

Fingerprint Indexing Using Minutiae Based Invariable Set of Multidimensional Features

*Thesis submitted in partial fulfilment
of the requirements for the degree*

of

**Master of Technology
in
Information Technology**

by

Om Prakash Singh
Roll No: 09IT6003

Under the supervision of

Dr. D. Samanta



**School of Information Technology
Indian Institute of Technology, Kharagpur
West Bengal, India
May 2011**

Declaration

I certify that

- a. The work contained in the thesis is original and has been done by myself under the general supervision of my supervisor.
- b. The work has not been submitted to any other Institute for any degree or diploma.
- c. I have followed the guidelines provided by the Institute in writing the thesis.
- d. I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.
- e. Whenever I have used materials (data, theoretical analysis, and text) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references.
- f. Whenever I have quoted written materials from other sources, I have put them under quotation marks and given due credit to the sources by citing them and giving required details in the references.

Om Prakash Singh



**School of Information Technology
Indian Institute of Technology, Kharagpur
West Bengal, India
May 2011**

Certificate

This is to certify that the thesis titled **Fingerprint Indexing Using Minutiae Based Invariable Set of Multidimensional Features** submitted by **Om Prakash Singh**, Roll No: **09IT6003**, in the School of Information Technology, Indian Institute of Technology, Kharagpur, for the award of the degree of **Master of Technology in Information Technology**, is a record of an original research work carried out by him under my supervision and guidance. The thesis fulfills all requirements as per the regulations of this institute. Neither this thesis nor any part of it has been submitted for any degree or academic award elsewhere.

Dr. D. Samanta
Assistant Professor
School of Information Technology
IIT Kharagpur
May, 2011

Abstract

Fingerprint identification process requires match the query fingerprint with all fingerprints in the large pool of fingerprints. Exhaustive search for fingerprint identification results large response time. In the area of fingerprint identification, where size of data is huge an exhaustive search is not acceptable due to large response time. Therefore fingerprint identification process demands a good clustering and indexing technique to avoid exhaustive search and narrow down the search space. In this work we propose a cluster based indexing technique with invariable set of features for fingerprint identification. We present local topology of minutiae using two closest point triangle for index key generation. This approach considers less or equal number of triangle $O(N)$ per fingerprint, where N is the number of minutiae in fingerprint. Small number of triangles requires less computation overhead. Local topology based Index key makes this approach suitable for latent, partial and distorted fingerprints in an identification process. Features which are invariant to rotation and scaling enable the approach to use with fingerprint form different devices and sensors without applying any image alignment and registration technique. This approach uses well known k -mean clustering algorithm for clustering the index keys. The approach produces the candidate set for reduces search space by ensuring the low penetration rate. The proposed approach has been tested on FVC 2004 database. The database contains four set of fingerprint form different devices, each set have 800 fingerprints (8 fingerprints from 100 persons).

The experimental results substantiate good performance of the proposed approach.

Acknowledgements

I would like to express my sincere gratitude to my supervisor Dr. D. Samanta who gave me his guidance and ample freedom to express myself in my work. This work is indebted to his well thought advice and constant support. I am deeply grateful to him for his counselling.

I also express my heartfelt thanks to Somnath Dey, research scholar of the School of Information Technology for their invaluable suggestions. I am thankful for the discussing and valued suggesting which I had form all my friends, fellow researchers and seniors.

I would also like to share my sincere thanks to staffs of our department for providing facility to caring out the project and for their kind support.

I will forever be indebted to my family for their unconditional love and support. I would take this opportunity to thank my mother for her prayer, my father who will forever be with us in spirit, and my brother for his constant guidance and motivation. Without them I would not have made it this far.

Om Prakash Singh

Contents

List of Figures	iv
List of Tables.....	vi
Chapter 1.....	1
Introduction	1
1.1 Biometrics.....	1
1.2 Mode of Operations	3
1.3 Fingerprint as Biometrics	4
1.2 Issues and Challenges	7
1.3 Motivation.....	9
1.4 Problem Statement.....	10
1.5 Organization of Thesis	11
Chapter 2.....	13
Related Work	13
2.1 Fingerprint Classification.....	13
2.2 Fingerprint Indexing.....	16
2.3 Conclusion.....	20
Chapter 3.....	21
Fingerprint Pre-processing	21
3.1 Normalization.....	22
3.2 Segmentation	22
3.3 Orientation Estimation	24
3.4 Ridge Frequency Estimation.....	28
3.5 Ridge Filtering	29
3.4 Binarization and Thinning.....	32
3.5 Conclusion.....	34
Chapter 4.....	35
Feature Extraction	35

4.1 Minutiae Feature Extraction	35
4.2 Post-processing.....	36
4.3 Orientation Estimation of Minutiae.....	37
4.4 Conclusion.....	39
Chapter 5.....	41
Fingerprint Indexing.....	41
5.1 Framework of the Proposed Approach.....	41
5.2 Preliminaries	44
5.2.1 <i>k</i> -means data clustering.....	44
5.2.2 <i>kd</i> -Tree	46
5.3 Two Closest Points Triangulation.....	48
5.4 Feature Extraction for Index Vector	49
5.5 Creation of Population of index Space.....	53
5.6 Querying.....	54
5.3 Conclusion.....	58
Chapter 6.....	59
Experimental Results	59
6.1 Fingerprint Database	59
6.2 Implementation Environment.....	60
6.3 Evaluation Metrics	60
6.4 Experimental Result of Image Enhancement	61
6.5 Experimental Result of Minutiae Extraction	62
6.6 Experimental Evaluation of Indexing	62
6.7 Analysis of Cluster Population in the Index Space Model.....	67
6.8 Comparison with Linear Search, Clustered Search and <i>kd</i> -tree Search.....	69
6.9 Comparison with Existing Methods.....	71
6.10 Conclusion.....	72
Chapter 7.....	74

Summary and Future Work.....	74
Bibliography	77

List of Figures

Figure 1-1: Different steps of a biometric authentication system.....	2
Figure 1-2: Human fingerprint ridge and valley	5
Figure 1-3: Fingerprint local features and global features [5].....	6
Figure 2-1: The five commonly used fingerprint classes [5]	15
Figure 3-1: Normalization process of fingerprint image	23
Figure 3-2: Fingerprint image segmentation using variance and thresholding	25
Figure 3-3: Even symmetric Gabor filter in spatial domain.....	31
Figure 3-4: Gabor filtered fingerprint image with different parameter value of σ_x and σ_y	32
Figure 3-5: Binarization and thinning of fingerprint image.....	33
Figure 4-1: Crossing number for ridge ending and bifurcation [5]..	36
Figure 4-2: Orientation of minutiae using 7×7 mask (in this case is 225 degree)	38
Figure 4-3: Convention for determining angle of minutiae [5].....	38
Figure 4-4: Ridge ending/bifurcation duality [5]	39
Figure 5-1: Linear search approach.....	42
Figure 5-2: Clustered search approach.....	43
Figure 5-3: Clustered <i>kd</i> -tree index search.....	44
Figure 5-4: Two dimensional <i>kd</i> -tree	47
Figure 5-5: Two closest point triangulation.....	49
Figure 5-6: Process of feature extraction for index vector	50
Figure 5-7: Component of minutiae triplet.....	51

Figure 5-8: Creation of index space population.....	54
Figure 5-9: Querying fingerprint on cluster search approach	57
Figure 5-10: Querying fingerprint on linear search approach.....	57
Figure 5-11: Querying fingerprint on clustered kd-tree index search approach.....	58
Figure 6-1: Performance of proposed approach on FVC-2004(DB1) with variable number of fingerprint per individual	64
Figure 6-2: Performance of proposed approach on FVC-2004(DB2) with variable number of fingerprint per individual	65
Figure 6-3: Performance of proposed approach on FVC-2004(DB3) with variable number of fingerprint per individual	66
Figure 6-4: Performance of proposed approach on FVC-2004(DB4) with variable number of fingerprint per individual	67
Figure 6-5: User distribution in cluster for FVC2004 DB1	68
Figure 6-6: User distribution in cluster for FVC2004 DB1	68
Figure 6-7: User distribution in cluster for FVC2004 DB1	69
Figure 6-8: User distribution in cluster for FVC2004 DB1	69
Figure 6-9: Comparison with linear search, clustered search and kd-tree search.....	70
Figure 6-10: Comparison of proposed approach with minutiae triplete [33] and directional field approach [33]	72
Figure 6-11: Comparison of proposed approach with fingerprint code [33] and orientation field approach [34]	72

List of Tables

Table 6-1: FVC-2004 fingerprint database set -A	60
Table 6-2: Average score of original image and Gabor filtered image using NFIQ tool	61
Table 6-3: Average number of minutiae extracted from different database	62
Table 6-4: Performance of proposed approach on FVC-2004(DB1) with variable number of fingerprint per individual	63
Table 6-5: Performance of proposed approach on FVC-2004(DB2) with variable number of fingerprint per individual	64
Table 6-6: Performance of proposed approach on FVC-2004(DB3) with variable number of fingerprint per individual	65
Table 6-7: Performance of proposed approach on FVC-2004(DB4) with variable number of fingerprint per individual	66
Table 6-8: Comparison with linear search, clustered search and <i>kd</i> - tree search.....	70
Table 6-9: Comparison of proposed approach with ross et al. [34]	71

Chapter 1

Introduction

In this chapter, we discuss the basic concept of biometric systems and the role of fingerprint in the area of biometric. The chapter is organized as follows. Section 1.1 gives basic introduction of biometric systems and explains basic steps of an authentication system. In Section 1.2, we describe the features and characteristic of the fingerprint biometric. Section 1.3 lists some issues and challenges facing in biometric system. Motivation behind carrying out this project is discussed in Section 1.4. In Section 1.5, we give the formal definition of the problem statement. Finally, Section 1.6 gives the organization of the thesis.

1.1 Biometrics

Biometrics comprises methods for uniquely recognizing human based upon one or more intrinsic physical or behavioural traits [1]. In computer science, in particular, biometrics is used as a form of identity access management and access control. It is also used to identify individuals in groups that are under surveillance.

In an increasingly digital world, reliable personal authentication has become an important human computer interface activity. National security, e-commerce and access to computer networks are some examples where establishing a person's identity is vital. Existing security measures rely on knowledge based approaches like passwords or token-based approaches such as swipe cards and password to control access to physical and virtual spaces. Such methods are not very secure. Tokens such as badges and access cards may be shared or stolen. Passwords and PIN numbers may be stolen electronically. Biometrics such as fingerprint, face and voice print offers means of reliable personal authentication that can address these problems and is gaining citizen and government acceptance.

A biometric system is a pattern recognition system that operates by acquiring biometric data from an individual, extracting a feature set from the data acquired, and comparing this feature set against the template set stored in the database. A typical biometric system have mainly four component acquisition, feature extraction, database and matching as shown in Figure 1.1.

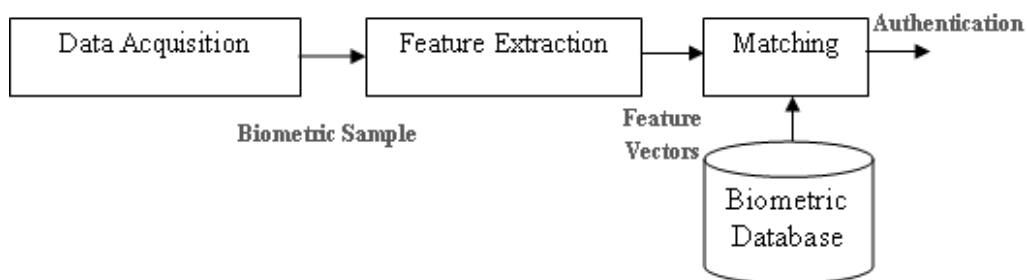


Figure 1-1: Different steps of a biometric authentication system

1. **Acquisition:** This step acquires biometric information from an individual. Acquisition is done by various kind of sensors for example camera for face recognition system, optical fingerprint scanner, for fingerprint and micro phone for voice recognition system. Then, some preprocessing technique is applied to improve the quality of enquired.
2. **Feature extraction:** Discriminatory features set are extracted from the acquired biometric in this stage. For example location of minutiae points and orientation, Location of core and delta point is used as fingerprint feature set. Geometric property is used for palm print recognition.
3. **Biometric database:** The database stores feature information with the identity of the person of all enrolled users. The information is stored as stander template in database.
4. **Matching:** The features extracted from a query biometric are compared against the stored template in database. In the matching process each comparison generates some score (called matching score) based on matching score we can confirm/deny the identity of a person.

1.2 Mode of Operations

A biometric system can operate in the following three modes:

- **Enrollment:** In this mode, biometric system enrolls the users and stored the features in the biometric database.

- **Verification:** A one to one comparison of a captured biometric with a stored template to verify that the individual is who he claims to be.
- **Identification:** A one too many comparison of the captured biometric against a biometric database in attempt to identify an unknown individual. The identification only succeeds in identifying the individual if the comparison of the biometric sample to a template in the database falls within a previously set threshold.

1.3 Fingerprint as Biometrics

Fingerprints have been used for over a century and are the most widely used form of biometric identification. Fingerprint identification is commonly employed in forensic science to support criminal investigations and in biometric systems such as civilian and commercial identification devices [2].

Fingerprints consist of a unique pattern of *ridges* and *valleys*. A ridge is defined as a single curved segment. Generally the ridges are depicted in dark/black colour. Valley is the region between two adjacent ridges and Valleys are represented in bright/white colour. Figure 1-2 shows the ridge and valley in a fingerprint. All the ridges run parallel to each other. It is observed that ridges vary in width from $100\mu m$, for very thin ridges, to $300\mu m$ for thick ridges. Generally, the period of a ridge/valley cycle is about $500\mu m$ [3] and inter-ridge distance, in a 500dpi image, is approximately 10 pixels [4]. The unique pattern of ridges and valleys characterized the fingerprint as biometric.



Figure 1-2: Human fingerprint ridge and valley

Features of fingerprint can be broadly classified in to way *local feature* and *global feature*

Local Features: Local feature of fingerprint is known as minutiae. Minutiae are the local discontinuities in the ridge flow pattern. Minutia provides the features that are used for identification. Details such as the type, orientation, and location of minutiae are taken into account when performing minutiae extraction. Based on the structure of ridge we can classify minutia in following category Ridge Iceland, Ridge end, Ridge dot, Ridge enclose, and ridge bifurcation [1]. Figure 1-3 shows the structure of all kind of ridge feature.

Global Features: The ridges and valleys on a fingerprint exhibit a smooth flow-like pattern. The orientations of ridge change smoothly from one region to another. But, in some region, the ridge orientations change sharply. Such region forms global feature of fingerprint. The global features of fingerprint are termed as Singular Point regions. These singular points can be classified in two broad categories *Core Point* and *Delta Point* [5]. The core region consists of arches and loops caused by sharp turnings in the ridge orientations, while the delta region is formed by three non-intersecting and tangential arches. Both these regions are shown in Figure 1-3.

