints (singular points and minutia points) in a graphical interface.

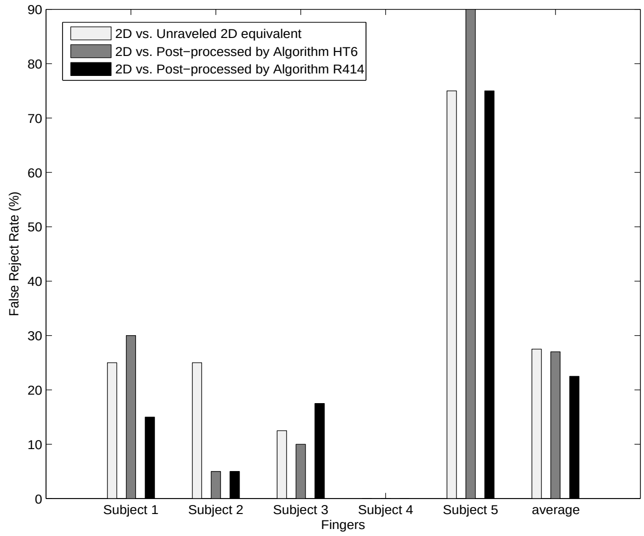


Figure 4.9: Recognition performance regarding different subjects

i).Testing by subjects

In this test, there are 5 groups (5 subjects in total) and each group contains 20 (10×2) 2D fingerprints, 20 unraveled 2D equivalent fingerprint images, 20 post-processed images using algorithm 'TH6', and 20 post-processed images using algorithm 'R414'. We use False Reject Rate (FRR) to evaluate the recognition performance. The lower the FRR, the better the performance. Fig. 4.9 demonstrates the performance regarding each subject in three scenarios: 2D to unraveled 2D equivalent images, 2D to post-processed images using algorithm 'TH6', and 2D to post-processed images using algorithm 'R414'.

As is shown in Fig. 4.9, the recognition performance of Subject 4 is the best because the FRR are all 0 in three scenarios; the performance of Subject 5 is the worst since the FRR are all very high in three testings, and the FRR even reaches 90% when identifying 2D to post-processed images using algorithm 'TH6'. Possible reason for the large difference is that there are too many creases (bad

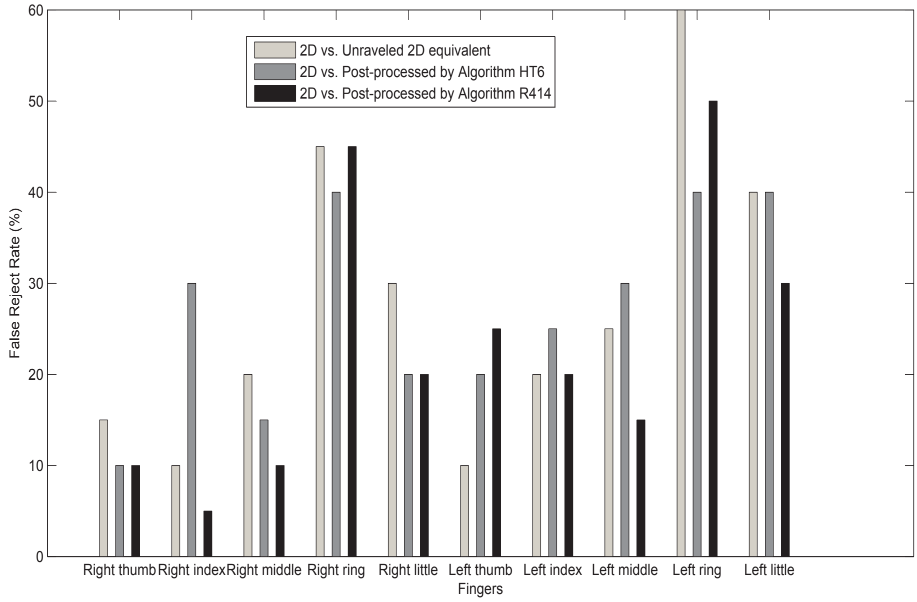


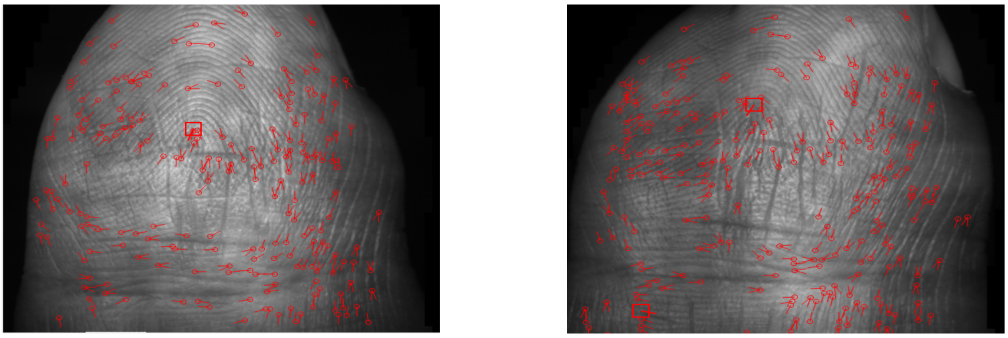
Figure 4.10: Recognition performance regarding different finger names

quality) in the fingerprint images of Subject 5 captured by the 3D scanner, but the ridges and valleys are very smooth (good quality) in the fingerprint images of Subject 4.

The average FRR, which demonstrates the performance of the whole database, is above 20% in all three scenarios, so the performance of 2D to 3D recognition is not good regarding the database we have collected. Meanwhile, we can see from the average FRR that, using Algorithm 'R414' to process unraveled 2D equivalent images can improve the performance of 2D to 3D recognition since the FRR in this scenario is lower than those in other two cases.

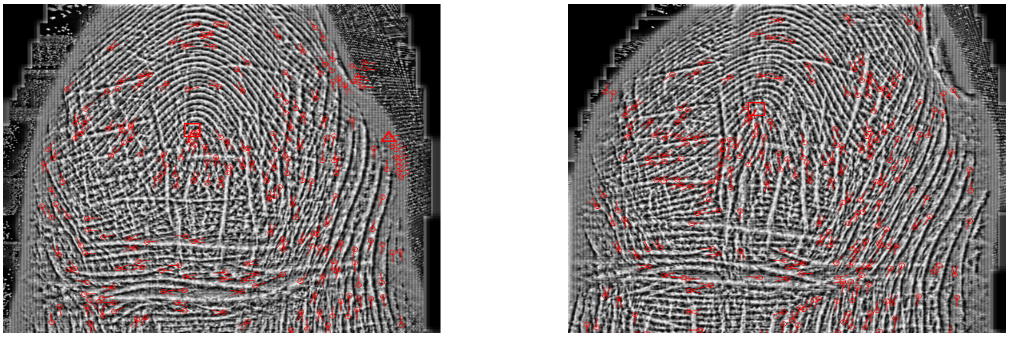
ii).Testing by finger names

In this test, there are 10 groups regarding left thumb finger, left index finger, left middle finger, left ring finger, left little finger, right thumb finger, right index finger, right middle finger, right ring finger, and right little finger, and each group contains 10 ($5 \times 1 \times 2$) 2D fingerprints, 10 unraveled 2D equivalent fingerprint



(a) (b)

Figure 4.11: Two captures of the same finger using the same 3D scanner



(a) (b)

Figure 4.12: Corresponding post-processed images in Fig. 5 using Algorithm 'R414'

images, 10 post-processed images using algorithm 'TH6', and 10 post-processed images using algorithm 'R414'. We also use False Reject Rate (FRR) to evaluate the recognition performance. Fig. 4.10 demonstrates the performance regarding different finger names in three scenarios: 2D to unraveled 2D equivalent images, 2D to post-processed images using algorithm 'TH6', and 2D to post-processed images using algorithm 'R414'.

As is shown in Fig. 4.10, the recognition performance of ring fingers is the worst since the FRR regarding both left ring and right ring fingers are all very high in three scenarios, the FRR reaches 60% when identifying 2D to unraveled 2D equivalent images for the left ring fingers; the recognition performance of little fingers is a little better than that of the ring fingers, but still not good. Relatively speaking, the performance of identifying the thumb fingers is the best, which may be due to the large region and smooth surface of the thumb fingers.

iii).Testing of 3D to 3D fingerprint recognition

We also tested the performance of 3D to 3D fingerprint recognition, in particular, we try to verify two unraveled 2D equivalent images of the same finger captured consecutively using the same 3D scanner. There are in total 50 (5×10) pairs of fingerprints to be verified by VeriFinger. The results show that not all pairs can be matched successfully, for example, the two fingerprints in Fig. 4.11 cannot be verified as the same fingerprint, and their corresponding post-processed images in Fig. 4.12 cannot be matched too.

The singular points and minutiae in Fig. 4.11 are marked red by VeriFinger, we can see that the area of the fingerprint in Fig. 4.11(b) is wider than that in Fig. 4.11(a) and the number of minutiae in Fig. 4.11(b) is larger than that in Fig. 4.11(a). Actually, there are too many spurious minutiae in both images, especially near the brim of the fingerprints, certain minutiae near the singular area are missing, and the extracted singular point in Fig. 4.11(b) deviates obviously from the ground truth. The same observations apply to images in Fig. 4.12 too. All these may result from the difference between the finger poses of two captures and the creases on the fingerprints.

iv).Discussion

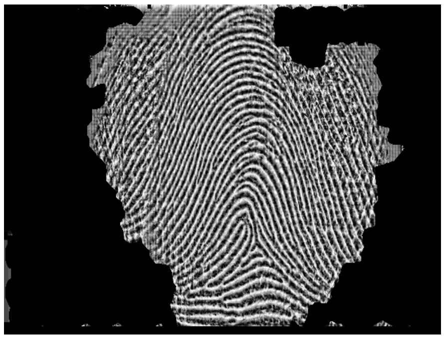
According to the above testing, we can see that, despite the advantages, 3D fingerprint technology is new and also has some drawbacks:

- the image resolution is not uniform and the contrast between the ridges and

(a) Enhanced 3D fingerprint image by algorithm R414

(b) Cropped image based on Singular Points

(c) Cropped image based on image quality



(d) Cropped image based on orientation reliability

Figure 4.13: A representative 3D image and its cropped ones by 3 different methods

the valleys is low in unraveled 2D equivalent fingerprint images, which will impact the feature extraction process.

- the scanner is too sensitive to the pose of the finger, little difference will lead to recognition failure.
- creases have a significant influence on the unraveled images and the subsequent post-processed images.

4.4.2 Comparison Experiments on Post-processed Enhanced Images

From the previous experiments, we can conclude that the enhanced 3D fingerprint images using algorithm 'R414' have the best performance, so we will use them as the 3D images in the later experiments. However, those images have too many spurious minutiae that will impact the feature extraction, especially at the peripheral region. To remove these noisy areas, we adopted 3 different approaches to crop the 3D images and carried out some experiments to find out the best cropping approach.

i).Process of enhanced images

- **cropping based on Singular Points** The singular points were extracted using VeriFinger first, then a constrain was put on the positions of the core points to remove spurious ones. Based on the corrected core points, an ellipse is applied to each image and only the areas in the ellipse were kept as the region of interests. As to those images without core points (e.g. Arch type), we chose the center points as the center of the ellipse and applied the ellipse cropping as well.

- **cropping based on image quality** Similar to partial fingerprints generation [87], we used Mindtct [82], a routine of NIST Biometric Image Software, to obtain a quality map marking reliability of local fingerprint image areas at 5 different levels. Then we extracted each image foreground with high quality levels and kept only the high quality areas as region of interest.