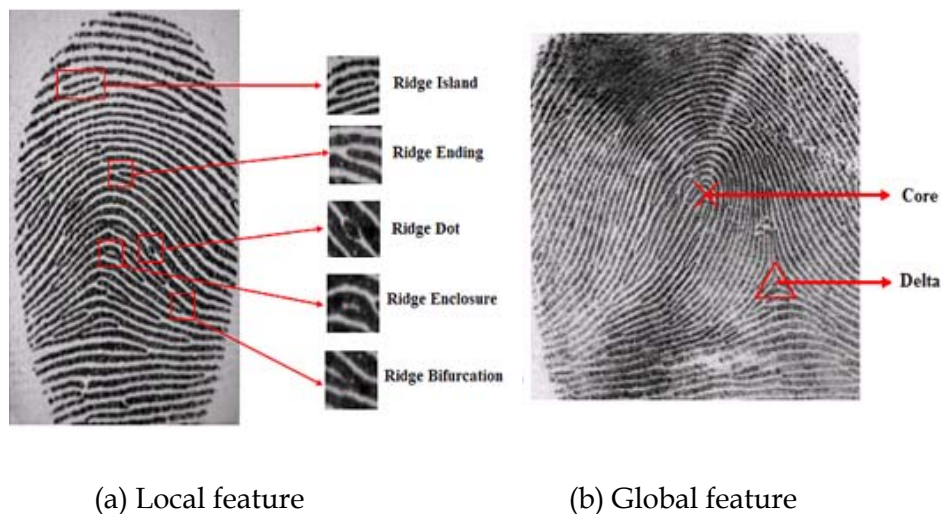


Figure 1-3: Fingerprint local features and global features [5]

There are several advantages and disadvantages associated with using fingerprint as a bio-metric. The advantages and disadvantages are summarized as following

Advantages:

- Fingerprint is considered to be unique to an individual and remain unchanged throughout the life of person
- Most people are familiar with the use of fingerprint for authentication and access control.
- Unlike other biometric scanners (e.g., iris and retina scanners), fingerprints scanners are easy to use.
- Most fingerprint scanners today are inexpensive and easy to install.

Disadvantages:

- Fingerprint spoofing is a major problem in unsupervised, fingerprint based biometric systems.
- The quality of fingerprint may suffer due to weather conditions, thus degrading recognition accuracy.
- Fingerprint is easily damaged by accident as it is the outer part of a human body.

1.2 Issues and Challenges

There are several issues and challenges associated with any biometric based authentication system. Some of the issues and challenges of a biometric system (specially fingerprint biometric) are listed in following

Inter and Intra class variability: At different time instances a user may have different sample of one biometrics which comes under *intra class variability*. For example, one user may have different facial expression at different time instance. This is happen due to incorrect interaction with

sensor or modification in sensor characteristics. Second is *inter class variability* which is the case of two different person have some same biometric value. For example, twins may have same face.

Noise and distorted sensor data: Defective sensors or an unfavourable ambient/physiological condition captures the noisy samples which may lead an authentication system fail.

Different source of data: Using different sensor at different time can add variance in biometric data. Images from different sources can make identification system less accurate.

Higher dimension feature set: Extracted feature set from biometric traits are of higher in dimension. Maintaining such higher dimension lead high computation time which may not be realistic.

Unordered feature set: Biometric data representation does not have any natural characteristic for sorting or ordering so, it is hard to define any indexing or clustering technique.

Partial and latent Impression: In some biometric application, acquired fingerprint may not be complete due to size of sensor or recovered fingerprint.

Non universality: This is the case when the certain biometrics does not exhibits all feature set for example approximately 4% of fingerprint [5] images have been observed to have poor ridge details.

Attack on biometric system: One of the major attacks on biometric system is spoof attack which is circumvent the system by imitation or using artificial traits.

1.3 Motivation

In today's world most of the information system require authentication. Reliable authentication schemes are most important for any system. Knowledge based system (e.g. password) and token based system (e.g. identity card) are example of security system which are used more commonly for authentication. These security systems do not ensure that the provided service is used by legitimate user only. These security systems also can be easily breach because identity card can be forged and password can be crack.

Biometric such as fingerprint, iris, face etc. are good alternative of traditional security system. Among all biometric traits fingerprint is most popular one it has been used for over a century and are the most widely used form of biometric authentication due to its uniqueness and permanence. Fingerprint identification is commonly employed in forensic science to support criminal investigations, and in biometric systems such as civilian and commercial identification devices. These identification systems have huge underlying biometric database. The goal of an identification system is to identify the individual form large set of users already enrolled in biometric database. Though the current biometric system have good response time in order to determine the identity of a

person in small database, but when it's comes under the large database, their performance is not good because of large response time.

There are three ways an identification system can be operated. The simplest approach is exhaustive search. In this approach, each enrolled fingerprint has to verify against the query fingerprint. Second approach is classification. In classification approach, fingerprint in database is divided in some predefined class. Before beginning the identification process, system determines the class of query fingerprint then search within specified class. This approach can reduce the search space with some degree and hence improve the response time. The third approach is to create a candidate set on the basis of some broad feature set (the candidate set consists of those fingerprints form database which are most likely to match with query fingerprint) then query fingerprint searches within the candidate set for a match. This approach considerably reduces the search space and has better response time than other two approaches.

1.4 Problem Statement

In order to search for a match in large database to indentify the identity of a person, the problem is to reduce the search space. Reduce set of fingerprint is called candidate set. The probability of given a query fingerprint match with the one of fingerprint in candidate set is called *Hit Rate*. The fraction of database is touched in order to find a match is called *Penetration Rate*. It is desirable to have high hit rate for low penetration rate. High hit rate is a parameter to define the accuracy of any identification system. Size of candidate set should be as low as possible

that means low penetration rate, to ensure high hit rate. Generation of candidate should have low computation overhead and response time. Formally, the problem can be defined as follows:

“Produce the candidate set C form given a fingerprint database D to match query fingerprint Q such that if Q has a match M in database D i.e. $M \in D$ and M matches Q then C must contains M with a very high probability i.e. $Pr (M \in C)$ should be very high and size of candidate set C should be very less as compare to database size i.e. $size(C) \ll size(M)$ ”

1.5 Organization of Thesis

The rest of the thesis is organized as follows

Chapter-2: In this chapter we present the current available fingerprint classification and indexing technique.

Chapter-3: The chapter gives method applied for fingerprint image enhancement.

Chapter-4: Basic feature extraction techniques are discussed in this chapter.

Chapter-5: This chapter presents our proposed approach for fingerprint indexing.

Chapter-6: All experimental results are discussed in this chapter.

Chapter-7: This chapter contains summary of our work and possible future work in field fingerprint clustering and indexing.

Chapter 2

Related Work

Fingerprint identification systems require search through a large database of fingerprint entries for possible matches based on the given query fingerprint. Each entry has an associated identity. Due to the large number of entries in the database, one to one matching of the query print with each fingerprint in the database would be computationally infeasible. Therefore, a filtering process is usually invoked in order to reduce the search space. Filtering can be achieved by two different approaches: *classification* and *indexing*.

Section 2.1 of this chapter tells us about fingerprint classification and some classical fingerprint classification techniques. In Section 2.2, we discuss about indexing techniques and state of art of fingerprint indexing. Finally, in Section 2.4, we give summery of this chapter.

2.1 Fingerprint Classification

Classification involves partitioning of database into multiple predefined classes and comparing the query fingerprint with fingerprints belonging to the class assigned to the query fingerprint. First fingerprint

classification rules were proposed in 1823 by Purkinje [5] who classified fingerprints into nine categories (transverse curve, central longitudinal, oblique stripe, oblique loop, almond whorl, spiral whorl, ellipse, circle, and double whorl) according to the global ridge configurations. The first scientific study on fingerprint classification was made by Francis Galton in 1892 [5], who divided the fingerprints into three major classes (arch, loop, and whorl) and further divided each category into subcategories. In 1900, Edward Henry [5] proposed refined version of Galton's classification by increasing the number of classes. This classification is known as Galton–Henry classification. This scheme is the most popular scheme which has been adopted in several countries. Most of the classification schemes currently used by law enforcement agencies worldwide are variants of the Galton–Henry classification scheme. Figure 2-1 shows the five most common classes of the Galton–Henry classification scheme (arch, tented arch, left loop, right loop, and whorl):

Fingerprint classification has many different approaches. Most of the existing fingerprint classification methods [6] [7] [8] are coarsely assigned to one of these categories: rule-based, syntactic, structural, statistical, neural network-based and multi-classifier approaches. These approaches are described as follows.

1 Rule-based approach: A fingerprint can be simply classified according to the number and the position of the singularities. This is the approach commonly used by human experts for manual classification, therefore several authors proposed to adopt the same technique for automatic classification. This approach is used in [9].

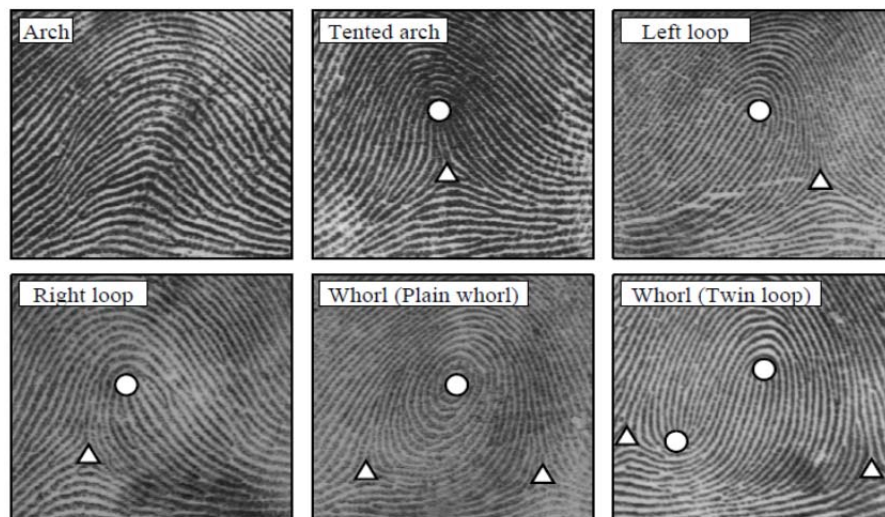


Figure 2-1: The five commonly used fingerprint classes [5]

2 Syntactic approaches: In this approach patterns are represented as symbols.

A grammar is defined for each class and a parsing process is responsible for classifying each new pattern [7]. Due to large variations in fingerprint patterns, a complicated grammar is required.

3 Structural approaches: Structural approaches are based on the relational organization of low-level features into higher-level structures. This relational organization is represented by means of symbolic data structures, such as trees and graphs, which allow a hierarchical organization of the information. The orientation image is well suited for structural representation.

4 Statistical approaches: In this approach, a fixed-size numerical feature vector is derived from each fingerprint and a general-purpose statistical classifier is used for the classification. Some of the most

widely adopted statistical classifiers are: Bayes decision rule, k -nearest neighbour, and Support Vector Machines (SVM) [8].

5 Neural network-based approaches: In this approach, multilayer perceptrons are used to classify the fingerprint in different classes [6].

2.2 Fingerprint Indexing

Fingerprint indexing is also referred as continuous classification. This approach does not partition fingerprints into disjoint classes, but rather represents them as feature vectors, such that, similar fingerprints are mapped to close points in a high-dimensional space. Retrieval is performed by matching the query fingerprint with all fingerprints in the database whose feature vectors lie in its vicinity.

Germain et al. [10] use minutiae triplet features for indexing using the FLASH technique (Fast Look up Algorithm for String Homology) [11]. The Flash algorithm uses a higher dimensional indexing scheme than geometric hashing by adding invariant properties of the feature subset to the index. Nine Dimension index [length of each side (3), the ridge count between each pair (3), angles measured with respect to the fiducial side (3)] based on minutiae are used as feature. The Flash algorithm uses redundant combinations of three feature points when forming indices. An exhaustive listing of the possible combinations of three feature points requires ${}^n C_3$ entries, where n is the number of minutiae. Hence $O(N^3)$ triangle is needed to evaluate. To keep the number of indices generated within bounds, the algorithm restricts the “acceptable” combinations of feature points used to form an index. It only uses triplets for which the

distances separating each pair of points fall into the specified range. This indexing-based approach accumulates evidence about a model by casting a vote for every entry stored in the indexed locations and by "histogramming" the entries to pick the ones which have received a high number of votes. The problem with this approach is that it takes into consideration only the number of votes received by a particular entry and does not consider whether these votes are consistent among themselves.

Bebis et al. [12] introduce Delaunay triangulation of minutiae point to reduce the number of the evaluated triangle. The key characteristics of the Delaunay triangulation of a set of points are that it is unique. Also, it can be computed efficiently in $O(N \log N)$ time [12]. One problem is that it is sensitive to noise and distortions (e.g. introduced by missing or spurious minutiae points however, both noise and distortion have only a local effect on it. This means that correct identification will be possible if some region of the fingerprint has not been seriously affected. For voting in the transformation space method proposed by Lamiroy and Gros [13] is used. The key idea behind this approach is to consider transformations which form large clusters in the transformation space. 8-Dimension integer feature vector is used (six dimensions for the parameters of the transformation, one for the *person_ID* and one for the *imprint_ID*). Array is used to store index. Hashing is used as indexing approach.

Bhanu et al. [14] also use triangulation of minutiae point with different feature set. They used maximum length of 3 sides, median and minimum angles, triangle handedness, type and direction, ridge count minutiae Density.

Ross et al. [15] add ridge curve with Delaunay triangulation of minutiae triplets. Two sets of features are extracted from each triplet. The first set of features correspond to the geometry of the triangle generated by the triplet, while the second set pertains to the shape of the ridges associated with the three minutiae points constituting the triplet. Three feature for geometry of triangle. The three geometry based features extracted from the triangles are invariant to rotation and scaling. The second set of features is based on fitting a quadratic curve to the ridges associated with each triplet. For every minutiae point detected in the fingerprint, a ridge tracing algorithm is invoked that gives a set of points lying on a ridge containing the minutiae point.

Mansukhani et al. [16] uses the fingerprint dataset in a tree-based structure. Each non-leaf node represents an arrangement of minutiae points based on the path from the root to that particular node. Fingerprint templates are enrolled at the leaf nodes. One or more templates can be enrolled at each leaf. Also, a fingerprint might be enrolled at multiple leaf nodes in the tree, representing multiple minutiae paths corresponding to various different minutiae arrangements in a single fingerprint image. A single tree for all fingerprints in the database, and all enrolled fingerprints will have some of the leaves representing them. The tree provides true index structure for fingerprint database; when to perform identification, traverse the tree and find leaves corresponding to minutia paths in the test fingerprint template; found leaves will most likely point to the matching enrolled fingerprints in the database. To arrange the minutiae points together, the concept of minutiae bins is used. Binning is

done based on the relative features of a particular minutiae point and its nearest neighbour. The feature set used for binning the points is distance of minutiae from current minutiae and its angular position. A discretization technique has been used to handle continuous featured values. Hence, the number of bins for a particular minutiae point will depend on the number and the properties of each individual feature selected.

Fingerprint indexing can be categorized in follow two ways:

- 1 Fingerprint sub-classification:** This approach is adopted by fingerprint experts to perform manual fingerprint matching in forensic applications. However, the rules for such sub-classifications are quite complicated and are dependent on the finger (thumb, index, middle, etc.) [5]. Implementing an automated fingerprint sub-classification system is much more difficult than classifying the fingerprints into the traditional classes (left loop, right loop, whorl, twin loop, arch, tented arch). Therefore this is not practical to be used in automatic fingerprint classification systems.
- 2 Fingerprint continuous-classification techniques:** This approach does not partition fingerprints into disjoint classes, but rather represents them as feature vectors, such that, similar fingerprints are mapped to close points in a high-dimensional space. Retrieval is performed by matching the query fingerprint with all fingerprints in the database whose feature vectors lie in its vicinity [5].

2.3 Conclusion

Fingerprint classification and indexing has been studied over last three decades. Different solutions have been given to meet the requirement. Fingerprint classification involves partitioning the database into multiple classes and comparing the query fingerprint with fingerprints belonging to the class assigned to the query fingerprint. Other approach is called fingerprint indexing. It is also referred as continuous classification. This approach does not partition fingerprints into disjoint classes, but rather represents them as feature vectors, such that, similar fingerprints are mapped to close points in a high-dimensional space. Retrieval is performed by matching the query fingerprint with all fingerprints in the database whose feature vectors lie in its vicinity.

Chapter 3

Fingerprint Pre-processing

Input fingerprint image of a biometric system does not necessarily be a good quality. To extract the feature from fingerprint image it need to enhance. This makes task of extracting features form fingerprint image easy and ensure the quality of extracted features. There are many techniques available for fingerprint image enhancement [17] [18] [19]. For fingerprint image we use the methods proposed by [20]. This chapter is about the different step to enhance the fingerprint image. In Section 3.2, we discuss the normalization process of fingerprint image to standardize the intensity value of image in desirable range. Section 3.3 explains method of segmentation, where we separate foreground fingerprint impression from undesirable background portion. Method of enhancing the quality of ridge valley structure is discussed in Section 3.4. Section 3.5 describes the method of binarization and thinning of enhanced image. Finally, we summarize the chapter in Section 3.6.

3.1 Normalization

The main purpose of normalization is to reduce the variations in gray-level values along ridges and valleys. Normalization is a pixel-wise operation. It does not change the clarity of the ridge and valley structures [20]. Normalization is used to standardize the intensity values in an image by adjusting the range of grey-level values so that it lies within a desired range of values.

Let $I(i, j)$ represents the grey-level value at pixel (i, j) , and $N(i, j)$ represent the normalized grey-level value at pixel (i, j) . The normalized image is done using Eq. (3.1) [21].

$$N(i, j) = \begin{cases} M_0 + \sqrt{\frac{V_0(I(i, j) - M)^2}{V}} & \text{if } I(i, j) > M \\ M_0 - \sqrt{\frac{V_0(I(i, j) - M)^2}{V}} & \text{Otherwise} \end{cases} \quad (3.1)$$

Where M and V are the estimated mean and variance of $I(i, j)$, respectively, and M_0 and V_0 are the desired mean and variance values, respectively. Figure 3-1 shows the effect of normalization process on image and on histogram of image. Normalize image is used in further steps of image enhancement process.

3.2 Segmentation

Segmentation is the process of separating the foreground regions in the image from the background regions. The foreground regions correspond to the clear fingerprint area containing the ridges and valleys, which is the area of interest. The background corresponds to the regions

outside the borders of the fingerprint area, which do not contain any valid fingerprint information. For minutiae we should consider only valid region of fingerprint image otherwise it would result many false minutiae.

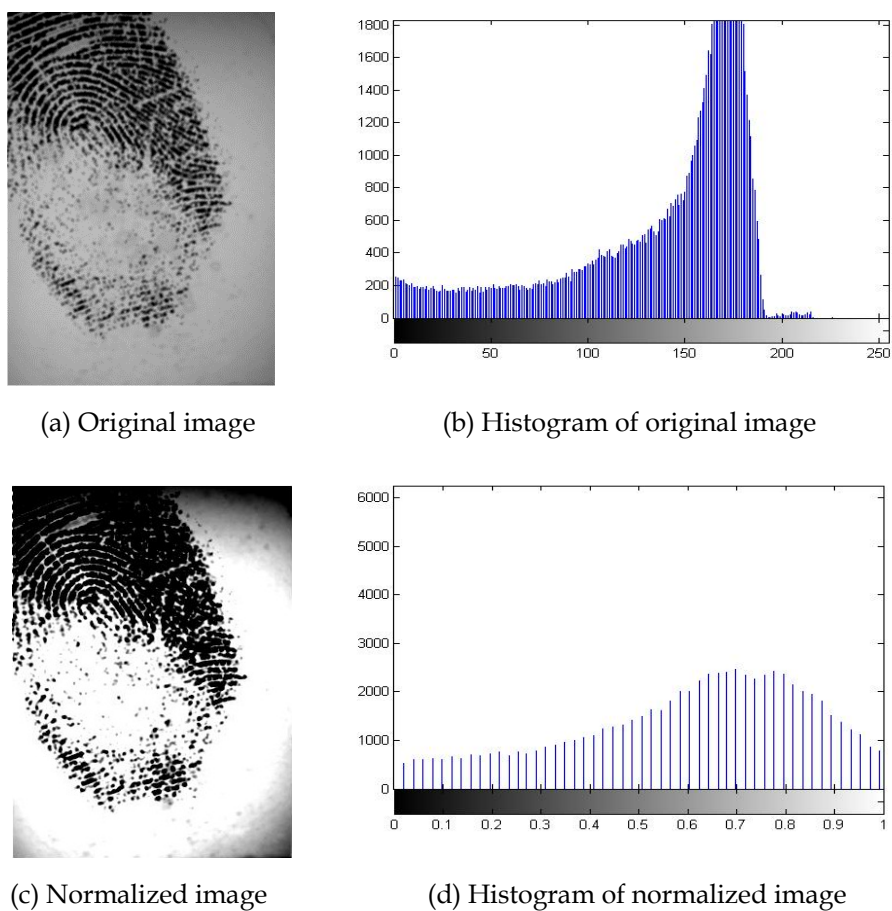


Figure 3-1: Normalization process of fingerprint image

In a fingerprint image, the background regions generally exhibit a very low grey-scale variance value, whereas the foreground regions have

a very high variance. Hence, a method based on variance threshold is used to perform the segmentation.

First, the normalized image is divided into non-overlapping blocks and the grey-scale variance is calculated for each block in the image. If the variance is less than the global threshold, then the block is assigned to be a background region; otherwise, it is assigned to be part of the foreground. By this process we get mask for defining the fingerprint region which is used in later stage of image processing and eliminating false minutiae.

The grey-level variance for a block of size $W \times W$ is defined in Eq. (3.2) [21].

$$V(k) = \frac{1}{W^2} \sum_{i=0}^{w-1} \sum_{j=0}^{w-1} (I(i, j) - M(k))^2 \quad (3.2)$$

Where $V(k)$ is the variance for block k , $I(i, j)$ is the grey-level value at pixel (i, j) and $M(k)$ is the mean grey-level value for the block k . Figure 3-2 shows the step wise segmentation process. Figure 3-2(b) is computer on block wise process of calculating local variance field. Figure 3-2(c) is result of applying threshold of variance field image. Figure 3-2(d) is result of applying mask generated in previous step with original image.

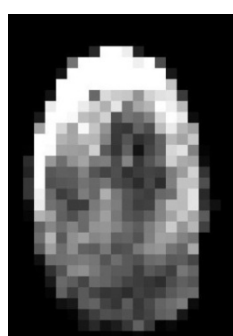
3.3 Orientation Estimation

The orientation field of a fingerprint image defines the local orientation of the ridges contained in the fingerprint. The orientation estimation is a fundamental step in the enhancement process as the

subsequent Gabor filtering stage relies on the local orientation in order to effectively enhance the fingerprint image. The least mean square estimation method employed by Maltoni et al. [21] is used to compute the orientation image. The steps for calculating the orientation at pixel (i, j) are as follows [20]:



(a) Original image



(b) Local variance



(c) Mask generated by thresholding



(d) Segmented image

Figure 3-2: Fingerprint image segmentation using variance and thresholding

1. First we divide normalized image $N(i, j)$ in blocks of size $w \times w$ centered at pixel at (i, j) .
2. For each pixel in the block, we compute the gradients $\partial_x(i, j)$ and $\partial_y(i, j)$ which are the gradient magnitudes in the x and y directions, respectively.
3. The local orientation at pixel (i, j) can then be estimated using the following Eq. (3.3), (3.4) and (3.5).

$$V_x(i, j) = \sum_{u=i-\frac{w}{2}}^{i+\frac{w}{2}} \sum_{v=j-\frac{w}{2}}^{j+\frac{w}{2}} 2\partial_x(u, v)\partial_y(u, v) \quad (3.3)$$

$$V_y(i, j) = \sum_{u=i-\frac{w}{2}}^{i+\frac{w}{2}} \sum_{v=j-\frac{w}{2}}^{j+\frac{w}{2}} \partial_x^2(u, v)\partial_y^2(u, v) \quad (3.4)$$

$$\theta(i, j) = \frac{1}{2} \tan^{-1} \left(\frac{V_y(i, j)}{V_x(i, j)} \right) \quad (3.5)$$

Where $\theta(i, j)$ is the least square estimate of the local orientation at the block centred at pixel (i, j) .

4. We smooth the orientation field in a local neighbourhood using a Gaussian filter. The orientation image is first converted into a continuous vector field, which is defined in Eq. (3.6) and Eq. (3.7).

$$\varphi_x = \cos(2\theta(i, j)) \quad (3.6)$$

$$\varphi_y = \sin(2\theta(i, j)) \quad (3.7)$$

Where φ_x and φ_y are the x and y components of the vector field, respectively. After the vector field has been computed, Gaussian smoothing is then performed using Eq. (3.8) and Eq. (3.9).

$$\varphi'_x(i, j) = \sum_{u=-\frac{w\varphi}{2}}^{\frac{w\varphi}{2}} \sum_{v=-\frac{w\varphi}{2}}^{\frac{w\varphi}{2}} G(u, v) \varphi_x(i - uw, j - vw) \quad (3.8)$$

$$\varphi'_y(i, j) = \sum_{u=-\frac{w\varphi}{2}}^{\frac{w\varphi}{2}} \sum_{v=-\frac{w\varphi}{2}}^{\frac{w\varphi}{2}} G(u, v) \varphi_y(i - uw, j - vw) \quad (3.9)$$

Where G is a Gaussian low-pass filter of size $w \times w$

5. The final smoothed orientation field O at pixel (i, j) is calculated using Eq. (3.10).

$$O(i, j) = \frac{1}{2} \tan^{-1} \left(\frac{\varphi'_y(i, j)}{\varphi'_x(i, j)} \right) \quad (3.10)$$

3.4 Ridge Frequency Estimation

The frequency image represents the local frequency of the ridges in a fingerprint. In a local neighbourhood where no minutiae and singular points appear, the grey levels along ridges and valleys can be modelled as a sinusoidal-shaped wave along a direction normal to the local ridge orientation. Therefore, local ridge frequency is another intrinsic property of a fingerprint image. Let G be the normalized image and O be the orientation image. The steps involved in local ridge frequency estimation are as follows [20]:

1. We divide G into non overlapping blocks of size $w \times w$.
2. For each block centered at pixel (i, j) , we compute an oriented window of size $l \times w$. The window is defined in the ridge coordinate system along a direction orthogonal to the local ridge orientation.
3. We calculate the average number of pixel between two consecutive peaks. Say, $T(i, j)$ be average number of pixel of block centered at (i, j) .
4. Frequency the block (i, j) block can be calculated using Eq. (3.11):

$$F(i, j) = \frac{1}{T(i, j)} \quad (3.11)$$

5. If no consecutive peaks can be detected from the x-signature, then the frequency is assigned a value of -1 to differentiate it from the valid frequency values.

For a fingerprint image scanned at a fixed resolution, the value of the frequency of the ridges and valleys in a local neighbourhood lies in a certain range. For a 500dpi image, this range is $[1/3, 1/25]$. Therefore, if the estimated value of the frequency is out of this range, then the frequency is assigned a value of -1 to indicate that a valid frequency cannot be obtained.

3.5 Ridge Filtering

A two dimensional Gabor filter consists of a sinusoidal plane wave of a particular orientation and frequency, modulated by a Gaussian envelope. Once the ridge orientation and ridge frequency information has been determined, these parameters are used to construct the even-symmetric Gabor filter.

Gabor filters are employed because they have frequency-selective and orientation selective properties. These properties allow the filter to be tuned to give maximal response to ridges at a specific orientation and frequency in the fingerprint image. Therefore, a properly tuned Gabor filter can be used to effectively preserve the ridge structures while reducing noise.

The even-symmetric Gabor filter is the real part of the Gabor function, which is given by a cosine wave modulated by a Gaussian (see Figure 3-3). An even symmetric Gabor filter in the spatial domain is defined is Eq. (3.12) [22].

$$G(x, y : \theta, f) = \exp \left\{ -\frac{1}{2} \left[\frac{x_\theta^2}{\sigma_x} + \frac{y_\theta^2}{\sigma_y} \right] \right\} \cos(2\pi f x_\theta) \quad (3.12)$$

Where,

$$x_\theta = x \cos \theta + y \sin \theta \quad (3.13)$$

$$y_\theta = -x \sin \theta + y \cos \theta \quad (3.14)$$

Where θ is the orientation of the Gabor filter, f is the frequency of the cosine wave, σ_x and σ_y are the standard deviations of the Gaussian envelope along the x and y axes, respectively. x_θ and y_θ define the x and y axes of the filter coordinate frame, respectively.