- **cropping based on orientation reliability** The reliability of the fingerprint orientation filed describes the consistency of the local orientations in

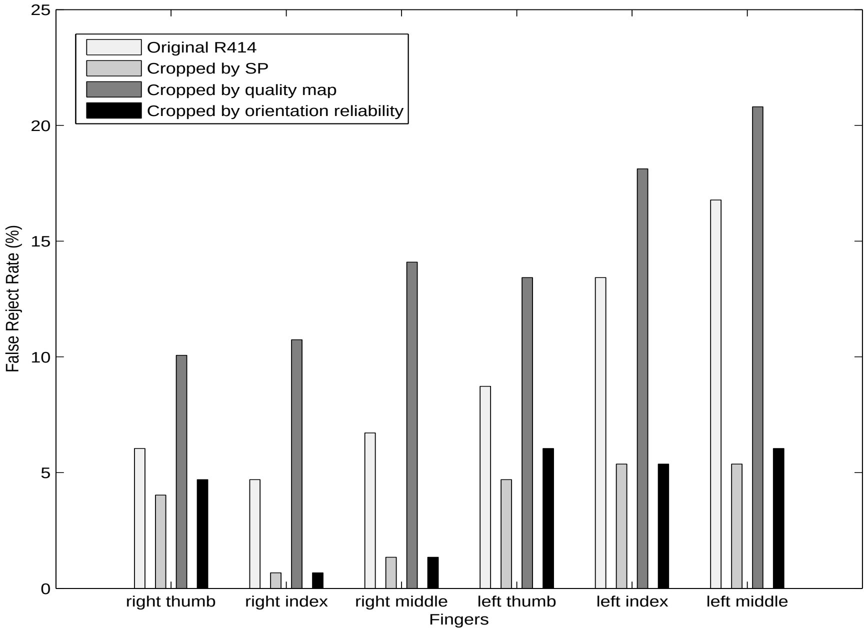


Figure 4.14: Recognition performance regarding different finger names

a neighborhood along the dominant orientation [88]. For 2D fingerprint images, the map of the orientation field reliability has peaks in the singular point locations, but for 3D fingerprint images, the map of the orientation field reliability also has peaks in noisy peripheral areas. According to this phenomenon, we cropped the 3D fingerprint images by some morphological operations [89] on the map of the orientation field reliability, and generated a foreground mask for each fingerprint image.

Figure 4.13 shows a typical example of such cropped fingerprint images under different methods and its original fingerprint.

ii). Comparison of post-processed enhanced images

Fig. 4.14 shows the performance comparison of 2D to 3D fingerprint verification using original 3D R414 images, cropped images based on SP, cropped images based on quality map and cropped images based on orientation reliability, respectively. It is obvious that quality map based cropping cannot improve the performance of 2D to 3D fingerprint verification, the possible reason is that images generated after quality map based cropping generally have singular points missing and subsequently have too many genuine minutiae missing. We can also find that both cropping based on SP and orientation reliability can improve the performance of 2D to 3D fingerprint verification greatly, especially the one based on SP. Therefore, we will use cropped 3D R414 images based on SP as the standard 3D images for our later experiments.

4.5 Summary

Non-contact 3D fingerprint technology has the tendency of replacing traditional fingerprint acquisition and recognition in many applications. This chapter first presents our recently established fingerprint database which contains both 2D fingerprints and their corresponding 3D fingerprints captured from 150 subjects. The potential use of this database is listed as follows:

- Existing databases for 3D fingerprint research contain at most 11 subjects [69], which is far from enough for verification or identification experiments. Meanwhile, there is no database containing both 2D and 3D fingerprints publicly available. Therefore, our database involving 150 subjects is the first in its kind and can be treated as a benchmark database for fingerprint biometrics. We have released the database publicly in 2015.
- The device and protocol for the 2D fingerprint subdatabase we have built are compatible with the existing databases, such as FVC database 2006,

NIST DB4, and the BioSecure database [84]. Therefore, the new 2D sub-database can be combined fully or partially with the existing databases to increase the number of available subjects.

- Effect of different acquisition devices on the quality of acquired samples and its impact on the recognition performance can be investigated using our newly established database.

- The compatibility of 2D fingerprints with their corresponding 3D fingerprints can be investigated comprehensively with our database. It can be seen from Fig. 4.7 and 4.8 that the fingerprint ridges are not continuous after post-processing, as a result, there are many falsely extracted minutiae, which will adversely affect the subsequent matching performance. So exact feature extraction of 3D fingerprints are very important to be compatible with legacy 2D fingerprints.

Based on the database we have collected, we futher conducted a series of experiments on raw 3D fingerprint images and enhanced 3D fingerprint images. The result shows that the raw 3D images, even with post-process, cannot achieve as good performance as the enhanced 3D images. So we will adopt the post-processed enhanced 3D images as the standard 3D images for our later experiments.

Chapter 5

Multi-Sensor Fingerprint Indexing Based on Minutia Cylinder-Code

5.1 Motivation and Contributions

In recent years, great improvement has been achieved in the fingerprint sensing technology. The development of sensor technology allows us to acquire fingerprints with various types of sensors; the latest is the 3D fingerprint sensor. The difference resulted from multi-type sensors significantly affects the characteristics of the raw data, the extracted features and subsequently the indexing performance. Only limited research has been carried out on the scale of impact [13][14] or non-linear distortion [15][16] in multi-sensor matching. It is still a challenging issue to exploit proper features or indexing algorithms for multi-sensor fingerprint indexing.

Among variants of features for fingerprint indexing, the Minutia Cylinder-Code (MCC) representation [90] derived from minutiae only is a robust and effective local feature descriptor. Recent studies showed that MCC can provide the best performance in terms of accuracy [33]. Its bit implementation is very flexible for representing a single minutia using hundreds of bits. Besides, it enables binary feature based fingerprint indexing [33] to be highly efficient.

Based on MCC, a Locality-Sensitive Hashing (LSH) scheme [33] has been designed to index fingerprint in large databases, which uses a numerical approximation for the similarity between MCC vectors. However, the LSH scheme is not robust enough when there is certain distortion between template and searched samples, such as fingerprints captured by multi-sensors. Therefore, we investigated the LSH and MCC characteristics of different databases, captured using different sensors. We propose an improved indexing approach based on LSH.

The main contribution of this work is threefold: (1) the comprehensive study on the LSH scheme and MCC descriptor, which points to a better solution for multi-sensor fingerprint indexing; (2) an improved indexing approach using s-liding window based LSH, which can improve the indexing performance greatly for multi-sensor fingerprint indexing, and can also improve the indexing performance for single sensor fingerprint indexing in most cases; (3) introduction of a new feature - the single maximum collision, and a fusion method that can further improve the indexing performance.

The rest of this chapter is organized as follows. Section 5.2 is a brief introduction to the MCC descriptor and the indexing algorithm based on LSH. Section 5.3 elaborates on the improved indexing scheme, including the analysis on MCC and LSH, the sliding window based LSH indexing scheme, and the final candidate list generation. Experiments on our collected multi-sensor fingerprint databases and two public benchmark fingerprint databases are demonstrated in Section 5.4 and Section 5.5 concludes the whole work.

5.2 Preliminaries On MCC and LSH

5.2.1 The MCC Descriptor

The MCC representation [90] is derived from the minutiae-only representation (i.e., location and direction) of ISO/IEC 19794-2 fingerprint minutiae templates. It encodes the neighborhood information of each minutia into a 3D data structure, called minutia cylinder, which is invariant to translation and rotation (being small at a local level), and is robust against skin distortion and small feature extraction errors. The 3D cylinder structure is divided into several sections, each corresponding to a directional difference in the range $[-\pi, \pi]$. Sections are discretized into a fixed number of $N * N$ cells. Each cell value is calculated by accumulating spatial and directional contributions from all other minutiae in the

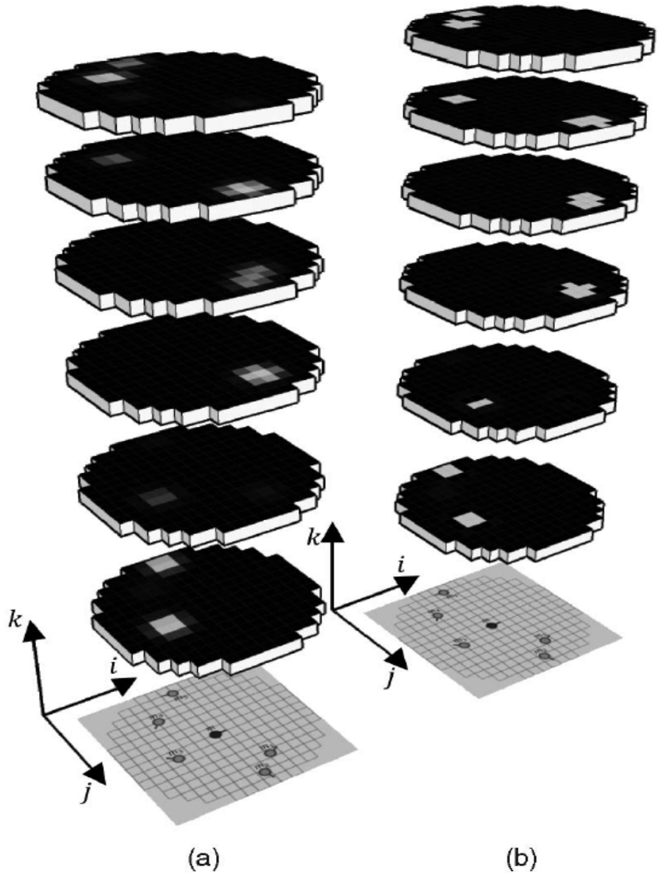


Figure 5.1: A graphical representation of a cylinder: the corresponding minutia and its neighborhood is shown below the base of the cylinder. (b) is the binary version of (a).

neighborhood for encoding, wherein the spatial contribution affects cell values in the base and the directional contribution affects the height (i.e., which section to be assigned a base value) of the 3D cylinder.

In the original MCC descriptor, each cylinder cell is associated with two bits: one denoting the cell value and the other specifying cell 'validity'. The corner cells may be labelled as 'invalid' so that they are not used in the cylinder matching phase. But in indexing approaches, for simplicity cell validity is disregarded and all of the cells are considered valid, and the cell values are quantized into binary values for bit implementation. In practice, bits from all sections are concatenated into a fixed-length binary feature vector. It is worth noting that the resulting binary representation is very sparse, with far more zeros than ones, around 95% of MCC bits are zeros on average [38]. Fig. 5.1 illustrates the minutia cylinder-code of a minutia in both original and binary versions.

5.2.2 The LSH Indexing Scheme

Based on MCC, Cappelli et al. also proposed a Locality-Sensitive Hashing (LSH) scheme for fingerprint indexing [33]. The hash-based indexing techniques are mostly built on the collision principle. The basic idea is to hash similar points to the same buckets (hash cells) such that, if two instances are similar, they would have a relatively high chance of finding colliding segments in at least some of these buckets. And a novel search algorithm has been designed using the derivation of a numerical approximation for the similarity calculation between MCC binary vectors.

Wang et al. [38] also proposed a simple yet effective geometric hashing technique based on the binary local descriptor of fingerprint minutiae cylinder-codes. The proposed technique exploited dual representations of minutiae points, both acquired from standard fingerprint templates, to build a 3D geometric hash table. And a hierarchical indexing scheme was developed which combined the merits of both LSH and geometric hashing.

5.3 Improved Indexing Approach based on LSH

5.3.1 Analysis on LSH

LSH is the most popular method solving the nearest neighbor search problem based on collision principle [91] in the Hamming space. It serves two purposes: reducing dimensionality of the input binary strings and clustering data points into buckets.