The Gabor filter is applied to the fingerprint image by spatially convolving the image with the filter. The convolution of a pixel (i, j) in the image requires the corresponding orientation value $O(i, j)$ and ridge frequency value $F(i, j)$ of that pixel. Hence, we apply the Gabor filter G to obtain the enhanced image E is performed using Eq. (3.15):

$$E(i, j) = \sum_{u=\frac{-wx}{2}}^{\frac{wx}{2}} \sum_{v=\frac{-wy}{2}}^{\frac{wy}{2}} G(u, v; O(i, j), F(i, j))N(i - u, j - v) \quad (3.15)$$

Where O is the orientation image, F is the ridge frequency image, N is the normalised fingerprint image, w_x and w_y are the width and height of the Gabor filter mask, respectively. Figure 3-4 shows the result of applying Gabor filter on fingerprint image with different values of σ_x and σ_y

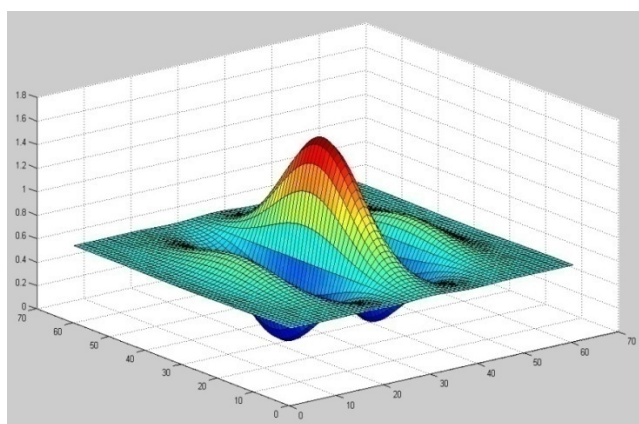


Figure 3-3: Even symmetric Gabor filter in spatial domain



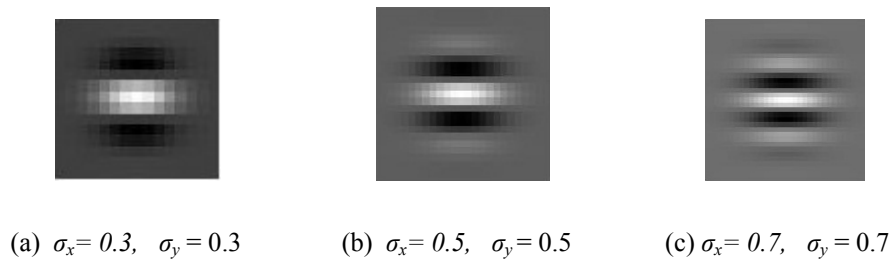


Figure 3-4: Gabor filtered fingerprint image with different parameter value of σ_x and σ_y

3.4 Binarization and Thinning

Most minutiae extraction algorithms operate on binary images where there are only two levels of interest: the black pixels that represent ridges, and the white pixels that represent valleys. Binarisation is the process that converts a grey level image into a binary image. This improves the contrast between the ridges and valleys in a fingerprint image, and consequently facilitates the extraction of minutiae. One useful property of the Gabor filter is that it has a DC component of zero, which means the resulting filtered image has a mean pixel value of zero. Hence, straightforward binarisation of the image can be performed using a global threshold of zero. The binarisation process involves examining the grey-level value of each pixel in the enhanced image, and if the value is greater than the global threshold, then the pixel value is set to a binary value one; otherwise, it is set to zero. The outcome is a binary image containing two levels of information, the foreground ridges and the background valleys.

The final image enhancement step typically performed prior to minutiae extraction is thinning. Thinning is a morphological operation

that successively erodes away the foreground pixels until they are one pixel wide. A standard thinning algorithm [23] is employed, which performs the thinning operation using two sub iterations. Each subiteration begins by examining the neighbourhood of each pixel in the binary image, and based on a particular set of pixel-deletion criteria, it checks whether the pixel can be deleted or not. These sub iterations continue until no more pixels can be deleted. The application of the thinning algorithm to a fingerprint image preserves the connectivity of the ridge structures while forming a skeletonised version of the binary image. This skeleton image is then used in the subsequent extraction of minutiae. Figure 3-5 (a) is result of binarization and Figure 3-5 (b) shows image after thinning process of the image.



(a) Image after binarization (b) Image after thinning

Figure 3-5: Binarization and thinning of fingerprint image

3.5 Conclusion

Fingerprint preprocessing is required to extract the feature form minutiae. This makes task of extracting feature form fingerprint easy and ensure the quality of extracted feature. To enhance the quality of ridge valley structure we follow the different steps. First step is normalization of fingerprint image. Normalization is used to standardize the intensity values in an image by adjusting the range of grey-level values so that it lies within a desired range of values. In next step, we do segmentation. Segmentation is the process of separating the foreground regions in the image from the background regions. After segmentation, we apply even symmetric Gabor filter to enhance ridge valley structure. Finally, we apply thinning operation to get one pixel wide ridge, which can be use in feature extraction process.

Chapter 4

Feature Extraction

Feature extraction is essential part of any biometric system. After enhancing the fingerprint image ridge-valley structure is become clearer and it eases the feature extraction process. In this chapter, we discuss the methods of extracting minutiae feature. Section 4.1 describes the method for minutiae feature extraction. Approach for removing false minutiae is discussed in Section 4.2. Section 4.3 gives the details of convention and technique of minutiae orientation determination. Finally, Section 4.4 summarizes this chapter.

4.1 Minutiae Feature Extraction

For minutiae extraction Crossing Number (CN) [5] method is used. This method use block operation on thinned fingerprint image. This method is extracts the ridge end and bifurcation on the basis of local neighbourhood pixel.

After the CN for a ridge pixel has been computed, the pixel is then classified according to the property of its CN value. As shown in Figure 4-

1(a), a ridge pixel with a CN of one corresponds to a ridge ending, and a CN of three corresponds to a bifurcation as show in Figure 4-1(b). For each extracted minutiae point, the following information is recorded:

- i) coordinate of minutiae (x, y)
- ii) type of minutiae (ridge ending or bifurcation)

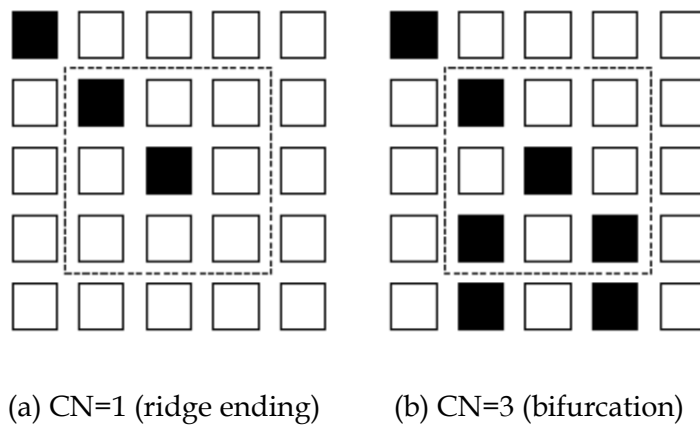


Figure 4-1: Crossing number for ridge ending and bifurcation [5]

4.2 Post-processing

Extracted minutiae from above method may have many false minutiae. In order to remove all those false minutiae, we use method of Euclidian distance. In this method, we consider some minimum distance which should be between to minutiae say it is d . We calculate distance between all ridge-ending to ridge-ending, bifurcation to bifurcation and ridge-ending to bifurcation. If Euclidian distance between two minutiae less then d , we consider them as false minutiae and hence it is dropped out.

Euclidian Distance (ED) is define between two minutiae pixel position using Eq. (4.1) and false minutiae can be determine by rule given in Eq. (4.2)

$$ED = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (4.1)$$

$$false\ minutiae = \begin{cases} ED(termination, bifurcation) < d \\ ED(termination, termination) < d \\ ED(bifurcation, bifurcation) < d \end{cases} \quad (4.2)$$

4.3 Orientation Estimation of Minutiae

To define orientation of each minutia, we use 7×7 mask technique with angles quantized to 15° and with the centre placed in a minutiae point. The orientation of an ending point is equal to the point where a ridge is crossing through the mask. The orientation of a bifurcation point can be estimated with the same method but only the leading ridge is considered, which is the ridge with maximum sum of angles to other two ridges of the bifurcation as shown in Figure 4-2.

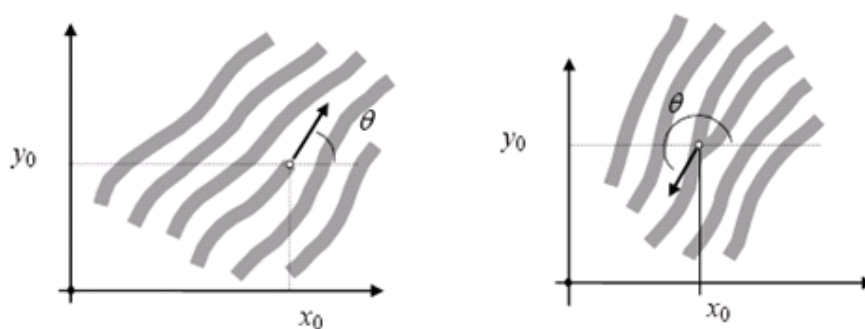
We follow a convention while determining orientation of minutiae, to cooperate the interoperability of matching algorithm which uses minutiae data from different source.

Figure 4-3 shows the convention for determining the angle of minutiae. Figure 4-3(a) shows ridge ending minutia. x_0, y_0 are the minutia coordinates and θ is the angle that the minutia tangent forms with the horizontal axis. Figure 4-3(b) shows a bifurcation minutia. θ is now

defined by means of the ridge ending minutia corresponding to the original bifurcation that exists in the negative image.

35	20	05	0	5	0	5
150						0
65						5
80						
95						45
10						30
25	40	55	70	85	00	15

Figure 4-2: Orientation of minutiae using 7×7 mask (in this case is 225 degree)



(a) Angle convention of ridge ending (b) Angle convention for bifurcation

Figure 4-3: Convention for determining angle of minutiae [5]

This convention is necessary as in inverted image minutia changes their type i.e. minutiae termination will become bifurcation and vice-versa. This is called ridge ending/bifurcation duality. This phenomenon is illustrated in Figure 4.4. Figure 4.4(b) is an inverted image of Figure 4.4(a). In Figure 4.4(a), there are two ending and one bifurcation, and in its inverted image in Figure 4.4(b), there are two bifurcations and one ending on same position.

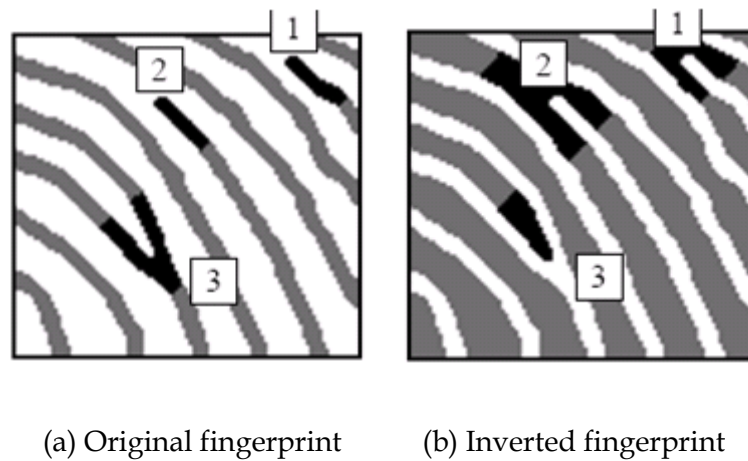


Figure 4-4: Ridge ending/bifurcation duality [5]

4.4 Conclusion

Feature extraction is essential part of any fingerprint based biometric system. After enhancing the fingerprint image ridge-valley structure is become clearer and it eases the feature extraction process. Minutiae features is most commonly used fingerprint feature for identification process. For extracting minutiae from thinned fingerprint crossing number

method is used. This method use block operation on thinned fingerprint image. In order to remove false minutiae, method of Euclidian distance is used. To define orientation of each minutia we have use 7×7 mask technique with angles quantized to 15° and with the center placed in a minutiae point. To determining the orientation of minutiae we follow a convention to overcome the ridge-valley duality phenomena.

Chapter 5

Fingerprint Indexing

This Chapter gives the details of proposed approach used for fingerprint indexing. Section 5.1 describes the general framework of indexing system with three different approaches. Some basic preliminary of k -means clustering algorithm and kd -tree is described in Section 5.2. In Section 5.3, we introduce a new technique for formation for minutiae triplet two closest points triangulation. The feature vector which is used for creation of index space is described in Section 5.3. Section 5.4 tells us about approach used for creation of population of index space. Finally, in Section 5.6, we see the different retrieval strategy for querying and approach used to generation of candidate set.

5.1 Framework of the Proposed Approach

The main focus of proposed technique relies on the creation of an 8-dimensional index space model based on minutiae triplets and Gabor energy associated with the minutiae. First, we extract the index vectors from all fingerprints available in database. This set of all index vectors is called index space of database. Then we evaluate the indexing system with

three different configurations; linear search, clustered search and *kd*-tree search followed by clustering. These configurations are shown in Figure 5-1, Figure 5-2 and Figure 5-3.

First approach is linear search approach. In this approach, minutiae are converted to the set of triplets. We extract 8-dimension feature vector from each minutiae triplet. Feature vector is extracted from each triplet of a fingerprint. The whole process runs on entire fingerprint database. This gives us set of all feature vector from all fingerprint in database. Each feature vector plays role of index key. Identity of person is associated with each feature vector extracted from his/her fingerprint. This whole process is called creation of index space. When query fingerprint has to match with the enrolled fingerprint in database, we extract the feature vector from query print and exhaustively find the closest match in index space. Each closet match of feature vectors of query print gives person's identity associated with it. We list top N most common identity of person as a candidate set. Here N depends on the desired penetration rate of indexing system. Steps of linear search approach are depicted in Figure 5-1.

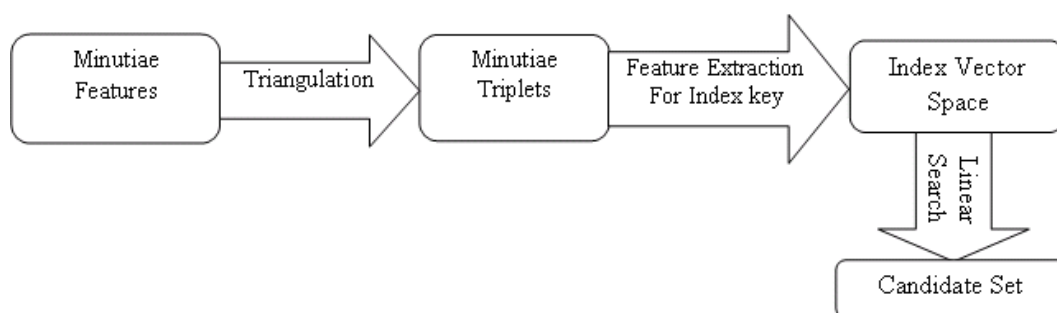


Figure 5-1: Linear search approach

Our next approach is clustered search approach. This approach is extension of linear search approach. After creating index space for a database, we apply k -means clustering to cluster the index space. When query fingerprint has to match with the fingerprint in enrolled database, we extract the feature vector from the query print. For each feature vector, first, we identify the cluster to which it belongs. Then we search for closest match in identified cluster. Candidate set will be top N most common identities associated with matched feature vector as in linear search approach. Steps of linear search approach are shown in Figure 5-2.

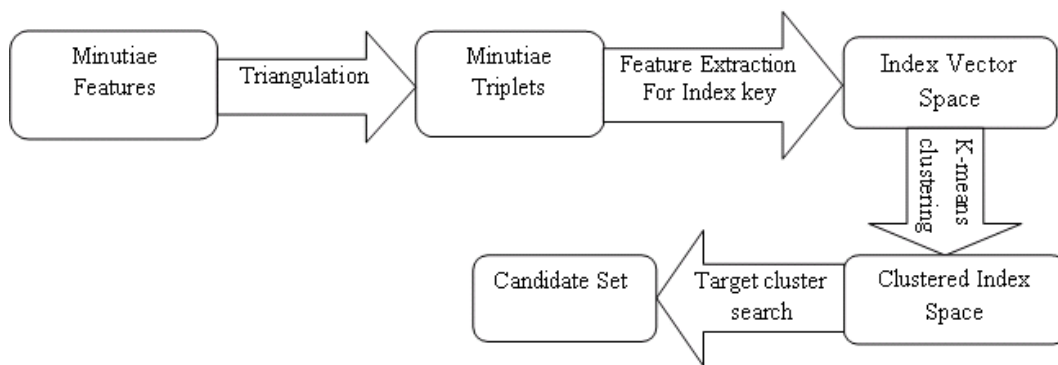


Figure 5-2: Clustered search approach

Last approach is clustered kd -tree index search approach. Again this approach is slight extension of clustered search approach. In this approach, after clustering the index space into number of cluster, each feature vector is indexed using kd -tree. In this case, we apply nearest neighbour search on kd -tree of identified cluster. Candidate set generation is same as cluster search approach approaches. Steps of clustered kd -tree index search are shown in Figure 5-3.

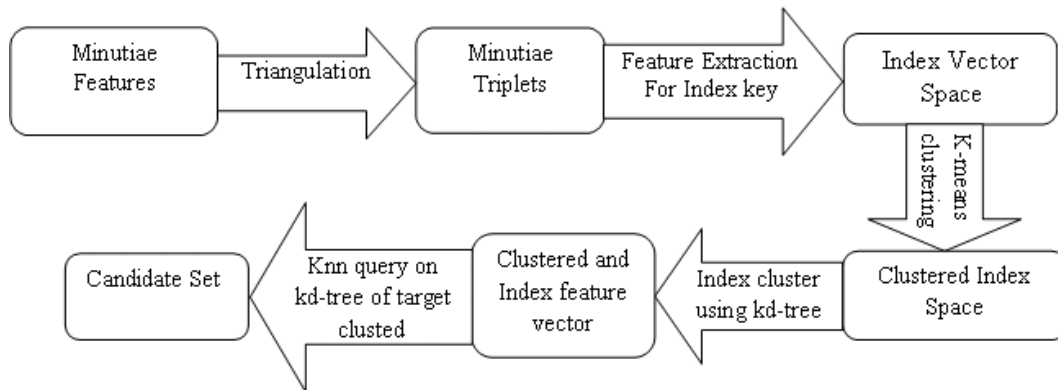


Figure 5-3: Clustered *kd*-tree index search

5.2 Preliminaries

5.2.1 *k*-means data clustering

Clustering is a process of partitioning the data into subsets or clusters, so that data in each subset share certain properties. Data clustering is a common statistical analysis tool used in different fields like image analysis, pattern recognition and machine learning. A classification or clustering technique can be supervised or unsupervised. In supervised clustering, class label of each training tuple is provided (i. e. learning of the classifier is “supervised” in that it is told to which class each training tuple belongs). While in unsupervised learning, which the class label of each training tuple is belong, not known and the number or set of classes to be learned may not be known in advance.

k -means algorithm is an unsupervised clustering algorithm to cluster objects into K partitions based on their attributes. Given N data point and K is number of cluster. k -means partition the data point into K cluster such that data points in same cluster have nearest mean which can be viewed as cluster's centroid. The goal is to define K centroids, one for each cluster. These centroids are the representative of their own cluster.

k -means algorithm follows simple step to classify a given dataset in cluster. Following are the basic step of k -means clustering algorithm for k number of data cluster

Formally, for a given dataset $X = \{x_1, x_2, x_3, \dots, x_n\}$, and k ($k \leq n$) number of cluster, k -means algorithm partition the dataset into k cluster $C_1, C_2, C_3, \dots, C_k$ with centroids $c_1, c_2, c_3, \dots, c_k$, so as to minimize the within-cluster sum of squares (WCSS).

$$\text{agr min } \sum_{i=1}^k \sum_{x_j \in C_i} \|x_i - c_i\|^2 \quad (5.1)$$

Algorithm 5.1: k -Means

- 1: Initialize k : number of clusters
 - 2: Initialize D : a data set containing n objects
 - 3: Randomly select k initial object from the data set D as cluster centers
 - 4: Assign each object to clusters to which the object is most similar based on mean value of the object in the cluster
 - 5: Update the cluster means by mean value of the object for each cluster
 - 6: Repeat step 4 to 5 until no change
-

k -means most popular clustering algorithm because it's quick convergence. This as example of heuristic algorithm, so there is no

guarantee that it will converge to the global optimum, and the result is depend on the initial randomly selected cluster points. Another problem with k -means is that we have to predefine the number of cluster. This might lead to undesirable result for data sets which are not naturally clustered. There are some statistical approaches to determine the number of cluster such as Elbow Method, information criterion approach. One simple thumb of rule to determine number of cluster is square root of half of the data point ($\sqrt{n/2}$) [24]. We use this rule to determine the number of cluster in our experiments.

5.2.2 kd -Tree

kd -tree is stands for k -dimensional tree. It is a space-partitioning data structure for organizing points in a k -dimensional space. The kd -tree is similar to a binary tree [25]in which every node is a k -dimensional point. Every non-leaf node can be thought of as implicitly generating a splitting hyperplane that divides the space into two parts, known as subspaces. Points to the left of this hyperplane represent the left sub-tree of that node and points right of the hyperplane are represented by the right sub-tree. The hyperplane direction is chosen in the following way: every node in the tree is associated with one of the k -dimensions, with the hyperplane perpendicular to that dimension's axis. So, for example, if for a particular split the "x" axis is chosen, all points in the subtree with a smaller "x" value than the node will appear in the left subtree and all points with larger "x" value will be in the right sub tree. In such a case, the hyperplane would be set by the x-value of the point, and its normal would be the unit x-axis. Since there are many possible ways to choose axis-aligned splitting planes,

there are many different ways to construct *kd*-trees. The canonical method of *kd*-tree construction has the following constraints:

- As one moves down the tree, one cycles through the axes used to select the splitting planes. (For example, the root would have an *x*-aligned plane, the root's children would both have *y*-aligned planes, the root's grandchildren would all have *z*-aligned planes, the next level would have an *x*-aligned plane, and so on.)
- Points are inserted by selecting the median of the points being put into the subtree, with respect to their coordinates in the axis being used to create the splitting plane. (Note the assumption that we feed the entire set of points into the algorithm up-front.)

This method leads to a balanced *kd*-tree, in which each leaf node is about the same distance from the root. Figure 5-5 shows an example of two dimensional *kd*-tree. *kd*-tree supports for nearest neighbour search and range query search. *kd*-tree implementation is used provided by [26].

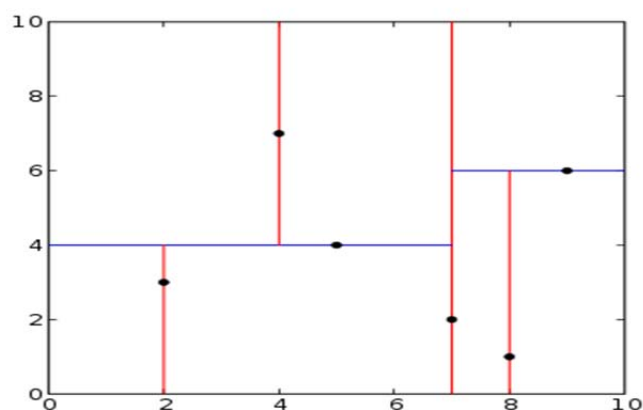


Figure 5-4: Two dimensional *kd*-tree

5.3 Two Closest Points Triangulation

Two closest points triangulation is a proposed technique to produce more meaningful geometric area of fingerprint for generation of index key. The algorithm use simple approach to find two closed point in two-dimensional space for a given point. These three points will become point of triangle. In case of fingerprint we use location of minutiae in two-dimensional space. Correspond to each minutiae point we find two closest minutiae. These three minutiae point will become points of a triangle. Two closest point triangle can be generate using Algorithm 5.2.

Algorithm 5.2: Two Closest Point Triangulation

- 1: Initialize $M = \{m_1, m_2, \dots, m_n\}$ //minutiae set
 - 2: For each m_i in M repeat step 3 to 5
 - 3: Find two closest minutiae point m_j, m_k in two dimension space using Euclidean distance
 - 4: Assign m_i, m_j, m_k to minutiae triplet t_i
 - 5: Add minutiae triplet t_i to triplet set T
 - 6: Eliminate duplicate entry in T
-

For given minutiae points N , two closest points triangulation gives T number of triangles then we can observe that in each case $T \leq N$ because of some repeated triangles. Delaunay triangulation for generation minutiae triplet is used in [12] [14] [27]. In case of using Delaunay triangulation, the

order of number of triangle is order of $9N+1$ [28]. Hence, two closes point triangulation is form less but more meaningful triangle. Less number of triangle lead less number of index key per fingerprint hence less computation and storage overhead is required. Figure 5-7 shows sample of two closest point triangulation of minutiae points of a fingerprint.

5.4 Feature Extraction for Index Vector

The main focus of selection of feature for index vector in to have property of invariance to rotation and scaling. Choosing such feature can eliminate extra effort of alignment and registration of image. Also it enables the database searchable for fingerprint from different sources.

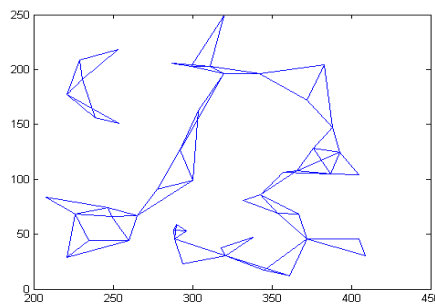


Figure 5-5: Two closest point triangulation

In our method, we use 8-dimension index vector for index key generation. The features are extracted from minutiae triplet generated by two closest points triangulation. In 8-dimension feature vector, first two dimensions give the geometric information of triplet. Next three dimensions add angle of each minutia associated with triplet, and final three dimensions gives information about local ridge configuration

around each minutiae point of triplet. Figure 5-6 gives idea of creation of 8-dimension feature.

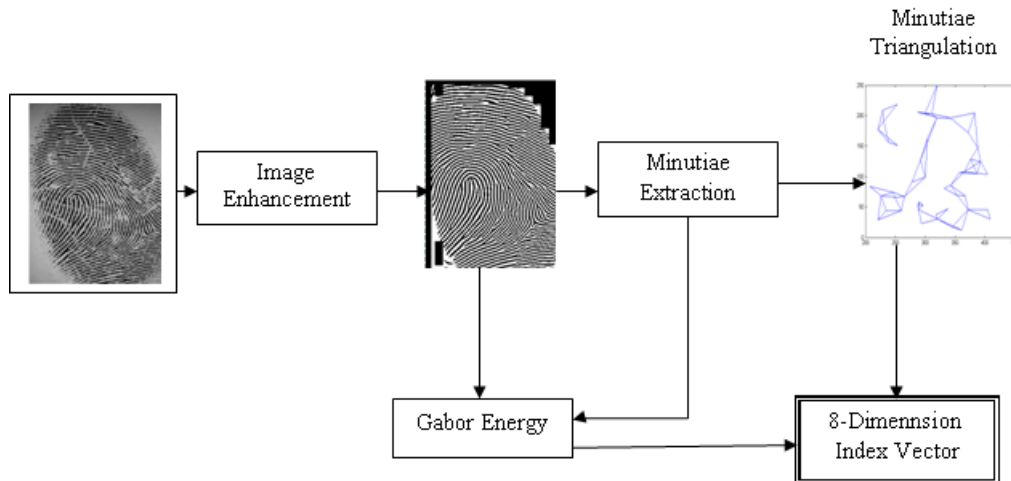


Figure 5-6: Process of feature extraction for index vector

Given a fingerprint image with minutiae set $M = (m_1, m_1, \dots, m_n)$, $m_i = (x_i, y_i, \theta_i)$, where (x_i, y_i) is the location of minutia m_i and θ_i is orientation of minutiae m_i . The process of triangulation generates set of triplet $T = (t_1, t_2, \dots, t_k)$, where $1 \leq k \leq n$. the index vectors are correspond to these set of triplet. Each triplet t_i contain three minutiae point $t_i = (m_1, m_2, m_3)$. These three point become point of triangle in two-dimensional space.

For a triplet t_i which consist minutiae m_1, m_2, m_3 , let l_1, l_2, l_3 are the length of side of triangle such that $l_1 \leq l_2 \leq l_3$. And $\theta_1, \theta_2, \theta_3$ are angle of minutiae m_1, m_2, m_3 . Let θ_m is the orientation of longest side of triangle and $\alpha_1, \alpha_2, \alpha_3$ are the interior angle of the triangle such that $\alpha_1 \leq \alpha_2 \leq \alpha_3$. Figure 5-7 shows the different parameter of triangle generated by minutiae triplet.

The different 8-dimensions ($d_1, d_2 \dots d_8$) are computed as follows.

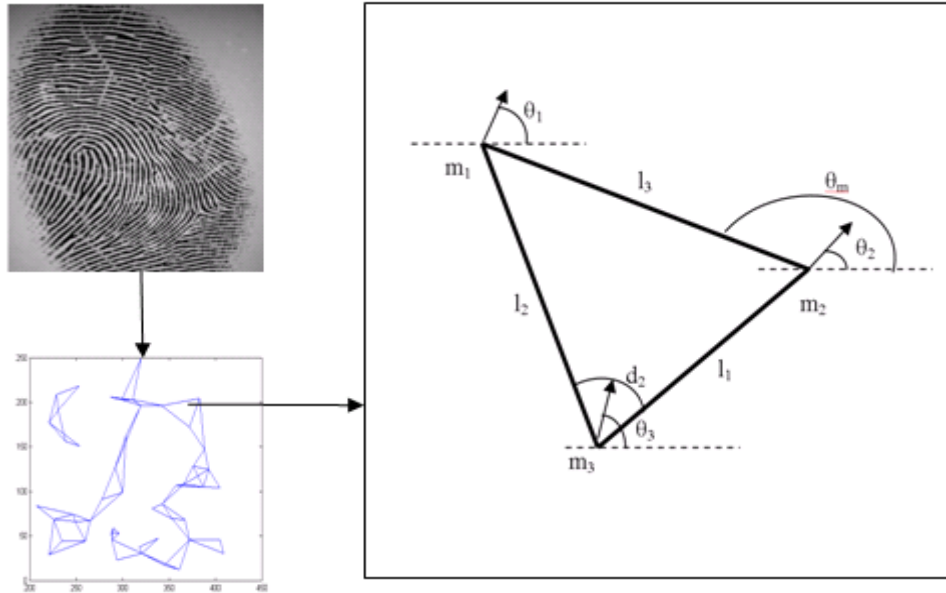


Figure 5-7: Component of minutiae triplet

d_1 : First dimension d_1 is defined as the ratio of two smaller sides of triangle. It is calculated using Eq. (5.2)

$$d_1 = \frac{l_2}{l_1} \quad (5.2)$$

d_2 : Second dimension d_2 of index vector is defined as interior angle between two smaller sides of triangle. In this case angle between l_1 and l_2

d_3, d_4 and d_5 : These three dimension are the orientation of minutiae m_1, m_2, m_3 with respect to orientation of longest side of triangle θ_m . d_3, d_4 and d_5 is calculated using Eq. (5.3), (5.4) and (5.5), respectively

$$d_3 = abs(\theta_m - \theta_1) \quad (5.3)$$

$$d_4 = abs(\theta_m - \theta_2) \quad (5.4)$$

$$d_5 = abs(\theta_m - \theta_3) \quad (5.5)$$

d₆, d₇ and d₈ : these three dimension *d₆, d₇, d₈* are the Gabor energy feature with the location orientation of minutiae *m₁, m₂, m₃* respectively. Gabor energy have been used in many [22] [29] [30] biometric indexing implementations. In these implementations they use Gabor filter with different orientation and scale. But in our approach we use single filter for each minutia with the orientation same as orientation of minutiae θ_m . Using orientation of minutiae makes this feature invariable to rotation of fingerprint image. We apply Gabor filtering on 20×20 block size of enhanced image. This Gabor energy feature gives numerical summery of local ridge configuration around the minutiae.