For binary feature vectors, LSH functions of random selecting bits can preserve Hamming distance. This is due to the fact that, if the number of selected bits is sufficiently large, the collision probability of two hashes is equal to the fraction of bit positions on which the two binary strings agree [92]. Thus, to achieve a good precision, LSH based methods require more sampling bits and hash tables, thus leading to a significant increase in query time and storage requirement for long inputs, typically seen in biometric representations, which often contain hundreds, if not thousands, of bits in a single instance.

For example, in [33], the LSH hash functions were chosen randomly, but a constraint was imposed to guarantee that bits were selected as evenly as possible. In particular, the number of bits in a cylinder was 312 and 24 bits were selected for each of the 32 hash functions (for a total of $32 * 24 = 768$ selected bits), while the random selection resulted in 168 (of 312) bits selected in two functions and 144 (of 312) in three ($168 * 2 + 144 * 3 = 768$). So the number of total selected bits has to be far larger than the number of bits in the cylinder. If the cylinder is composed of thousands of bits, the number of hash functions will be hundreds, which is impractical in terms of both time and storage consumptions.

In addition, 3D fingerprint images generally have more minutiae than 2D fingerprints, especially false extracted minutiae, which will impact the formation of corresponding MCC binary vectors and the bit vectors selected by each hash function. Therefore, we try to analyze the MCC features of different fingerprints from the same finger, captured by different sensors, and improve the LSH indexing algorithm to make it adaptive to indexing multi-sensor fingerprint images.

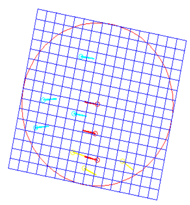
Without lose of generosity, we use the same notation as in [33] for analyzing and describing the improved indexing algorithms. Each cylinder is treated as a set of n binary cells. Let $v_m \in \{0,1\}^n$ be the binary vector obtained by linearizing the cells of cylinder C_m corresponding to a given minutia m. Hence, from a template T, a set of binary vectors V can be derived:

$$V = \{v_m \mid v_m \quad \text{obtained from} \quad C_m, C_m \in CS\} \tag{5.1}$$

where v_m is the binary vector obtained from the cylinder of minutia m and CS is the cylinder set of template T.

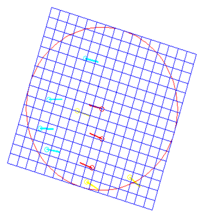
The projection of a binary vector v into a subspace with h dimensions ($h < n$) simply consists of selecting a subset of h bits from the n bits in v. More formally, let $H = \{i_1, i_2, \ldots, i_h\} \subseteq \{1, \ldots, n\}$. The projection of a given binary vector v on H is defined as $v[H] = [v_{i_1}, v_{i_2}, \ldots, v_{i_h}]$. The set of indices H defines a hash function $f_H : \{0,1\}^n \to \mathbb{N}$ that maps a binary vector to the natural number whose binary representation is $v[H]$. In the LSH approach, l hash functions are defined by randomly choosing l subsets $H_1, H_2, \ldots, H_l$, and the index consists of l hash tables $\mathbb{H}_1, \mathbb{H}_2, \ldots, \mathbb{H}_l$. Given a set of binary vectors to be indexed, each vector v is placed into bucket $f_{H_k}(v)$ of each hash table $\mathbb{H}_k$, for $k = 1, \ldots, l$. To perform a similarity search, the hash functions are applied to the query vector and all the vectors in the corresponding buckets are retrieved as candidates. The candidates are finally ranked according to their Hamming distance. In practice, LSH allows

(a) an MCC base on a 2D fingerprint image



(b) MCC base of the 2D fingerprint image

(c) an MCC base on a 3D fingerprint image



(d) MCC base of the 3D fingerprint image

Figure 5.2: An example of the MCC difference between good quality images

to drastically reduce the number of distances to be calculated by considering only those vectors that collide with the query vector under one or more of the hash functions. Moreover, if a certain degree of approximation is tolerated, the computation of distances can be completely avoided since the Hamming distance between two binary vectors can be estimated by simply counting the number of collisions of each pair of binary vectors. Intuitively, if two vectors collide under many hash functions, then their normalized Hamming similarity is likely to be high, while if the number of collisions is small, then probably the two vectors are not very similar.

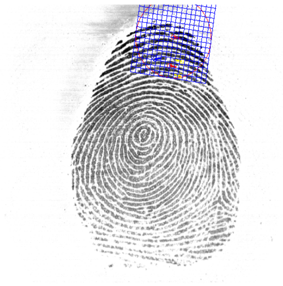
5.3.2 Analysis on MCC of Multi-sensor Fingerprints

Fig. 5.2 is a typical example showing the difference between MCC of the same minutia, but on different images. Fig. 5.2(a) is the 2D fingerprint image with the MCC base on it, and Fig. 5.2(b) is the MCC base only, whose minutiae can be identified more clearly. Fig. 2(c) is the 3D fingerprint image with the MCC base on it, and Fig. 2(d) is the MCC base without the original 3D image. Both the 2D and 3D fingerprint images are of good quality, and different colors represent different sections the minutiae should contribute to. We can see that most of the neighboring minutiae have the same spatial and directional relations to the central minutia, but there are still some differences in the neighborhood due to the spurious or missing minutiae. The difference will have a direct impact on the bit one distribution in the MCC binary vector of the minutia.

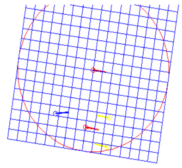
Fig. 5.3 is another example demonstrating the similarity between the MCC of the same minutia, but in different images. Fig. 5.3(a) is the 2D fingerprint image with the MCC base on it, and Fig. 5.3(b) is the MCC base only. Fig. 5.3(c) is the 3D fingerprint image with the MCC base on it, and Fig. 5.3(d) is its MCC base only. We can see that these two MCC bases have the same neighboring relations, which means that their MCC binary vectors will be the same.

From these two typical examples, we can find some characteristics of the MCC binary vectors between multi-sensor fingerprint images: 1) overall, most of the MCC binary vectors of the same minutia but from multi-sensor fingerprint images differ in some bits, which will have an influence on the natural numbers of the hash selected sub binary vectors; and 2) in some local areas, the MCC binary vectors of the same minutia but from multi-sensor fingerprint images are completely the same.

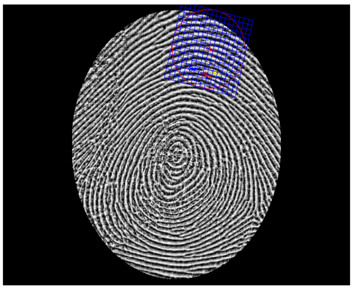
On one hand, the LSH based approach can tolerate some degree of distortion,



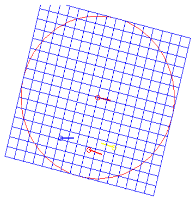
(a) an MCC base on a 2D fingerprint image



(b) MCC base of the 2D fingerprint image



(c) an MCC base on a 3D fingerprint image



(d) MCC base of the 3D fingerprint image

Figure 5.3: An example of the MCC similarity

because it chooses many hash functions (e.g. 32) instead of just a few, and the number of selected bits for each hash function is often large enough (e.g. 24). However, if two binary vectors differ in 1 bit under a certain hash function, the collision will fail, which will impact the final similarity score. To solve this problem, we propose to divide each sub binary vector, selected by LSH hash functions, into finer vectors, and a good solution is to use a sliding window based strategy for division. On the other hand, suppose two fingerprints of the same finger but captured by different sensors generally have a pair of MCC binary vectors more similar than those of all the other false indices, this single maximum collision can be adopted to supervise the indexing process, and it can be easily calculated during the LSH based collision computation.

5.3.3 The Indexing Approach

In the sliding window based LSH approach, we divide each binary set H into several finer subsets. Suppose the window size is $w_s, w_s < h$, the first subset is generated from the first binary bit to the w_sth binary bit, then the window slides right one binary bit every time, generating in total $h - w_s + 1$ subsets. Therefore, each hash function H_k is replaced by $h - w_s + 1$ sub functions, which are denoted by $H_{k-q}, 1 \leq k \leq l, 1 \leq q \leq h - w_s + 1$. Accordingly, given a set of binary vectors to be indexed, each vector v is placed into bucket $f_{H_{k-q}}(v)$ of each hash table $\mathbb{H}_{k-q}$. To perform a similarity search, the hash functions are applied to the query vector and all the vectors in the corresponding buckets are retrieved as candidates.

Fig. 5.4 is a simple example of the sliding window based LSH indexing. Suppose the first hash function H_1 chooses 18 bits ($h = 18$) from two MCC binary vectors, v_1 and v_2. According to the original LSH indexing approach, $f_{H_1}(v_1) = 33284$ and $f_{H_1}(v_2) = 516$, therefore, if v_1 is the template binary vector, and v_2 is the query binary vector, v_2 will not collide with v_1 under hash function H_1, although $v_1[H_1]$ and $v_2[H_1]$ differ only at the third bit. In the sliding window based approach ($w_s = 12$), H_1 is divided into 7 sub functions H_{1-q} ($1 \leq q \leq 7$) by sliding. Although the first three hash values ($f_{H_{1-1}}(v_1)$ vs. $f_{H_{1-1}}(v_2)$, $f_{H_{1-2}}(v_1)$ vs. $f_{H_{1-2}}(v_2)$, and $f_{H_{1-3}}(v_1)$ vs. $f_{H_{1-3}}(v_2)$) are different, the remaining four hash values are the same. That is, if v_1 is the template binary vector and v_2 is the query binary vector, v_2 collides with v_1 under four sub functions of H_1.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18		f_{H1}
v_1 [H_1]	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0		33284
v_2 [H_1]	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0		516

	1	2	3	4	5	6	7	8	9	10	11	12			
v_1 [$H_{1\text{-}1}$]	0	0	1	0	0	0	0	0	1	0	0	0		$f_{H1\text{-}1}(v_1)$	520
v_1 [$H_{1\text{-}2}$]	0	1	0	0	0	0	0	1	0	0	0	0		$f_{H1\text{-}2}(v_1)$	1040
v_1 [$H_{1\text{-}3}$]	1	0	0	0	0	0	1	0	0	0	0	0		$f_{H1\text{-}3}(v_1)$	2080
v_1 [$H_{1\text{-}4}$]	0	0	0	0	0	1	0	0	0	0	0	0		$f_{H1\text{-}4}(v_1)$	**64**
v_1 [$H_{1\text{-}5}$]	0	0	0	0	1	0	0	0	0	0	0	1		$f_{H1\text{-}5}(v_1)$	**129**
v_1 [$H_{1\text{-}6}$]	0	0	0	1	0	0	0	0	0	0	1	0		$f_{H1\text{-}6}(v_1)$	**258**
v_1 [$H_{1\text{-}7}$]	0	0	1	0	0	0	0	0	0	1	0	0		$f_{H1\text{-}7}(v_1)$	**516**

	1	2	3	4	5	6	7	8	9	10	11	12			
v_2 [$H_{1\text{-}1}$]	0	0	0	0	0	0	0	0	1	0	0	0		$f_{H1\text{-}1}(v_2)$	8
v_2 [$H_{1\text{-}2}$]	0	0	0	0	0	0	0	1	0	0	0	0		$f_{H1\text{-}2}(v_2)$	16
v_2 [$H_{1\text{-}3}$]	0	0	0	0	0	0	1	0	0	0	0	0		$f_{H1\text{-}3}(v_2)$	32
v_2 [$H_{1\text{-}4}$]	0	0	0	0	0	1	0	0	0	0	0	0		$f_{H1\text{-}4}(v_2)$	**64**
v_2 [$H_{1\text{-}5}$]	0	0	0	0	1	0	0	0	0	0	0	1		$f_{H1\text{-}5}(v_2)$	**129**
v_2 [$H_{1\text{-}6}$]	0	0	0	1	0	0	0	0	0	0	1	0		$f_{H1\text{-}6}(v_2)$	**258**
v_2 [$H_{1\text{-}7}$]	0	0	1	0	0	0	0	0	0	1	0	0		$f_{H1\text{-}7}(v_2)$	**516**

Figure 5.4: An example of sliding window based LSH

i).Creating the index

Algo. 5.1 shows index creation. Any binary vector v_j of each template T_i is given as the input to all of the LSH hash functions first. Since binary vectors obtained with the MCC representation tend to be quite sparse (having more 0s than 1s), those buckets whose binary representations have only a few 1 bits are more likely to contain a large number of pairs and hence are less selective. For this reason, a parameter (min_{PC}) is used to discard selections with a low number of 1 bits in the LSH mapping, and another parameter (min_{PC_s}) is used to discard buckets with a low number of 1 bits in the sub hash tables. The pair (i, j), which identifies template T_i but also vector v_j (corresponding to minutia m_j of T_i), is stored in the corresponding buckets of the sub hash tables.

Algorithm 1: Creation of the index

Input:
Database of minutiae templates $DB = \{T_1, T_2, \ldots\}$;
Set of hash functions $\mathbb{F} = \{f_{H_k}, k = 1, \ldots, l\}$;
Sliding window size w_s.
Output:
Sub hash tables $\mathbb{H} = \{\mathbb{H}_{k-q}, k = 1, \ldots, l, q = 1, \ldots, h - w_s + 1\}$.

```
Generate sub hash functions F_s = {f_{H_{k-q}}, k = 1,...,l, q = 1,...,h − w_s + 1}.
foreach template T_i in DB do
    Use MCC to create binary vector set V_i from T_i
    foreach vector v_j in V_i do
        foreach hash function f_{H_k} in F do
            b = f_{H_k}(v_j)
            if PopCount(b) ≥ min_PC then
                foreach sub hash function f_{H_{k-q}} under f_{H_k} do
                    b_s = f_{H_{k-q}}(v_j)
                    if PopCount(b_s) ≥ min_{PC_s} then
                        Store (i, j) in bucket b_s of table H_{k-q}
                    end
                end
            end
        end
    end
end
```

ii).Candidate list generation

At retrieval time, the same hash functions and sub hash functions such as those being adopted in index creation are applied to each binary vector of the searched template, and the number of collisions with database vectors is counted using an accumulator matrix $\mathbb{S}$ (see Algorithm 5.2). In this way, the most similar database templates are efficiently determined by accumulating the vector similarity of all the minutiae and the candidate list is easily produced.

Apart from discarding buckets corresponding to a small number of 1 bits in both LSH mapping process and sub hash tables, to reduce the number of elements considered by the accumulator, we only consider the minutiae which satisfy basic

Algorithm 2: Candidate list generation

Input:
Minutiae template T of the searched fingerprint
Database of minutiae templates $DB = \{T_1, T_2, ...\}$;
Set of hash functions $\mathbb{F} = \{f_{H_k}, k = 1, ..., l\}$;
Set of sub hash functions $\mathbb{F}_s = \{f_{H_{k-q}}, k = 1, ..., l, q = 1, ..., h - w_s + 1\}$.
Set of sub hash tables $\mathbb{H} = \{\mathbb{H}_{k-q}, k = 1, ..., l, q = 1, ..., h - w_s + 1\}$.
Maximum number of candidates max_C
Output:
A set of candidates $CL = \{(i_k, s_k)\}$ sorted by sliding window base LSH, where each candidate (i_k, s_k) consists of a database template index i and its score s.
A set of candidates $CL_{smc} = \{(i_p, s_p)\}$ sorted by single maximum collison, where each candidate (i_p, s_p) consists of a database template index i and its score s.

```
Use MCC to create binary vector set V from T
Initialize the hash score accumulator S
Initialize the single maximum collision score accumulator S_smc
foreach vector v in V do
    Let m be the minutia of T corresponding to v
    Reset collision accumulator A
    foreach hash function f_{H_k} in F do
        b = f_{H_k}(v)
        if PopCount(b) ≥ min_PC then
            foreach sub hash function f_{H_{k-q}} in F_s do
                b_s = f_{H_{k-q}}(v)
                if PopCount(b_s) ≥ min_{PC_s} then
                    foreach pair (i, j) in bucket b_s of table H_{k-q} do
                        Let m_j be the j^th minutia of template T_i
                        if Compatible(m, m_j) = true then
                            A[i, j] = A[i, j] + 1
                        end
                    end
                end
            end
        end
    end
    foreach T_i with at least one collision in A do
        Let CF_max = max_j {A[i, j]}
        S[i] = S[i] + (CF_max)^(p/h)
        if S_smc[i] < CF_max then
            S_smc[i] = CF_max
        end
    end
end
foreach template index i in S do
    S[i] = S[i] / (|V|·(l·(h−w_s+1)))^(p/h)
end
Create CL by selecting at most max_c pairs (i, S[i]) with the highest scores in S
Create CL_smc by selecting at most max_c pairs (i, S_smc[i]) with the highest scores in S_smc
```

geometric constraints. Function $Compatible(m, m_j)$ is defined as follows:

$$Compatible(m, m_j) = \begin{cases} true, & if \quad d_\theta(m, m_j) \leq \delta_\theta \quad and \quad d_{xy}(m, m_j) \leq \delta_{xy}; \\ false, & otherwise. \end{cases} \tag{5.2}$$

where d_θ and d_{xy} are the angular difference and the Euclidean distance between the two minutiae, respectively.

In addition, when counting the number of collisions with database vectors, another accumulator vector $\mathbb{S}_{smc}$ is used to record the number of single maximum collision with database template, and another candidate list is generated by sorting the accumulator vector $\mathbb{S}_{smc}$.

iii).Candidate list fusion

The candidate list produced by sliding window based LSH indexing is expected to be more reliable because the vector $\mathbb{S}$ accumulates the similarity of every minutia in the searched template to database templates. That is, it is the averaged similarity at a global level. And the candidate list generated according to vector $\mathbb{S}_{smc}$ provides some information about local similarity and can be used to produce the final candidate list. Similar to our work in partial fingerprint indexing [87], we treat all the template fingerprints as candidates and apply a fuzzy-based fusion scheme to the two candidate lists to produce a new candidate list.

The fuzzy-based candidate list fusion scheme is reduced to only one stage (without the training stage) - the testing stage, because we believe the penetration rate of sliding window based LSH indexing is lower than the single maximum collision indexing, which means that the candidate list generated by the sliding window based LSH indexing is more reliable, and its top K candidates can keep the original order in the new candidate list.

The fuzzy sets and fuzzy rules are the same as those for partial fingerprint indexing [87]. After applying the fuzzy rules to both candidate lists, the new candidate list is generated as the final candidate list.

5.4 Performance Evaluation and Analysis

To evaluate the proposed multi-sensor fingerprint indexing approach, comparison experiments have been carried out on both our collected 2D and 3D fingerprint databases and two public fingerprint databases. Section 5.4.1 describes the databases, tools and the performance measures used in our experiments and Section 5.4.2 demonstrates the experimental results on our collected 2D and 3D fingerprint databases. Section 5.4.3 demonstrates the experimental results on two public databases and Section 5.4.4 is the analysis on the computational complexity. The comparison experiment between even bit selection indexing and hash based indexing is in Section 5.4.5.

5.4.1 Databases, Tools and Performance Measures

i).Database

- 2D database: the 2D database we collected [93] contains 6000 fingerprints from 1500 fingers (4 impressions per finger). Only the first and the second impressions were used in our experiments.

- 3D database: the 3D database we collected [93] contains nearly 3000 fingerprints from 1500 fingers (2 impressions per finger). Only cropped fingerprints from thumb, index, and middle fingers were used in our experiments.

- FVC2000 DB2: the second FVC2000 database contains 800 fingerprints from 100 fingers (8 impressions per finger) captured using the capacitive scanner Touch-Chip by ST Microelectronics.

- FVC2002 DB1: the first FVC2002 database, containing 800 fingerprints from 100 fingers (8 impressions per finger) captured using the optical scanner TouchView II by Identix.

ii).Tools

In all the experiments, we used the third-party commercial software VeriFinger 4.0 from NeuroTechnology [79] to extract minutiae features from the fingerprints. No particular pre-processing steps, such as image enhancement, foreground segmentation or fingerprint alignment, were carried out before feature extraction. After converting the radians into standard ISO/IEC 19794-2 format, the extracted minutiae templates are input to the MCC SDK v2.0 software [94] to create the MCC binary representations: Cylinders with an 16-cells in diameter and six sections were created (resulting in a total of 1536 cells, 256 per section). The same parameters reported in [33] have been used to create the cylinders, with the only exception of μ_Ψ, which has been set to 1/200 here instead of 1/100. This new value is more appropriate for indexing since it allows a sufficient number of one-valued cells to be obtained without reducing the discriminability of the features. In addition, we followed the practice in [33][38] that disregards the cell validity bits by considering all cells to be valid.

iii).Performance Measures

The performance of fingerprint indexing approaches is typically evaluated by reporting the trade-off between error rate and penetration rate: this trade-off usually depends on a parameter such as the maximum number of candidates to be considered. The error rate is defined as the percentage of searched fingerprints that are not found; the penetration rate is the portion of database that the system has to search on average.

Apart from the above performance indicators, some researchers [33] also consider another retrieval scenario (incremental search), where an ideal matching algorithm is used to stop the search as soon as the right candidate is retrieved. In

Table 5.1: Major Parameters Used

Symbol	Explanation	Values
n	number of cells (bits) in a cylinder	1536
h	number of bits selected by each hash function	24
l	number of hash functions	32
w_s	size of the sliding window to generate sub hash bits	14
p	shape parameter of distance function	30
min_{PC}	minimum one-bits for LSH index to be used	2
min_{PC_s}	minimum one-bits for sub bucket index to be used	1
δ_θ	maximum global fingerprint rotation (radians)	$\pi/4$
δ_{xy}	maximum global fingerprint translation (pixels)	256
K	number of fixed candidates	40, 5

such a scenario there are no retrieval errors, since in the worst case the search can be extended to the whole database, and the average penetration rate is reported as a single performance indicator.

5.4.2 Comparison Experiments on Our Collected Databases

i).Performance of 2D to 3D fingerprint indexing

For 2D to 3D fingerprint indexing, part of the first impressions (from 6 fingers per person: thumbs, index fingers and middle fingers) in 3D fingerprint database were treated as database templates, and the first impression of corresponding 2D fingerprints were used as the searched templates. The parameters we adopted and their detailed explanations were listed in Table 5.1, and the following experiments would use the same parameters for indexing.

To be compatible with the original LSH indexing, we choose to report the trade-off between error rate and penetration rate as the performance measure of 2D to 3D fingerprint indexing. The original LSH indexing was performed using the Matlab function about indexing in MCC SDK 2.0, and the sliding window

Table 5.2: Performance in Penetration Rate

Database	Indexing experiments	K	Original LSH	Sliding LSH	Fusion
Our databases	2D vs. 3D	40	**2.41**	**1.24**	**1.18**
	2D vs. 2D		0.32	0.34	0.32
	3D vs. 3D		**0.49**	**0.33**	0.33
Public databases	FVC 2000 db2	5	**1.96**	**1.71**	**1.63**
	FVC 2002 db1		1.76	2.16	2.08

base LSH indexing and fusion algorithms were also realized in Matlab using the same parameters.