Let $I(x, y)$ be the enhanced fingerprint image and $G(f, \theta)$ be Gabor filter as discussed in Section 3.5 with frequency f and orientation θ . Gabor response GR can be obtain by 2-D convolution of image $I(x, y)$ and Gabor filter $G(f, \theta)$. It is computed using Eq. (5.6). Here frequency f is mean frequency of local ridge frequency of the finger print. It is computed using Eq. (3.11) in Section 3.5. w is width and height of block of image. In our case w is 20 and θ is angle of minutia.

$$GR(i, j) = \sum_{u=-\frac{w}{2}}^{\frac{w}{2}} \sum_{v=-\frac{w}{2}}^{\frac{w}{2}} G(f, \theta)I(i - u, j - v) \quad (5.6)$$

These responses are called Gabor coefficients values. These are complex value. We calculate the magnitude value of the responded image at each pixel. The Gabor energy is calculated by summing up the square value of the magnitude of Gabor responses at each pixel. Gabor energy GE_θ is calculated using Eq. (5.7).

$$GE_\theta = \sum_{x,y} [|GR(x,y)|]^2 \quad (5.7)$$

5.5 Creation of Population of index Space

Each fingerprint image can be represented as a set of triangles made by two closest point triangle $T = \{t_1, t_2, \dots, t_n\}$ that is generated from its minutiae distribution. Each triplet is characterized by 8-Dimension feature vector as discussed in Section 5-4. This 8-dimensional entity can be viewed as a single point in hyperspace. Thus each fingerprint image will have a collection of points correspond to each triplet, residing in this 8-dimensional space. We collect all feature vectors from all fingerprints in database. This process forms index space of database. Then we apply k -means clustering with number of clusters $\sqrt{n/2}$. Where n is total number of index vector in database. Using $\sqrt{n/2}$ number of cluster in thumb rule to determine the number or cluster for any unsupervised clustering algorithm [24]. This is an offline process of creation of index space. Further, for an online fingerprint identification system, first we decompose into triplet and then map to set of 8-dimentional feature vectors to enroll new fingerprint in database. Each Index vector

correspond to each triplet is assign to exactly one cluster based of minimum distance rule. Figure 5-8 summarise the process of creation of index space for clustered search approach.

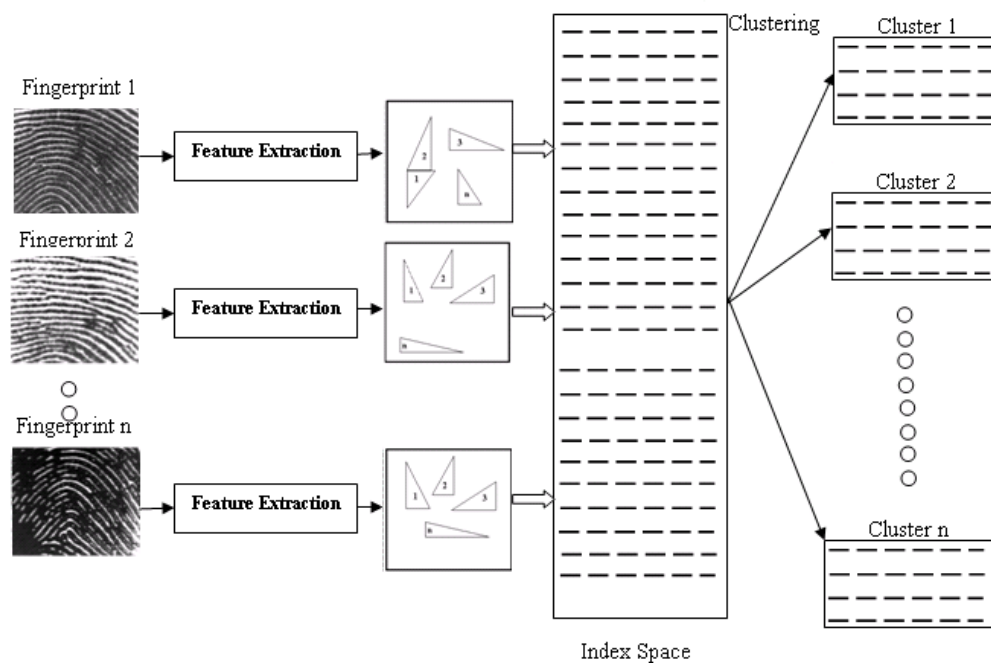


Figure 5-8: Creation of index space population

5.6 Querying

Querying is the process of creation of candidate set of enrolled fingerprints corresponds to a query fingerprint, which are most likely to match with the query fingerprint. We adopt different retrieval strategies for querying process.

Retrieval strategies

After determining the class to which a query fingerprint belong a good retrieval strategy is help full to improve the accuracy of classification or

indexing system. A retrieval strategy is an important consideration for an indexing or a classification scheme. For any given classification or indexing technique, the following retrieval strategies can be used as suggested in [5].

- 1. Search target class only:** In this strategy, only fingerprints belonging to the target class are searched. Target class is defined as the class to which the query fingerprint has been assigned. The search is stopped when a matching fingerprint is found or the entire target class is visited.
- 2. Search according to predefined search order:** If the query fingerprint does not match with the target class the entire database is search in predefined order. Search is stopped when query fingerprint found or entire database visited.
- 3. Search according to variable search order:** In this strategy, we have predefine preference of class based on score of similarity or degree of likelihood with the class. If the fingerprint does not match with the first preferred class it has to search all other class in sequence till match is found.
- 4. Fixed radius search:** In this strategy, only those fingerprints whose corresponding vectors are within a predefined radius from the vector corresponding to the query fingerprint are searched. This search strategy is relevant to the continuous classification scheme for indexing [5]. The Potential candidates for matching lie within the hyper-sphere whose center is defined by the query fingerprint vector. This strategy significantly reduces the search space. The

search is stopped either match found or part of database is visited within the predefined radius.

- 5. Incremental search:** The search space is expanded in small increments if a matching fingerprint is not found within the radius specified initially. The search is carried on until query fingerprint found, some threshold radius reach or entire database is visited.

In our approach, we use search target cluster only approach. Figure 5-9 shows the steps of retrieving candidate set form clustered index space in clustered search approach. First we extract minutiae feature, these minutiae are converted into set of triplet using two closest point triangle. Each triplet is then map to 8-dimension index vector. Next we indentify to which cluster an index vector belongs. Then we search for the closest match in identified cluster. We apply this to all index vector of query print. Each closest match in the cluster add vote to its corresponding id. Based on votes we list the top N candidates for the query print. Where N is depend on the desired penetration rate of indexing system. If the query print is not found in listed candidate set we increase the size of candidate set. By which we can add some probability of finding the match in candidate set. Figure 5-10 shows the query process in linear search approach and Figure 5-11 shows the query process in clustered search approach.

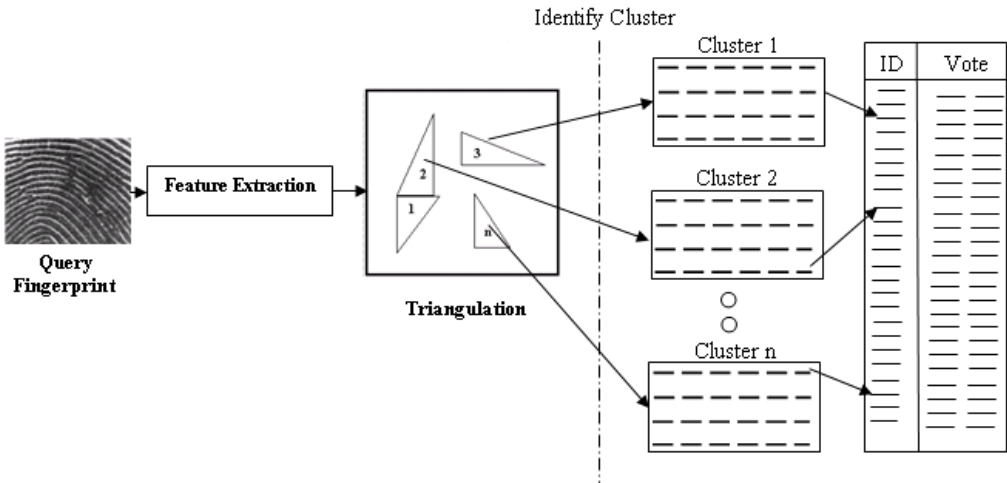


Figure 5-9: Querying fingerprint on cluster search approach

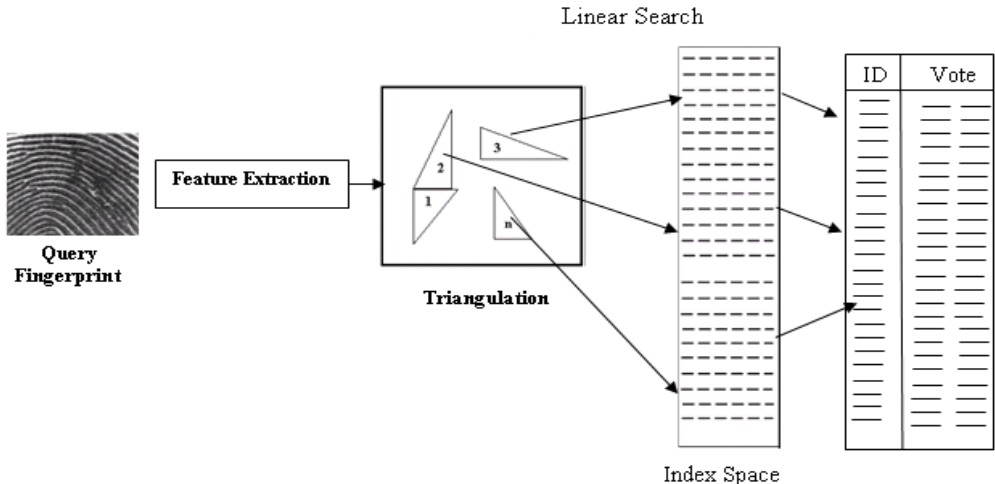


Figure 5-10: Querying fingerprint on linear search approach

Chapter 6

Experimental Results

In this chapter, we describe detail experimental results to show the performance of the proposed approach. In Section 6.1, we give detail of database we used. Implementation environment is described in Section 6.2. In Section 6.3, we define the evaluation parameters of experiments. Section 6.4 gives experimental result related to image enhancement. The result of minutiae feature extraction is given in Section 6.5. Section 6.6 gives detail experimental result of proposed approach. Analysis of cluster population is given in Section 6.7. Comparison of approaches described in Section 5.1 is given in Section 6.8. Section 6.9 presents comparison with existing work. Finally, chapter is summarized in Section 6.10.

6.1 Fingerprint Database

Fingerprint Verification Contest (FVC) 2004 Database: This database contain four set of 800 image of 100 different individual. Each individual have 8 fingerprint impressions. Four set of database in made from

different sensors and of different resolution. Table 6-1 gives detail of FVC 2004 database [31].

Database	Sensor	Image Size	Resolution	
1	DB	Optical Sensor	640×480(307 Kpixel)	500 dpi
2	DB	Optical Sensor	328×364(119 Kpixel)	500 dpi
3	DB	Thermal sweeping Sensor	300×480(144 Kpixel)	512 dpi
4	DB	Synthetic	288×384(108 Kpixel)	About 500 dpi

Table 6-1: FVC-2004 fingerprint database set -A

6.2 Implementation Environment

All the methods and algorithms described in our approach were implemented using MATLAB V7.90 on the windows operating system. Methods are evaluated with Intel core-2 Duo processor of speed 2.53 GHz and main memory of 2 GB.

6.3 Evaluation Metrics

The performance of the proposed indexing mechanism is evaluated using two metrics, namely, the *penetration rate* and the *hit rate*. The hit rate is defined as the probability that the correct user identity is retrieved. Hit rate is computed using Eq. (6.1). The *penetration rate* defines the fraction of user identities retrieved from the database upon presentation of the query print. Penetration rate is computed using Eq. (6.2).

$$hit\ rate = \frac{number\ of\ correctly\ identify\ fingerprint \times 100}{total\ number\ of\ fingerprint} \quad (6.1)$$

$$penetration\ rate = \frac{number\ of\ access\ fingerprint \times 100}{total\ number\ of\ fingerprint\ in\ database} \quad (6.2)$$

One there parameter *average penetration rate* is user to give broad idea about performance of indexing system. In this scenario, average penetration rate is define as mean of penetration rates correspond hit rate range from 1 to 100 percent. Average penetration rate is calculated using Eq. (6.3).

$$average\ penetration\ rate = \frac{\sum_{hit\ rate=1}^{100}(penetration\ rate)}{100} \quad (6.3)$$

6.4 Experimental Result of Image Enhancement

Table 6-2 list the average score original image and corresponding enhanced image. Average score in calculated using Eq. (6.4). The score of a fingerprint is calculated using NFIQ software [32], which rate each fingerprint image in 1-5 scale. Score 1 means (class 1 image) very good quality of image and score 5 means (class 5 image) very bad quality of image. Form table 2.2 we can conclude the fingerprint enhancement process can improve the class of fingerprint image.

$$average\ score = \frac{\sum NFIQ\ score\ of\ fingerprint}{total\ number\ of\ fingerprint\ in\ database} \quad (6.4)$$

	DB1	DB2	DB2	DB4
Original Image	1.8	2.45	2.15	1.83
Enhanced Image	1.4	1.35	1.85	1.42

Table 6-2: Average score of original image and Gabor filtered image using NFIQ tool

6.5 Experimental Result of Minutiae Extraction

Table 6-3 summarizes the result of minutiae extracted from different database. We extract two main minutiae feature ridge ending and ridge bifurcation. Here, we list average number of minutiae extracted from FVC2004 database using our approach of minutiae extraction.

	DB1	DB2	DB3	DB4
Average number of ridge ending	39	32	56	33
Average number of bifurcation	31	23	38	33
Total number of minutiae	70	55	94	66

Table 6-3: Average number of minutiae extracted from different database

6.6 Experimental Evaluation of Indexing

The result of performance of our proposed approach is shown by hit rate and penetration rate as describe in Section 6.3.

In first Section, we see the affect of increasing the number of fingerprint per individual in enrolled database. We divide each set of database (DB1, DB2, DB3, DB4) in to 8 sets, each set contain only one fingerprint impression per individual. We test the performance of indexing system with one, three and five sets to enroll in database. Rest of the set is used for query. (e.g. if we enroll five sets then set 6, 7 and 8 are used for querying)

Experimental Results

Table 6-4, 6-5, 6-6 and 6-7 give detail result of proposed approach clustered search on FVC2004 database, DB1, DB2, DB3 and DB4 respectively. Figure 6-1, 6-2, 6-3 and 6-4 are the plot of experimental result of database DB1, DB2, DB3 and DB4 respectively.