The performance comparison of 2D to 3D fingerprint indexing is shown in Fig. 5.5. It can be seen that no matter what penetration rate is, the error rate is much smaller in sliding window based indexing than that in original LSH indexing. On average, the error rate dropped by half. In the fusion scheme, we used the candidate list produced by sliding window based indexing as the reliable ranking and fixed top 40 of its original order ($K = 40$) as the same order in the final candidate list. In Fig. 5.5, we could not identify clearly the advantage of the fusion scheme, so we adopted another indicator, the average penetration rate in incremental search, as the performance measure. The performance after fusion is shown in Table 5.2, which indicates the performance after fusion (1.18%) is better than that in sliding window base LSH indexing (1.24%).

ii).Performance of 2D to 2D fingerprint indexing

For 2D to 2D fingerprint indexing, all of the first impressions in the 2D fingerprint database (1500 fingerprints) were treated as database templates, and all the second impressions in the 2D fingerprint database were used as the searched templates. The trade-off between error rate and penetration rate of 2D to 2D fingerprint indexing is shown in Fig. 5.6. We can see that when the penetration

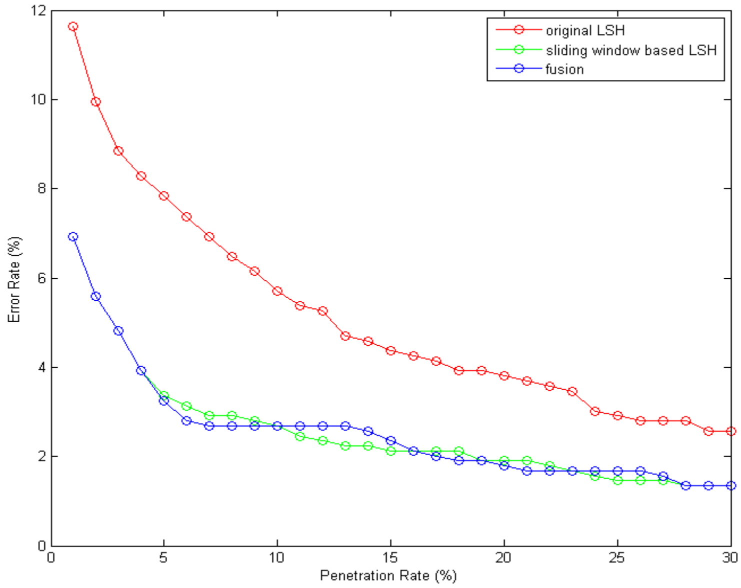


Figure 5.5: Indexing performance of 2D to 3D database

rate is less than 24%, the error rate is generally much smaller in sliding window based indexing than that in original LSH indexing, especially when the penetration rate is between 7% and 22%. Compared to 2D to 3D fingerprint indexing, the error rate is much smaller (0.75% vs. 3.5% at penetration rate 5%). In the fusion scheme, we also used the candidate list produced by sliding window based indexing as the reliable ranking and fixed top 40 of its original order ($K = 40$) as the same order in the final candidate list. Fig. 5.6 shows that the performance after fusion is better than the sliding window based LSH indexing most of time. On average, their performance is similar, which can be shown in Table 5.2 using the average penetration rate measurement.

iii).Performance of 3D to 3D fingerprint indexing

For 3D to 3D fingerprint indexing, similar to 2D to 3D fingerprint indexing, part of the first impressions in the 3D fingerprint database were treated as database tem-

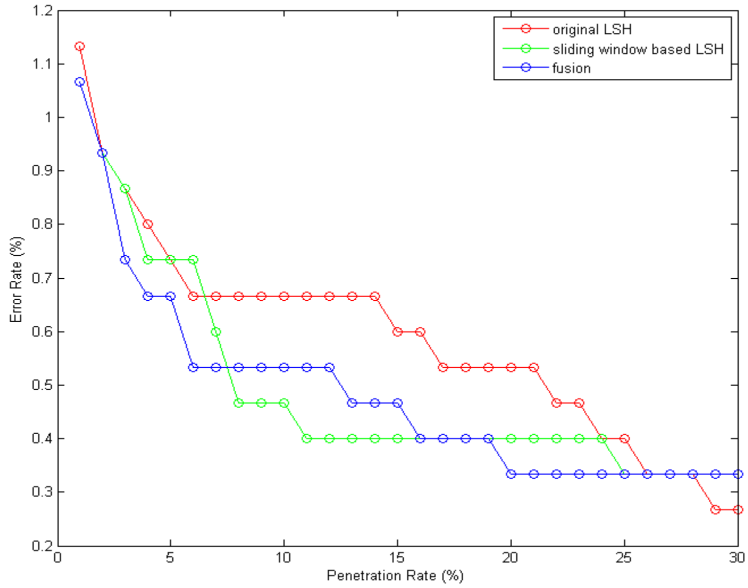


Figure 5.6: Indexing performance on the 2D database

plates, and their corresponding second impressions in the 3D fingerprint database were used as the searched templates. The trade-off between error rate and penetration rate of 3D to 3D fingerprint indexing is shown in Fig. 5.7. We can see that the error rate is on average much smaller in sliding window based indexing than the one in original LSH indexing. In the fusion scheme, we also used the candidate list produced by sliding window based indexing as the reliable ranking and fixed top 40 of its original order ($K = 40$) as the same order in the final candidate list. Fig. 5.7 shows that the performance after fusion is not as good as that in sliding window based LSH indexing when the penetration rate is between 4% and 29%, but the average penetration rate is the same as that in sliding window based LSH indexing, which can also be seen in Table 5.2.

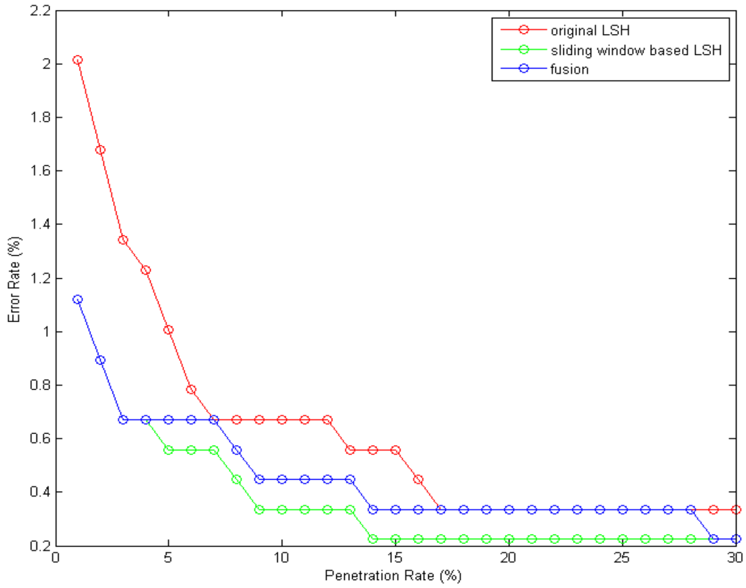


Figure 5.7: Indexing performance on the 3D database

5.4.3 Comparison Experiments on Public Databases

We also compared the indexing performance on two public benchmark fingerprint databases, FVC 2000 db2 and FVC 2002 db1. In these experiments, all of the results have been obtained by using the first impression for index creation and the remaining seven for searching.

i).Indexing performance on FVC 2000 db2

The performance of fingerprint indexing on FVC 2000 db2 is shown in Fig. 5.8 by reporting the trade-off between the error rate and penetration rate. We can see that the error rate is much smaller in sliding window based indexing than the one in original LSH indexing, and the error rate drops to 0 when the penetration rate is 9% in sliding window based indexing, while the penetration rate is 11% in original LSH indexing to achieve the same error rate. In the fusion scheme, we also

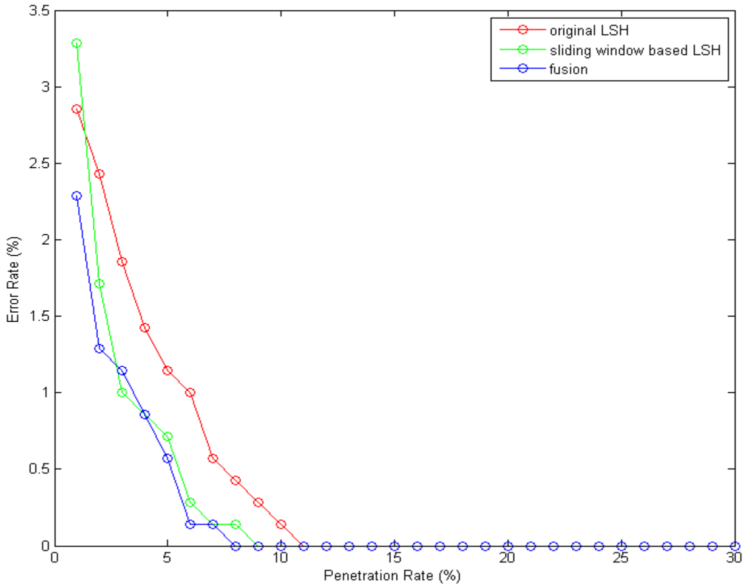


Figure 5.8: Indexing performance on FVC 2000 DB2

used the candidate list produced by sliding window based indexing as the reliable ranking. Since the number of templates in FVC is 100, which is much smaller than the number of templates in the 2D or 3D fingerprint database, we only fixed top 5 of its original order ($K = 5$) as the same order in the final candidate list. Fig. 5.8 shows that the performance after fusion is generally better than that in sliding window base LSH indexing, which can also be demonstrated in Table 5.2 using the average penetration rate measurement.

ii).Indexing performance on FVC 2002 db1

The performance of fingerprint indexing on FVC 2002 db1 is illustrated in Fig. 5.9. In the fusion scheme, we also used the candidate list produced by sliding window based indexing as the reliable ranking and fixed top 5 of its original order ($K = 5$) as the same order in the final candidate list. We can see that although the error rate drops to 0 when the penetration rate is 15% in all the

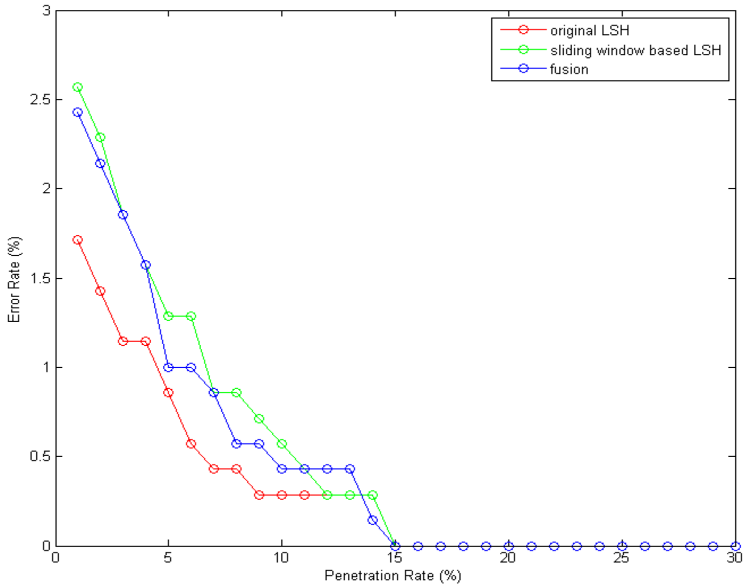


Figure 5.9: Indexing performance on FVC 2002 DB1

cases, the error rates in sliding window based indexing and the fusion scheme are generally larger than the one in original LSH indexing. Therefore, the sliding window based scheme and fusion scheme cannot improve the indexing performance on this database, which can also be demonstrated in Table 5.2 using the average penetration rate measurement. The possible reason is that among the 8 impressions per finger in this database, there is one impression with only several minutiae, which can result in more false collisions under sliding window based LSH indexing.

5.4.4 Computational Complexity Analysis

Suppose the fingerprint database contains N minutiae templates and each template contains F minutiae points. In our approach, the creation of index consists of two preprocesses. The first one is to obtain the binary MCC codes, which takes $O(F)$ time [38]. This part of preprocessing is very fast since each MCC bit-based representation can be computed within several milliseconds using the MCC SDK [94] on a 2.66 GHz Intel Quad Core CPU [33]. The second one is to generate the hash tables. Since every minutia is used as an input, and there are l hash functions and each hash function is further divided into $h-w_s+1$ sub hash functions, the cost of the second preprocess for each template is $O(Fl*(h-w_s+1))$ [33]. Therefore, the total time for creating the index is $O(NF)+O(NFl*(h-w_s+1))$ for the whole database.

At retrieval time, the key value of a query point is hashed, and then the corresponding bucket is searched for the matching item. The time for hashing and bucket access is constant, but the time to search a bucket for the matching item is linear with the number of items in the bucket [38]. In sliding window based LSH, the number of buckets is controlled by the number of bits selected by each sub hash function, and the ideal number is $l*(h-w_s+1)*2^{w_s}$. Assume the most occupied bucket has M entries (the worst case being $M = NF$), a query point needs to search M items to find the best match with maximum collisions of the hash indices. For a fingerprint with F minutiae points, the search time complexity is thus $O(MF)$ [38]. The accumulated maximum collision counts and the single maximum collision counts are then accumulated for all N fingerprints in the database. The final results of N scores from accumulated maximum collision counts are sorted to produce a list of match candidates, which can be done in $O(N \log N)$ time [38]. The same time is needed to produce the second candidate list by sorting the scores from the single maximum collision counts. Finally, the fusion scheme also costs $O(N)$ time.

5.4.5 Comparison of Hash Bit Selection and Even Selection

The LSH based indexing scheme can approximate the similarity in the Hamming space, another similarity measure is the nearest neighbor search in Euclidean space. To obtain the exact Euclidean distance between two MCC vectors, a simple way is to use even bit selection instead of bit selection by hash functions. In the previous experiments, the length of each MCC binary vector (n) was 1536, and all the LSH hash functions selected 768 ($32*24$) bits (some bits were selected repeatedly) in total, so not every bit was considered during the selection and contributed to the final indexing. In this experiment, we tried to select every bit once sequentially, namely even selection, and compare the indexing performance of it with the LSH based scheme.

i).Performance of hash bit selection vs. even selection

For even selection, we set the sub vector length ($h = 24$) to be the same as that in LSH indexing, so there are 64 even functions (1536 divided by 24). The first even function chooses bit 1 to bit 24, the second one chooses bit 25 to bit 48, and the following even functions will choose their bits in sequence. The LSH hash functions are the same as those in previous experiments, that is, randomly selecting 24 bits as a hash function, and 32 hash functions in total. The experiments were carried out regarding 2D to 3D fingerprint indexing.

Fig. 5.10 is the indexing performance comparison between even selection and LSH hash selection, indicated by the tradeoff between error rate and penetration rate. We can see that sliding window based LSH indexing significantly outperforms both original even selection and siding window based even selec-

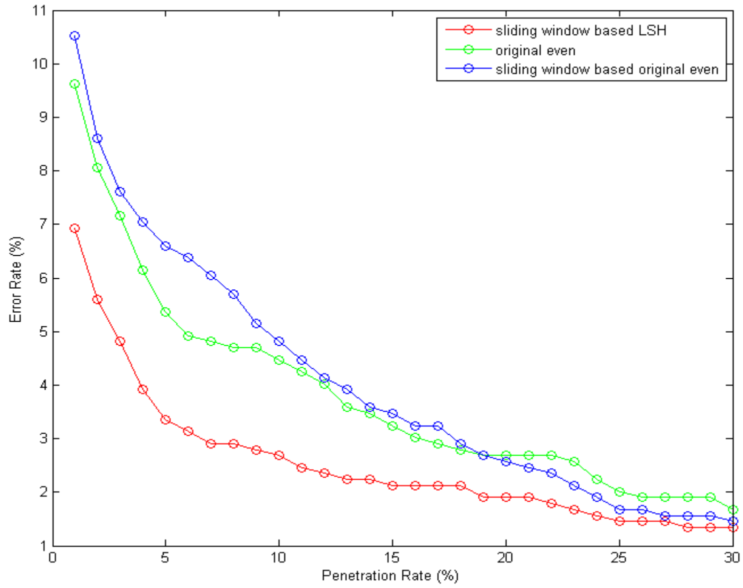


Figure 5.10: Even bit selection vs. LSH hash selection on 2D and 3D database

tion indexing, although the LSH hash functions select fewer bits than the even selection.

ii).Analysis of bit selection

From the above experiment, we can conclude that even selection is not as good as random hash selection. We plotted the accumulated bit frequency on 2D and 3D database and the bit occurrence rate in random hash selection in Fig. 5.11 and Fig. 5.12, respectively. From Fig. 5.11, it is clear that the occurrence frequency of bit 1 in both 2D and 3D fingerprint databases has the same distribution, that is, some positions (of 1536 bits) have a high rate of meeting bit 1, whereas some positions have no bit 1 in all the MCC binary vectors, and the accumulated bit frequency is symmetric to the central point (around 768). Because 3D fingerprints have more minutiae than 2D fingerprints on average, the absolute value of 3D accumulated bit frequency is larger than that of 2D accumulated bit frequency.

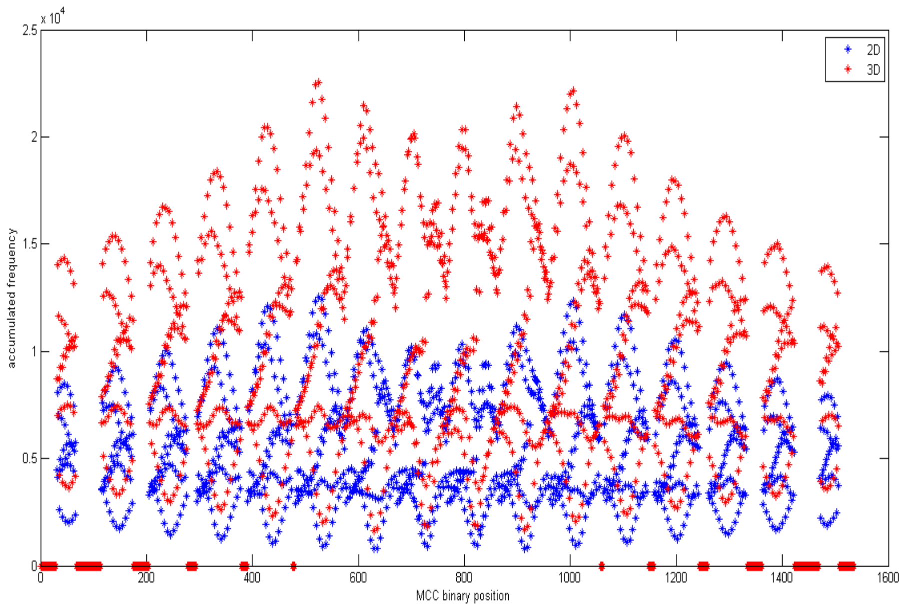


Figure 5.11: Accumulated bit frequency on 2D and 3D database

Since the bit 1 occurrence frequency is quite different regarding a single database, the bits with a high occurrence rate are less selective in even selection. Fig. 5.12 can better demonstrate this phenomenon in LSH hash selection, not all the bits have been selected (600 out of 1536) and not all the bits selected occur just once. It can be observed from Fig. 5.12 that most of the selected bits occur once, some twice, some three times and one bit even occurs four times.

5.5 Summary

With the rapid expansion of fingerprint databases, fingerprint indexing before matching becomes more and more important, and the latest advance in sensor technology requires us to develop new algorithms for multi-sensor fingerprint indexing. In this chapter, we have given a comprehensive study on the state-of-the-art MCC descriptor and LSH based indexing algorithm and pro-

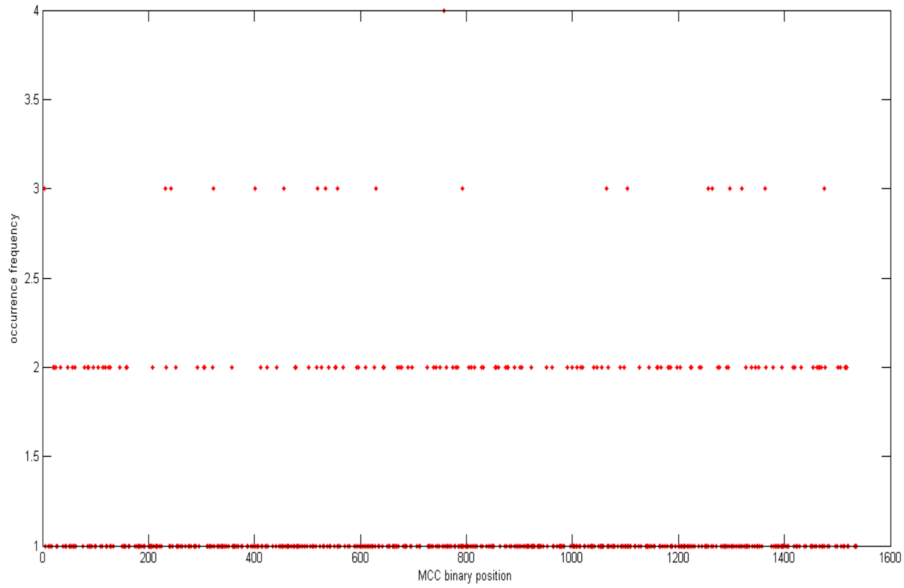


Figure 5.12: Bit occurrence in LSH

posed an improved indexing approach based on them. The proposed approach divides the hash bit vectors, selected by LSH using a sliding window, into finer sub-vectors for indexing. We also take into consideration another feature - the single maximum collision for generating another candidate list, and fuse the candidate lists produced by both sliding window based LSH indexing and single maximum collision indexing to produce the final candidate list. Experimentations carried out on our collected multi-sensor fingerprint database showed that the proposed indexing approach could improve the performance of fingerprint indexing greatly. Evaluation was also conducted on some public benchmark databases for fingerprint indexing, and the results demonstrated that the proposed approach could outperform existing ones in almost all the cases.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

Identifying a partial fingerprint against a large database has always been extremely challenging, because it is hard to narrow down the candidate list for partial fingerprint verification with limited features. Besides, the traditional contact-based capture of fingerprints is gradually being complemented by contact-less 3D fingerprint technology. The compatibility and recognition between different fingerprints captured by different sensors is another challenge in fingerprint biometric. Investigating these two challenges, this thesis made the following contributions:

- We have proposed to use two levels of features for partial fingerprint indexing, the reconstructed global feature based on common fingerprint pattern and local features derived from existing minutiae. On the local level, we design several simple and effective novel features of minutiae triplets in addition to some commonly used features to constitute the local minutiae triplet features. We then propose to combine the reconstructed global feature and local minutiae triplet features to improve the performance of partial fingerprint indexing. Specifically, two candidate lists are generated first by using the minutiae triplet based indexing scheme and the FOMFE coefficients based indexing scheme, respectively, then a fuzzy-based fusion scheme based on ranking is designed to generate the final candidate list for matching. To investigate the performance of the new features added to minutiae triplet feature set, we have carried out several experiments on FVC 2000 DB2a, FVC 2002 DB1a and NIST SD 14, and the results demonstrate that the features can improve the indexing performance greatly. Moreover, experiments carried out on the public database NIST SD 14 show that the comprehensive partial fingerprint indexing approach can improve the performance that has been achieved by individual partial fingerprint indexing algorithms before fusion.