Here, number of fingerprint per individual refers to the number of sample enrolled for indexing. Total number of feature vector denotes the sum of feature vectors extracted from the enrolled fingerprint for indexing. Average time is average time require to generate candidate set from enrolled finger for a query fingerprint, where query fingerprint is in form of minutiae representation template.

No. of fingerprint per individual	Total number of Feature vector	Average time (second)	Hit Rate	Penetration Rate	Average Penetration Rate
1	3404	1.2	100	8.2	3.66
			95	6.3	
			90	6	
3	11163	1.56	100	7.33	3.62
			95	6	
			90	5.7	
5	20035	2.1	100	7	3.6
			95	5.6	
			90	5.4	

Table 6-4: Performance of proposed approach on FVC-2004(DB1) with variable number of fingerprint per individual

Experimental Results

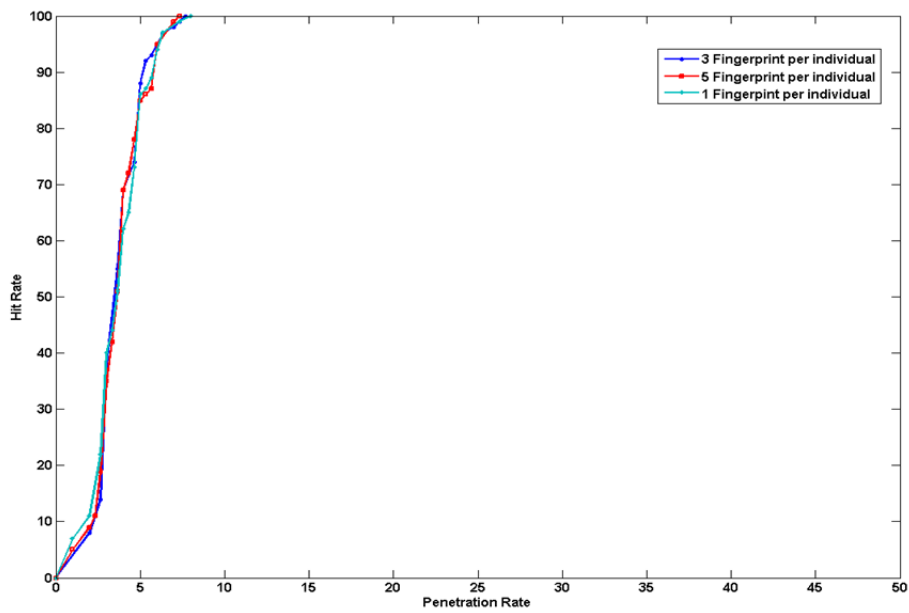


Figure 6-1: Performance of proposed approach on FVC-2004(DB1) with variable number of fingerprint per individual

No. of Fingerprints per Individual	Total number of Feature vectors	Average time (seconds)	Hit Rate	Penetration Rate	Average Penetration Rate
1	3504	1.1	100	9	4.26
			95	8.5	
			90	7.2	
3	11002	1.8	100	8.7	4.1
			95	8.2	
			90	7	
5	20152	2.3	100	8	3.8
			95	7.2	
			90	7	

Table 6-5: Performance of proposed approach on FVC-2004(DB2) with variable number of fingerprint per individual

Experimental Results

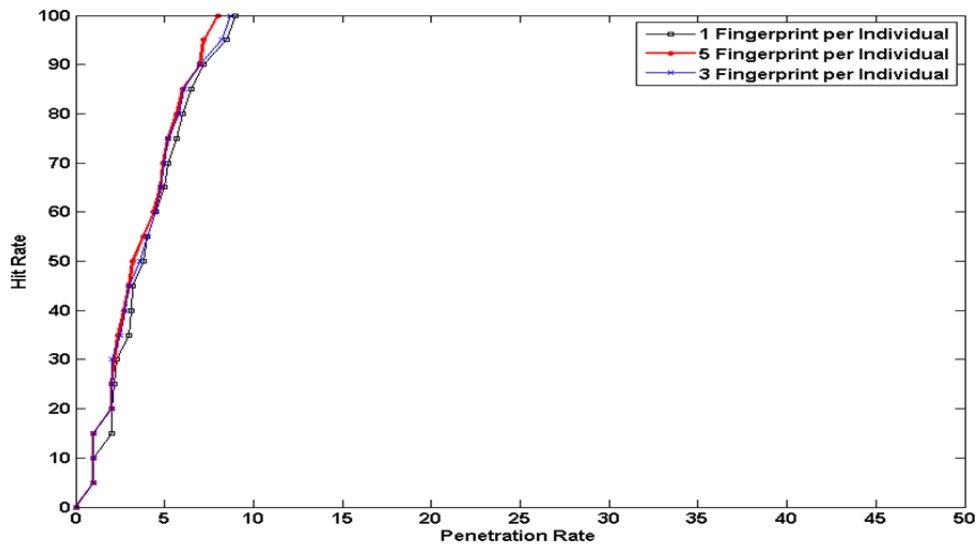


Figure 6-2: Performance of proposed approach on FVC-2004(DB2) with variable number of fingerprint per individual

No. of Fingerprints per Individual	Total number of Feature vectors	Average time (seconds)	Hit Rate	Penetration Rate	Average Penetration Rate
1	7429	1.2	100	7.33	4.1
			95	5.66	
			90	5.32	
3	22364	2.1	100	7	3.63
			95	6.23	
			90	5.62	
5	37201	2.5	100	7	3.5
			95	5.2	
			90	6.66	

Table 6-6: Performance of proposed approach on FVC-2004(DB3) with variable number of fingerprint per individual

Experimental Results

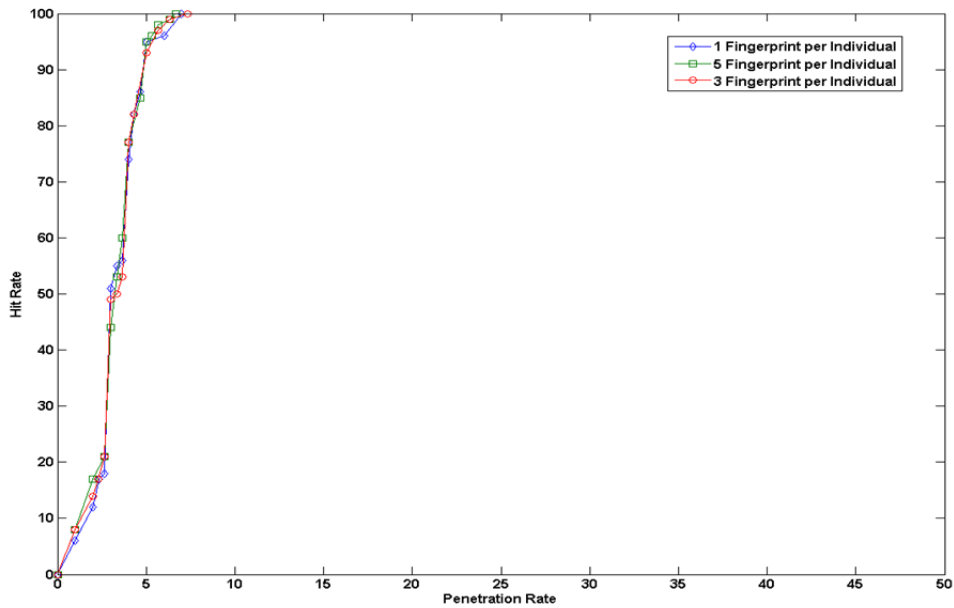


Figure 6-3: Performance of proposed approach on FVC-2004(DB3) with variable number of fingerprint per individual

No. of Fingerprints per Individual	Total number of Feature vectors	Average time (seconds)	Hit Rate	Penetration Rate	Average Penetration Rate
1	3699	1.8	100	8	3.7
			95	6.65	
			90	5.66	
3	10958	2.5	100	7	3.5
			95	6.33	
			90	5.33	
5	18191	3.1	100	7	3.5
			95	5.33	
			90	5	

Table 6-7: Performance of proposed approach on FVC-2004(DB4) with variable number of fingerprint per individual

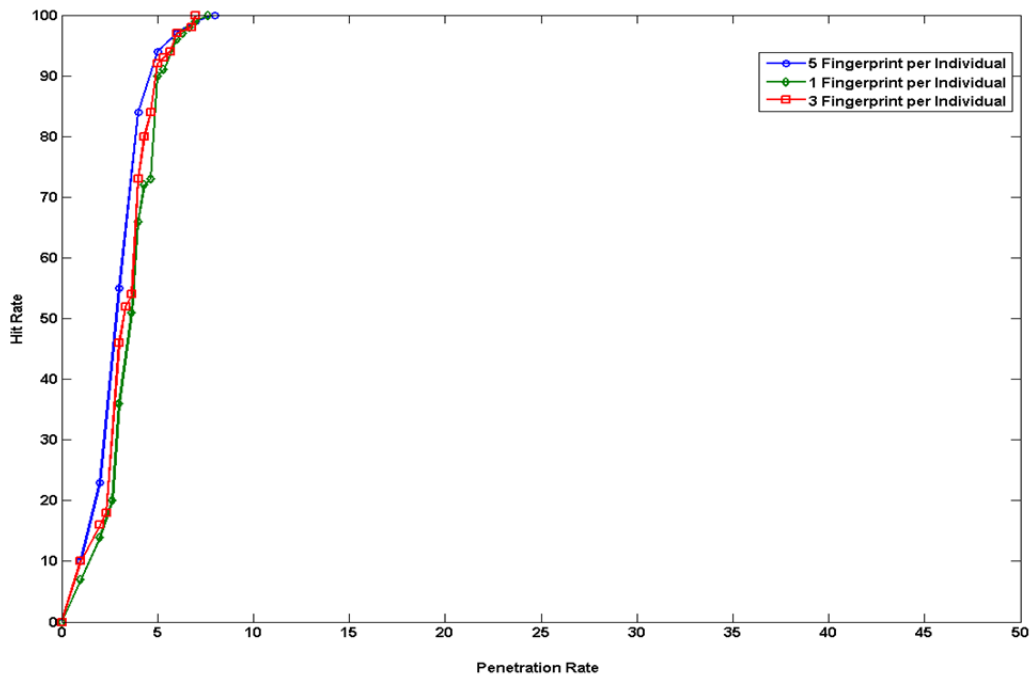


Figure 6-4: Performance of proposed approach on FVC-2004(DB4) with variable number of fingerprint per individual

From above result we can conclude that increasing the number of fingerprint per individual increase the response time but does not much improve the performance in term of hit rate/penetration rate. Or else we can say that it does not require enrolling multiple fingerprints per individual. This also confirms the quality of index key of being invariable to rotation and scaling.

6.7 Analysis of Cluster Population in the Index Space Model

The entities of the fingerprints (triangles) are allotted to clusters. Therefore, each cluster is associated with user identities. Such cluster-user

Experimental Results

association for FVC2004 is shown in the Figure 6-5, 6-6, 6-7 and 6-8. The result shows the balance between number of cluster and user per cluster also it shows the uniform distribution of users in clusters.

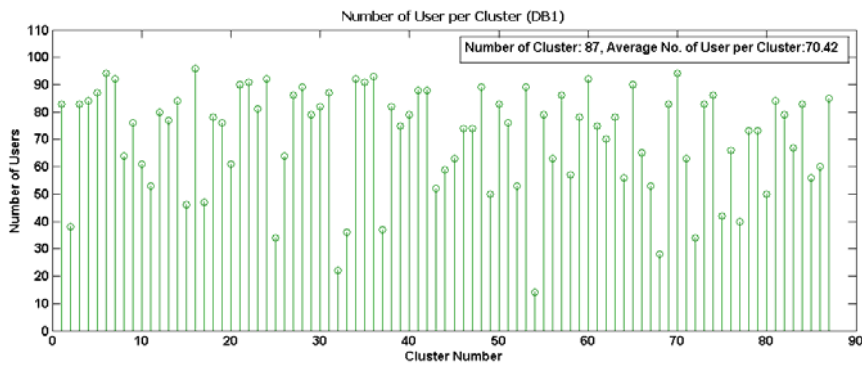


Figure 6-5: User distribution in cluster for FVC2004 DB1

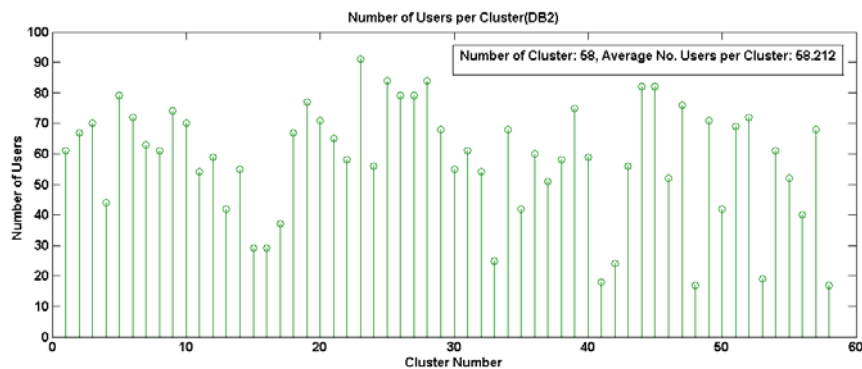


Figure 6-6: User distribution in cluster for FVC2004 DB1

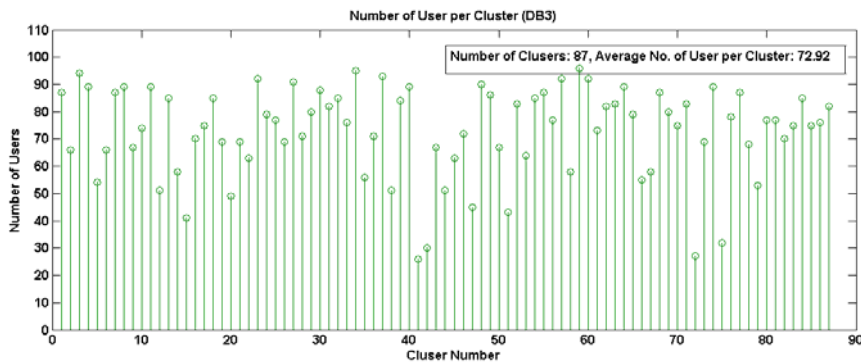


Figure 6-7: User distribution in cluster for FVC2004 DB1

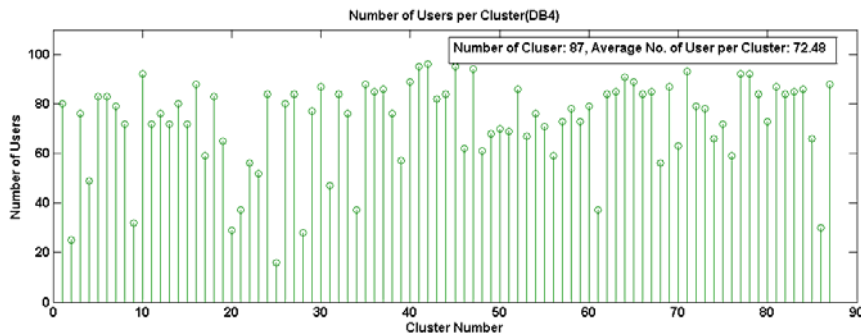


Figure 6-8: User distribution in cluster for FVC2004 DB1

6.8 Comparison with Linear Search, Clustered Search and *kd*-tree Search

Figure 6-9 and Table 6-8 shows the performance three approaches explain in chapter-5 linear search, clustered search and clustered *kd*-tree index search. The given result generated by experiment performed on FVC2004 DB1 with 3 Fingerprint per individual for fingerprint index. Results show that linear search is slower but give have good performance in terms hit rate/penetration rate. Clustered *kd*-tree search is fastest but have high penetration rate. Clustered search approach is gives moderate result in both respect response time and hit rate/penetration rate.