- To better investigate the 3D fingerprint biometric, we have collected a multi-sensor fingerprint database consisting of both 3D fingerprints as well as their corresponding 2D fingerprints captured by two commercial fingerprint scanners from 150 subjects in Australia. We have also carried out a series of experiments on 2D fingerprint verification, 3D fingerprint verification, and 2D to 3D fingerprint verification, using different 3D images: 3D raw images, post-processed 3D raw images, 3D enhanced images, and post-processed 3D enhanced images. The results show that the post-processed 3D enhanced images can achieve the best performance regarding 2D to 3D fingerprint recognition. In addition, the database was released publicly in 2015. The large size of the database will provide meaningful statistical analysis and a truthful assessment of the performance of the state-of-the-art algorithms in this area.

- We have made a comprehensive study on the most effective indexing algorithm - LSH indexing built on MCC descriptor, and proposed a finer hash bit selection method based on LSH for multi-sensor fingerprint indexing. Specifically, we divide the hash bit vectors, selected by LSH using a sliding window, into finer sub-vectors with certain fixed length, and then convert these sub-vectors into numerical approximation for MCC indexing, and finally generate the first candidate list. Moreover, we take into consideration another feature - the single maximum collision for indexing. This feature can generate another candidate list. Finally, the two candidate lists are fused using the fusion scheme for partial fingerprint indexing to generate the final candidate list. Experimentations carried out on our collected 2D and 3D databases show that the proposed indexing approach greatly improves the performance of fingerprint indexing. Evaluation was also conducted on some public benchmark databases for fingerprint indexing, and the results demonstrated that the new approach outperforms existing ones in almost all the cases.

6.2 Future Work

3D fingerprint technology is developing fast and complementing the traditional contact-based fingerprint technology. However, this technology is not mature and there are still a number of interesting avenues for future research. Some typical problems are summarized as follows.

- From Chapter 4, we have noticed that the 3D fingerprint sensor we adopted cannot produce a real 3D fingerprint model in 3D space, which limited the research on fingerprint recognition in 3D space. Therefore, the 3D fingerprint model capture or reconstruction, feature extraction in 3D space, and direct 3D fingerprint indexing or recognition need to be solved in the near future. What is more, partial 3D fingerprint indexing will be very challenging, as it is more complex in both the feature construction and algorithm design.

- The unraveled 2D equivalent fingerprint images produced by 3D fingerprint sensor generally contain much noise, which influences the performance regarding both 3D to 3D recognition and 2D to 3D recognition. Although post-processed enhanced 3D images can achieve good performance, compared to 2D fingerprint images, more new techniques are needed to enhance the raw 3D images.

- Both the LSH indexing and the sliding-window based LSH indexing rely on hash bit selection. According to the experiments in Chapter 5, the discrimination of each bit in the MCC binary code is different, so it is possible to develop a better bit selection strategy than hash bit selection based on the discrimination information.

References

[1] T. Ahmad, J. Hu, and S. Wang, "Pair-polar coordinate-based cancelable fingerprint templates," *Pattern Recogn.*, vol. 44, pp. 2555–2564, Oct. 2011.

[2] S. Wang and J. Hu, "Alignment-free cancelable fingerprint template design: A densely infinite-to-one mapping (ditom) approach," *Pattern Recogn.*, vol. 45, pp. 4129–4137, Dec. 2012.

[3] K. Xi, T. Ahmad, F. Han, and J. Hu, "A fingerprint based bio-cryptographic security protocol designed for client/server authentication in mobile computing environment.," *Journal of Security and Communication Networks*, vol. 4, no. 5, pp. 487–499, 2011.

[4] K. Xi and J. Hu, *Introduction to Bio-cryptography.* Handbook of Information and Communication Security, Springer Verlag, 2010.

[5] K. Xi, J. Hu, and F. Han, "Mobile device access control: an improved correlation based face authentication scheme and its java me application," *Concurr. Comput. : Pract. Exper.*, vol. 24, pp. 1066–1085, July 2012.

[6] K. Xi, Y. Tang, and J. Hu, "Correlation keystroke verification scheme for user access control in cloud computing environment," *Comput. J.*, vol. 54, pp. 1632–1644, Oct. 2011.

[7] "Tbs biometrics. http://www.tbs-biometrics.com/."

[8] "Flashscan3d. http://www.flashscan3d.com/."

[9] Y. Wang and J. Hu, "Global ridge orientation modeling for partial fingerprint identification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, pp. 72–87, Jan. 2011.

[10] A. K. Jain and J. Feng, "Latent fingerprint matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, pp. 88–100, Jan 2011.

[11] G. F. S. N. S. H. S. Prasad, "Use of ridge points in partial fingerprint matching," in *Proc. SPIE: Biometric Technology for Human Identification IV*, vol. 6539, April 2007.

[12] Y. Wang, J. Hu, and D. Phillips, "A fingerprint orientation model based on 2d fourier expansion (fomfe) and its application to singular-point detection and fingerprint indexing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, pp. 573–585, Apr. 2007.

[13] J. Jang, S. J. Elliott, and H. Kim, "On improving interoperability of fingerprint recognition using resolution compensation based on sensor evaluation," in *Proceedings of the 2007 International Conference on Advances in Biometrics*, ICB'07, (Berlin, Heidelberg), pp. 455–463, Springer-Verlag, 2007.

[14] X. Tan and B. Bhanu, "Fingerprint matching by genetic algorithms," *Pattern Recogn.*, vol. 39, pp. 465–477, Mar. 2006.

[15] Y. He, J. Tian, Q. Ren, and X. Yang, *Audio- and Video-Based Biometric Person Authentication: 4th International Conference, AVBPA 2003 Guildford, UK, June 9–11, 2003 Proceedings*, ch. Maximum-Likelihood Deformation Analysis of Different-Sized Fingerprints, pp. 421–428. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003.

[16] A. Ross, S. Dass, and A. Jain, "A deformable model for fingerprint matching," *Pattern Recogn.*, vol. 38, pp. 95–103, Jan. 2005.

[17] A. Lumini, D. Maio, and D. Maltoni, "Continuous versus exclusive classification for fingerprint retrieval," *Pattern Recogn. Lett.*, vol. 18, pp. 1027–1034, Oct. 1997.

[18] R. Cappelli, A. Lumini, D. Maio, and D. Maltoni, "Fingerprint classification by directional image partitioning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 5, pp. 402–421, 1999.

[19] X. Jiang, M. Liu, and A. Kot, "Fingerprint retrieval for identification," *Information Forensics and Security, IEEE Transactions on*, vol. 1, pp. 532–542, Dec 2006.

[20] M. Liu, X. Jiang, and A. Chichung Kot, "Efficient fingerprint search based on database clustering," *Pattern Recogn.*, vol. 40, pp. 1793–1803, June 2007.

[21] T. Liu, C. Zhang, and P. Hao, "Fingerprint indexing based on las registration," in *Image Processing, 2006 IEEE International Conference on*, pp. 301–304, Oct 2006.

[22] J. Li, W.-Y. Yau, and H. Wang, "Fingerprint indexing based on symmetrical measurement," in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, vol. 1, pp. 1038–1041, 2006.

[23] M. Liu and P.-T. Yap, "Invariant representation of orientation fields for fingerprint indexing," *Pattern Recogn.*, vol. 45, pp. 2532–2542, July 2012.

[24] T. Liu, G. Zhu, C. Zhang, and P. Hao, "Fingerprint indexing based on singular point correlation," in *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, vol. 3, pp. II–293–6, Sept 2005.

[25] R. S. Germain, A. Califano, and S. Colville, "Fingerprint matching using transformation parameter clustering," *IEEE Computational Science and Engineering*, vol. 4, pp. 42–49, Oct 1997.

[26] G. Bebis, T. Deaconu, and M. Georgiopoulos, "Fingerprint identification using delaunay triangulation," in *Information Intelligence and Systems, 1999. Proceedings. 1999 International Conference on*, pp. 452–459, 1999.

[27] B. Bhanu and X. Tan, "Fingerprint indexing based on novel features of minutiae triplets," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 5, pp. 616–622, 2003.

[28] T. yang Jea and V. Govindaraju, "Partial fingerprint recognition based on localized features and matching," 2005.

[29] X. Liang, A. Bishnu, and T. Asano, "A robust fingerprint indexing scheme using minutia neighborhood structure and low-order delaunay triangles," *Information Forensics and Security, IEEE Transactions on*, vol. 2, no. 4, pp. 721–733, 2007.

[30] A. Ross and R. Mukherjee, "Augmenting ridge curves with minutiae triplets for fingerprint indexing," April 2007.

[31] R. Singh, M. Vatsa, and A. Noore, "Fingerprint indexing using minutiae and pore features.," in *IPCV*, pp. 870–875, CSREA Press, 2009.

[32] O. Iloanusi, A. Gyaourova, and A. Ross, "Indexing fingerprints using minutiae quadruplets," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference on*, pp. 127–133, 2011.

[33] R. Cappelli, M. Ferrara, and D. Maltoni, "Fingerprint indexing based on minutia cylinder-code," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, pp. 1051–1057, May 2011.

[34] B. Yuan, F. Su, and A. Cai, "Fingerprint retrieval approach based on novel minutiae triplet features," in *Biometrics: Theory, Applications and Systems (BTAS), 2012 IEEE Fifth International Conference on*, pp. 170–175, 2012.

[35] A. Vij and A. M. Namboodiri, "Fingerprint indexing based on local arrangements of minutiae neighborhoods.," in *CVPR Workshops*, pp. 71–76, IEEE, 2012.

[36] A. Muoz-Briseo, A. G. Alonso, and J. H. Palancar, "Fingerprint indexing with bad quality areas.," *Expert Syst. Appl.*, vol. 40, no. 5, pp. 1839–1846, 2013.

[37] A. Gago-Alonso, J. HernáNdez-Palancar, E. RodríGuez-Reina, and A. MuñOz-BriseñO, "Indexing and retrieving in fingerprint databases under structural distortions," *Expert Syst. Appl.*, vol. 40, pp. 2858–2871, June 2013.

[38] Y. Wang, L. Wang, Y.-M. Cheung, and P. Yuen, "Learning compact binary codes for hash-based fingerprint indexing," *Information Forensics and Security, IEEE Transactions on*, vol. 10, pp. 1603–1616, Aug 2015.

[39] J. Feng and A. Cai, "Fingerprint indexing using ridge invariants," in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, vol. 4, pp. 433–436, 2006.

[40] J. B. de, A. M. Bazen, and S. H. Gerez, "Indexing fingerprint databases based on multiple features," in *Proceedings SAFE, ProRISC, SeSens 2001*, (Utrecht, The Netherlands), pp. 300–306, STW, November 2001.

[41] J. Feng and A. K. Jain, "Filtering large fingerprint database for latent matching," in *in Proc. Int. Conf. on Pattern Recognition (ICPR08)*, pp. 1–4, 2008.

[42] A. K. Jain and J. Feng, "Latent fingerprint matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 1, pp. 88–100, 2011.

[43] R. Cappelli and M. Ferrara, "A fingerprint retrieval system based on level-1 and level-2 features," *Expert Systems with Applications*, vol. 39, no. 12, pp. 10465 – 10478, 2012.

[44] A. A. Paulino, E. Liu, K. Cao, and A. K. Jain, "Latent fingerprint indexing: Fusion of level 1 and level 2 features," in *Biometrics: Theory, Applications and Systems*, (Washington, D.C.), 2013.

[45] A. Gyaourova and A. Ross, "A novel coding scheme for indexing fingerprint patterns," in *Proceedings of the 2008 Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition*, SSPR & SPR '08, (Berlin, Heidelberg), pp. 755–764, Springer-Verlag, 2008.

[46] R. Cappelli, M. Ferrara, and D. Maio, "Candidate list reduction based on the analysis of fingerprint indexing scores," *Information Forensics and Security, IEEE Transactions on*, vol. 6, pp. 1160–1164, Sept 2011.

[47] X. Shuai, C. Zhang, and P. Hao, "Fingerprint indexing based on composite set of reduced sift features," in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pp. 1–4, 2008.

[48] J. Hartloff, J. Dobler, S. Tulyakov, A. Rudra, and V. Govindaraju, "Towards fingerprints as strings: Secure indexing for fingerprint matching," in *Biometrics (ICB), 2013 International Conference on*, pp. 1–6, June 2013.

[49] U. Jayaraman, S. Prakash, and P. Gupta, "Use of geometric features of principal components for indexing a biometric database," *Mathematical and Computer Modelling*, vol. 58, no. 1-2, pp. 147 – 164, 2013.

[50] M. M. Chong, H. N. Tan, L. Jun, and R. K. Gay, "Geometric framework for fingerprint image classification," *Pattern Recognition*, vol. 30, no. 9, pp. 1475 – 1488, 1997.

[51] T. Kaniel and M. Mizoguchi, "Fingerprint preselection using eigenfeatures," in *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on*, pp. 918–923, Jun 1998.

[52] A. K. Jain, S. Prabhakar, and L. Hong, "A multichannel approach to fingerprint classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, pp. 348–359, Apr 1999.

[53] Y. Yao, G. L. Marcialis, M. Pontil, P. Frasconi, and F. Roli, "Combining flat and structured representations for fingerprint classification with recursive neural networks and support vector machines," *Pattern Recognition*, vol. 36, no. 2, pp. 397 – 406, 2003.

[54] Q. Zhang and H. Yan, "Fingerprint classification based on extraction and analysis of singularities and pseudo ridges," *Pattern Recognition*, vol. 37, no. 11, pp. 2233 – 2243, 2004.

[55] M. Liu, "Fingerprint classification based on adaboost learning from singularity features," *Pattern Recognition*, vol. 43, no. 3, pp. 1062 – 1070, 2010.

[56] Y. Wang, Q. Hao, A. Fatehpuria, L. G. Hassebrook, and D. L. Lau, "Quality and matching performance analysis of three-dimensional unraveled fingerprints," *Optical Engineering*, vol. 49, no. 7, pp. 077202–077202–10, 2010.

[57] V. Yalla, L. Hassebrook, R. Daley, C. Boles, and M. Troy, "Full-hand 3d non-contact scanner using sub-window-based structured light-illumination technique," *Proc. SPIE*, vol. 8371, pp. 83711O–83711O–15, 2012.

[58] M. Troy, L. Hassebrook, V. Yalla, and R. Daley, "Non-contact 3d fingerprint scanner using structured light illumination," *Proc. SPIE*, vol. 7932, pp. 79320C–79320C–13, 2011.

[59] B. Hiew, A. Teoh, and Y. Pang, "Touch-less fingerprint recognition system," in *Automatic Identification Advanced Technologies, 2007 IEEE Workshop on*, pp. 24–29, 2007.

[60] C. Lee, S. Lee, and J. Kim, "A study of touchless fingerprint recognition system," in *Structural, Syntactic, and Statistical Pattern Recognition* (D.-Y. Yeung, J. Kwok, A. Fred, F. Roli, and D. Ridder, eds.), vol. 4109 of *Lecture Notes in Computer Science*, pp. 358–365, Springer Berlin Heidelberg, 2006.

[61] R. D. Labati, A. Genovese, V. Piuri, and F. Scotti, "Virtual environment for 3-d synthetic fingerprints," in *Virtual Environments Human-Computer Interfaces and Measurement Systems (VECIMS), 2012 IEEE International Conference on*, pp. 48–53, 2012.

[62] G. Abramovich, K. Harding, S. Manickam, J. Czechowski, V. Paruchuru, R. Tait, C. Nafis, and A. Vemury, "Mobile, contactless, single-shot, fingerprint capture system," *Proc. SPIE*, vol. 7667, pp. 766708–766708–12, 2010.

[63] X. Pang, Z. Song, and W. Xie, "Extracting valley-ridge lines from point-cloud-based 3d fingerprint models," *IEEE Computer Graphics and Applications*, vol. 33, no. 4, pp. 73–81, 2013.

[64] A. Kumar and C. Kwong, "Towards contactless, low-cost and accurate 3d fingerprint identification," in *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '13, (Washington, DC, USA), pp. 3438–3443, IEEE Computer Society, 2013.

[65] R. Labati, A. Genovese, V. Piuri, and F. Scotti, "Fast 3-d fingertip reconstruction using a single two-view structured light acquisition," in *Biometric Measurements and Systems for Security and Medical Applications (BIOMS), 2011 IEEE Workshop on*, pp. 1–8, 2011.

[66] Y. Chen, F. Han, H. Liu, and J. Lu, "3d reconstruction from planar points: A candidate method for authentication of fingerprint images captured by mobile devices," in *Circuits and Systems (ISCAS), 2012 IEEE International Symposium on*, pp. 153–156, 2012.

[67] G. Parziale, E. Diaz-Santana, and R. Hauke, "The surround imager: a multi-camera touchless device to acquire 3d rolled-equivalent fingerprints," in *Proceedings of the 2006 international conference on Advances in Biometrics*, ICB'06, (Berlin, Heidelberg), pp. 244–250, Springer-Verlag, 2006.

[68] Y. Wang, Q. Hao, A. Fatehpuria, L. Hassebrook, and D. Lau, "Data acquisition and quality analysis of 3-dimensional fingerprints," in *Biometrics, Identity and Security (BIdS), 2009 International Conference on*, pp. 1–9, 2009.

[69] Y. Wang, L. Hassebrook, and D. Lau, "Data acquisition and processing of 3-d fingerprints," *Information Forensics and Security, IEEE Transactions on*, vol. 5, no. 4, pp. 750–760, 2010.

[70] V. Yalla, R. Daley, C. Boles, L. Hassebrook, K. Fleming, and M. Troy, "High-quality 3d fingerprint acquisition using a novel sub-window-based structured light illumination approach," *Proc. SPIE*, vol. 7797, pp. 77970R–77970R–11, 2010.

[71] Y. Chen, G. Parziale, E. Diaz-Santana, and A. Jain, "3d touchless fingerprints: Compatibility with legacy rolled images," in *Biometric Consortium Conference, 2006 Biometrics Symposium: Special Session on Research at the*, pp. 1–6, 2006.

[72] Y. Wang, D. L. Lau, and L. G. Hassebrook, "Fit-sphere unwrapping and performance analysis of 3d fingerprints," *Appl. Opt.*, vol. 49, pp. 592–600, Feb 2010.

[73] A. Fatehpuria, D. L. Lau, and L. G. Hassebrook, "Acquiring a 2d rolled equivalent fingerprint image from a non-contact 3d finger scan," *Proc. SPIE*, vol. 6202, pp. 62020C–62020C–8, 2006.

[74] C. B. Atkins, J. P. Allebach, and C. A. Bouman, "Halftone postprocessing for improved rendition of highlights and shadows," *J. Elec. Imaging*, vol. 9, pp. 200–0, 2000.

[75] S. Shafaei, T. Inanc, and L. Hassebrook, "A new approach to unwrap a 3-d fingerprint to a 2-d rolled equivalent fingerprint," in *Biometrics: Theory, Applications, and Systems, 2009. BTAS '09. IEEE 3rd International Conference on*, pp. 1–5, 2009.

[76] Q. Zhao, A. K. Jain, and G. Abramovich, "3d to 2d fingerprints: Unrolling and distortion correction.," in *IJCB* (A. K. Jain, A. Ross, S. Prabhakar, and J. Kim, eds.), pp. 1–8, IEEE, 2011.

[77] "Nist special database 27. http://www.nist.gov/srd/nistsd27.cfm."

[78] Y. Wang, J. Hu, and H. Schroder, "A gradient based weighted averaging method for estimation of fingerprint orientation fields," in *Digital Image Computing: Techniques and Applications, 2005. DICTA '05. Proceedings 2005*, dec. 2005.

[79] "Verifinger sdk. http://www.neurotechnology.com/verifinger.html," 2013.

[80] L. A. Zadeh, "Fuzzy logic: issues, contentions and perspectives," in *Acoustics, Speech, and Signal Processing, ICASSP-94., IEEE International Conference on*, vol. 6, 1994.

[81] "Nist special database 14. http://www.nist.gov/srd/nistsd14.cfm," 2013.

[82] "Nist biometric image software. http://www.nist.gov/itl/iad/ig/nbis.cfm," 2013.

[83] A. J. Tom Oswald, Kim Ward, "3-d fingerprint phantoms improve fingerprint-matching technology." http://msutoday.msu.edu/news/2014/3-d-fingerprint-phantoms-improve-fingerprint-matching-technology, 3 2014.

[84] J. Ortega-Garcia, J. Fierrez, F. Alonso-Fernandez, J. Galbally, M. R. Freire, J. Gonzalez-Rodriguez, C. Garcia-Mateo, J.-L. Alba-Castro, E. Gonzalez-Agulla, E. Otero-Muras, S. Garcia-Salicetti, L. Allano, B. Ly-Van, B. Dorizzi, J. Kittler, T. Bourlai, N. Poh, F. Deravi, M. W. Ng, M. Fairhurst, J. Hennebert, A. Humm, M. Tistarelli, L. Brodo, J. Richiardi, A. Drygajlo, H. Ganster, F. M. Sukno, S.-K. Pavani, A. Frangi, L. Akarun, and A. Savran, "The multiscenario multienvironment biosecure multimodal database (bmdb)," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 6, pp. 1097–1111, 2010.

[85] M. Ferrara, D. Maltoni, and R. Cappelli, "Noninvertible minutia cylinder-code representation," *Information Forensics and Security, IEEE Transactions on*, vol. 7, pp. 1727–1737, Dec 2012.

[86] "Contrast-limited adaptive histogram equalization (clahe). http://au.mathworks.com/help/images/ref/adapthisteq.html."

[87] W. Zhou, J. Hu, S. Wang, I. Petersen, and M. Bennamoun, "Partial fingerprint indexing: a combination of local and reconstructed global features," *Concurrency and Computation: Practice and Experience*, 2015.

[88] M. K. K. Mohammed Sayim Khalil, Dzulkifli Muhammad and K. Alghathbar, "Singular points detection using fingerprint orientation field reliability," *International Journal of Physical Sciences*, vol. 5, no. 4, pp. 352–357, 2010.

[89] M. I. Processing, "Morphological image processing.." https://www.cs.auckland.ac.nz/courses/compsci773s1c/lectures/ImageProcessing-html/topic4.htm.

[90] R. Cappelli, M. Ferrara, and D. Maltoni, "Minutia cylinder-code: A new representation and matching technique for fingerprint recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, pp. 2128–2141, Dec 2010.

[91] P. Indyk, "Nearest neighbors in high-dimensional spaces," 2004.

[92] K. Grauman and R. Fergus, *Machine Learning for Computer Vision*, ch. Learning Binary Hash Codes for Large-Scale Image Search, pp. 49–87. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.

[93] W. Zhou, J. Hu, I. Petersen, S. Wang, and M. Bennamoun, "A benchmark 3d fingerprint database," in *Fuzzy Systems and Knowledge Discovery (FSKD), 2014 11th International Conference on*, pp. 935–940, Aug 2014.

[94] B. S. Laboratory, "Mcc sdk 2.0." http://biolab.csr.unibo.it, 2015.