	Linear Search	Clustered Search	Clustered <i>kd</i> -tree index search
Average Time(seconds)	5.6	1.56	0.78
Penetration Rate for Hit Rate 100	3.2	7.33	35
Penetration Rate for Hit Rate 95	3	6	25.22
Penetration Rate for Hit Rate 90	2.2	5.7	22.3

Table 6-8: Comparison with linear search, clustered search and *kd*-tree search

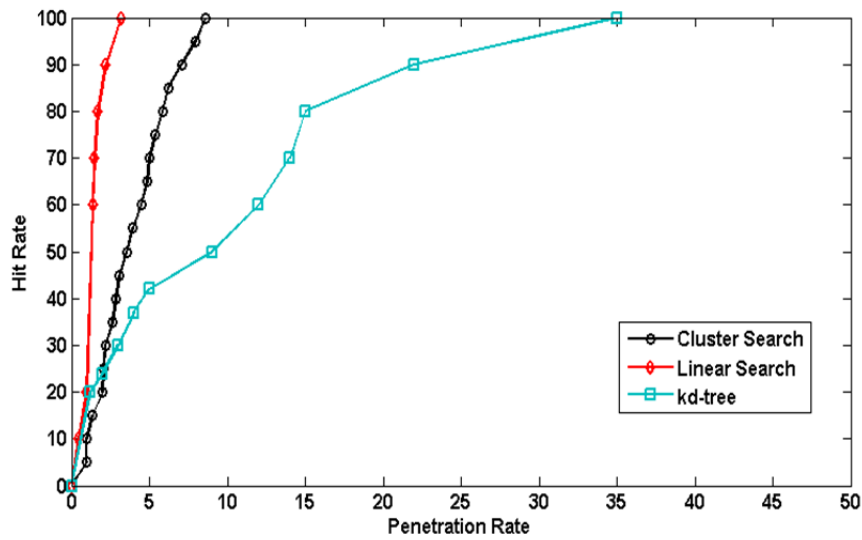


Figure 6-9: Comparison with linear search, clustered search and *kd*-tree search

6.9 Comparison with Existing Methods

We compare our approach with the approach given in [33] and [34] show in Figure 6-10 and 6-11. Result shows that proposed approach is affective for hit rate in higher rage as compare to the other approaches. Table 6-9 gives the comparison with the approach used in Ross et al. [35] with proposed approach. Given results produces by similar database (FVC-2004, DB1).

Hit Rate	Ross et al. [35]	Proposed Approach
	Penetration Rate	
100	51.40	8.56
95	48.75	7.96
90	45.97	7.06
85	43.03	6.2
80	40.04	5.83

Table 6-9: Comparision of proposed approach with ross et al. [35]

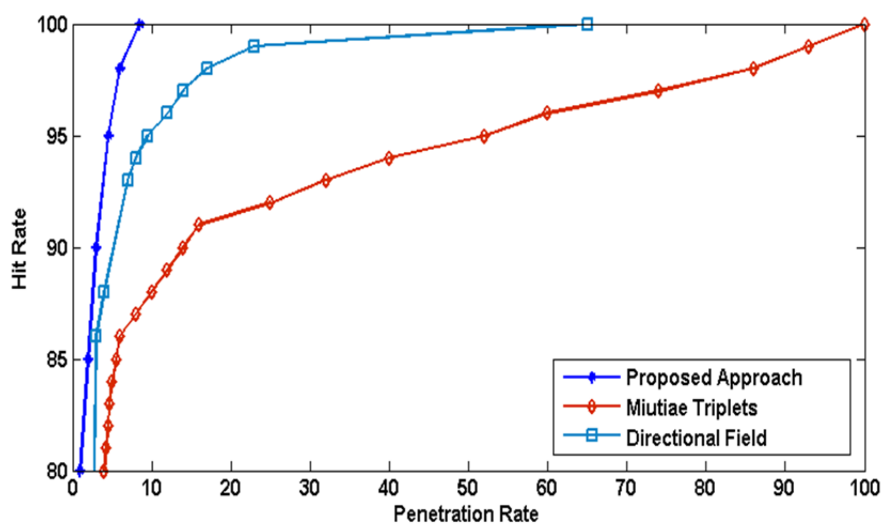


Figure 6-10: Comparison of proposed approach with minutiae triplet [33] and directional field approach [33]

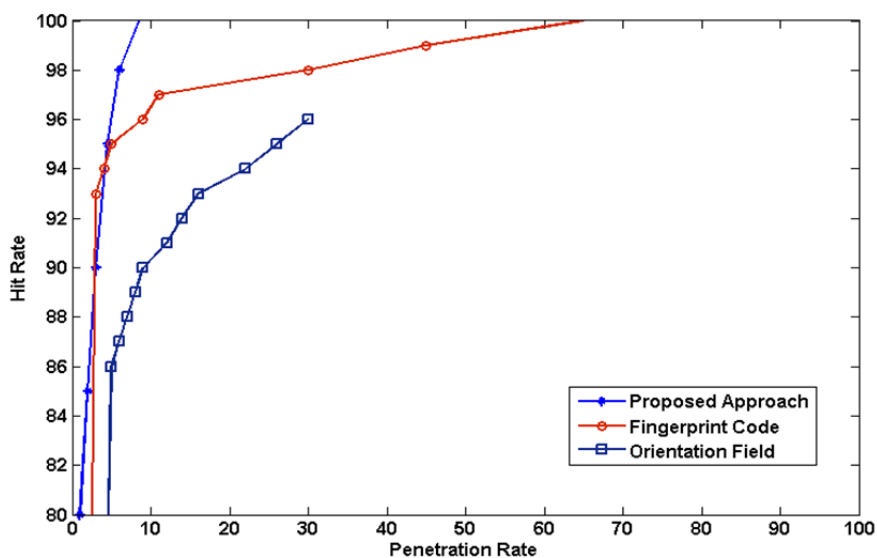


Figure 6-11: Comparison of proposed approach with fingerprint code [33] and orientation field approach [34]

6.10 Conclusion

We test the performance of proposed approach with different database. Experiment with increasing number of fingerprint per individual shows that it does not much affect the performance. Enrolment of single fingerprint can lead to good hit rate due to property of index vector of invariable to rotation and scaling. We test the performance of three different approach, linear search, clustered search and clustered *kd*-tree index search. Results shows that proposed clustered search approach have good balance between response time and accuracy. Comparison with the previous indexing technique show that proposed approach has better accuracy.

Chapter 7

Summary and Future Work

Performances of fingerprint clustering and indexing techniques are test with the different framework and configuration. We have experiment with the linear search, clustering and clustering with the *kd*-tree. Linear search produce higher hit rate correspond to low penetration rate, but takes large time to compute candidate set. Continuous classification with clustering gives moderate hit rate and penetration rate in moderate response time. We also observe that good response time is achieved using *kd*-tree with clustering, but the hit rate is low and penetration rate is high. It is a trade of between response time, hit rate and penetration rate, when we choose one of the above techniques.

Using more than one fingerprint per individual does not affect the much on the performance of indexing system. This confirms the quality of select invariable set of feature for indexing.

Scope of Future Work

1. In proposed approach, we concentrate on rotation and scale invariant feature from the local region of fingerprint. But these

features will not work efficiently in case of transformed fingerprint image. We could work of transform invariant feature for fingerprint indexing.

2. The index key is generated by local topology of minutiae. Different index key of a fingerprint is associated with the different part of fingerprint; hence we suppose that this approach work well partial fingerprint and latent fingerprint. With the availability of latent fingerprint database we can test the performance for conformation.'
3. Geometric hashing is good technique for content based search [13]. For indexing part, we could use geometric hashing for higher response rate.
4. The proposed approach is tested with static and small database. Performance could be test on online and relatively large database to confirm the effectiveness of approach.

Bibliography

- [1] A. Jain, A. Ross, and S. Prabhakar, "An introduction to biometric recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 1, pp. 4-20, January 2004.
- [2] A. K. Jain, L. Hong, S. Pankanti and R. Bolle, "An identity authentication system using fingerprints," in *Proceedings of the IEEE*, vol. 85, 1997, pp. 1365-1388.
- [3] J.D Stosz and L.A. Alyea, "Automated system for fingerprint authentication using pores and ridge structure," in *Proceedings of SPIE*, vol. 2277, 1994, pp. 210-223.
- [4] Salil Prabhakar, Lin Hong, and Sharath Pankanti Anil K. Jain, "Filterbank-based fingerprint matching," *IEEE Transactions on Image Processing*, pp. 846-859, 2000.
- [5] D. Maio, A. K. Jain, S. Prabhakar D. Maltoni, *Handbook of Fingerprint Recognition*, Second Edition ed.: Springer-Verlag New York Inc, 2003.
- [6] A. K. Jain, A Ross and L. Hong, "A multichannel approach to fingerprint classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 4, pp. 348-359, April 1999.
- [7] B. Moayer and K. S. Fu, "A syntactic approach to fingerprint pattern

- recognition," *Pattern Recognition*, vol. 7, no. 1-2, pp. 1-23, 1975.
- [8] A. Senior, "A hidden Markov model fingerprint classifier," in *proceedings of the 31st Asilomar conference on Signals, Systems and Computers*, vol. 1, 1997, pp. 306-310.
- [9] K. Karu and A.K. Jain, "Fingerprint classification," *Pattern Recognition*, vol. 29, no. 3, pp. 389-404, 1996.
- [10] R. Germain, A. Califano, and S. Colville, "Fingerprint matching using transformation parameter clustering," *IEEE Computational Science and Engineering*, vol. 4, no. 4, pp. 42-49, October 1997.
- [11] A. Califano and I. Rigoutsos, "Flash: A fast look-up algorithm for string homology," in *Proceedings CVPR'93*, 1993, pp. 353-359.
- [12] G. Bebis, T. Deaconu and M. Georgiopoulos, "Fingerprint identification using Delaunay triangulation," in *Information Intelligence and Systems, 1999. Proceedings. 1999 International Conference on*, 1999, pp. 452-459.
- [13] B. Lamiroy and P. Gros, "Rapid object indexing and recognition using enhanced geometric hashing," in *European Conference on Computer Vision (ECCV)*, 1996, pp. 59-70.
- [14] B. Bhanu and X. Tan, "Fingerprint indexing based on novel features of minutiae triplet," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 616-622, 2003.

- [15] A. Ross and R. Mukherjee, "Augmenting Ridge Curves with Minutiae Triplets for Fingerprint Indexing," in *Proceedings of SPIE Conference on Biometric Technology for Human Identification IV*, vol. 6539, Orlando, USA, 2007, p. 65390C.
- [16] P. Mansukhani, S. Tulyakov, and V. Govindaraju, "A Framework for Efficient Fingerprint Identification Using a Minutiae Tree," *IEEE SYSTEMS JOURNAL*, vol. 4, no. 2, pp. 126-136, June 2010.
- [17] L. O'Gormann and J. V. Nickerson, "An approach to fingerprint filter design," *Pattern Recognition*, vol. 22, no. 1, pp. 29-38, 1989.
- [18] B.G. Sherlock, D.M. Monro and K. Millard, "Fingerprint enhancement by directional Fourier filtering," *Vision, Image and Signal Processing, IEE Proceedings*, vol. 141, no. 2, pp. 87-94, 1994.
- [19] C. Sharat, N. C. Alexander and G. Venue, "Fingerprint enhancement using STFT analysis," *Pattern Recognition*, vol. 40, no. 1, pp. 198-211, 2007.
- [20] Y. Wan, A. K. Jain L. Hong, "Fingerprint image enhancement Algorithm and performance evaluation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 777-789, 1998.
- [21] L. Hong, Y. Wan, and A. K. Jain, "Fingerprint image enhancement Algorithm and performance evaluation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 777-789, 1998.

- [22] A. K. Jain and F. Farrokhnia, "Unsupervised texture segmentation using Gabor filters," *Pattern Recognition*, vol. 24, no. 12, pp. 167-186., 1991.
- [23] Z. Guo, and R. W. Hall, "Parallel thinning with two-subiteration algorithms," *Communications of the ACM*, vol. 32, no. 3, pp. 359-373, March 1989.
- [24] J. T. Kent and J. M. Baibby K. V. Mardia, "Multivariate Analysis," 1980.
- [25] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Communications of the ACM*, vol. 18, no. 9, pp. 509--517, 1975.
- [26] Guy Shechter, "Perform closest point search or range query using a k-D tree implementation.," *Available from:* <http://www.mathworks.com/matlabcentral/fileexchange/4586-k-d-tree>.
- [27] A. Califano and I. Rigoutsos, "Flash: A fast look-up algorithm for string homology," in *Computer Vision and Pattern Recognition, 1993. Proceedings CVPR'93., 1993 IEEE Computer Society Conference on, 1993*, pp. 353-359.
- [28] M. De Berg, O. Cheong, M. Van Kreveld and M. Overmars, *Computational geometry: algorithms and applications*. Berlin Heidelberg: Springer-Verlag New York Inc, 2008.
- [29] T. Barbu, "Content-Based Image Retrieval Using Gabor Filtering," in

- Database and Expert Systems Application, 2009. DEXA'09. 20th International Workshop on, 2009, pp. 236 - 240.*
- [30] X. Chu and K. L. Chan, "Rotation and Scale Invariant Texture Analysis with Tunable Gabor Filter Banks," *Advances in Image and Video Technology*, pp. 83-93, 2009.
- [31] "<http://bias.csr.unibo.it/fvc2004/databases.asp>".
- [32] L. Wilson, and I. Watson E. Tabassi, "Fingerprint Image Quality," *National Institute of Slandered and Technology, NISTIR7151, August 2004.*
- [33] J.D. Boer, A.M. Bazen and S.H. Gerez, "Indexing fingerprint database based on multiple features," in *Proceedings of Workshop on Circuits, Systems and Signal Processing (ProRISC 2001)*, 2001, pp. 300-306.
- [34] X.D. Jiang, M. Liu and A. Kot, "Fingerprint retrieval for identification," *IEEE Trans. Information Forensics and Security*, vol. 1, no. 4, pp. 532-542, 2006.
- [35] A. Ross and R. Mukherjee, "Augmenting Ridge Curves with Minutiae Triplets for Fingerprint Indexing," in *SPIE Conference on Biometric Technology for Human Identification IV, Orlando, USA, 2007.*