# MINUTIAE LOCAL STRUCTURES FOR FINGERPRINT INDEXING AND MATCHING

Thesis submitted in partial fulfillment
of the requirements for the degree of

*MS by Research*
*in*
*Computer Science*

by

Akhil Vij
200702003
akhil.vij@research.iiit.ac.in

CVIT
International Institute of Information Technology
Hyderabad - 500 032, INDIA
June 2013

International Institute of Information Technology
Hyderabad, India

# CERTIFICATE

It is certified that the work contained in this thesis, titled "Minutiae Local Structures for Fingerprint Indexing and Matching" by Akhil Vij, has been carried out under my supervision and is not submitted elsewhere for a degree.

| Date | Adviser: Prof. Anoop Namboodiri |
| --- | --- |

To My Parents and My Guide

# Acknowledgments

I would sincerely like to thank my guide and mentor Dr. Anoop Namboodiri for his guidance during the last 3 years. He has given me complete freedom as a researcher. There was no pressure of constant meetings and his work style really helped me in my research and and achieve a sense of responsiblity. I was completely free to express my own ideas and work on them. I think such independent thinking is very important for good research. Also, we had various discussions outside academic boundaries which really affected me in my stay at IIIT. I sincerely thank him for all the above. I would also like to thank Dr. C.V Jawahar, Dr. PJ Narayan and Dr. Jayanthi Siwaswamy for their guidance in various courses related to machine learning and computer vision which greatly helped me in my research in biometrics.

Then I would like to thank my CVIT lab mates Atif, Siddhartha, Shrikant, Rohit, Yashaswi, Prateek, Madhulika and Rohan for their constant help and support. This acknowledgement would not be complete without thanking my parents who have constantly supported me and motivated me to work hard.

# Abstract

Human beings use specific characteristics of people such as their facial features, voice and gait to recognize people who are familiar to us in our daily life. The fact that many of the physiological and behavioral characteristics are sufficiently distinctive and can be used for automatic identification of people has led to the emergence of *biometric recognition* as a prominent research field in recent years. Several biometric technologies have been developed and successfully deployed around the world such as fingerprints, face, iris, palmprint, hand geometry, and signature. Out of all these biometric traits, fingerprints are the most popular because of their ease of capture, distinctiveness and persistence over time, as well as the low cost and maturity of sensors and algorithms.

This thesis is focused on improving the efficiency of fingerprint recognition systems using local minutiae based features. Initially, we tackle the problem of large scale fingerprint matching called fingerprint identification. Large size of databases (sometimes containing billions of fingerprints) and significant distortions between different impressions of the same finger are some of the major challenges in identification. A naive solution involves explicit comparison of a probe fingerprint image/template against each of the images/templates stored in the database. A better approach to speed up this process is to index the database, where a light-weight comparison is used to reduce the database to a smaller set of candidates for detailed comparison.

In this thesis, we propose a novel hash-based indexing method to speed up fingerprint identification in large databases. For each minutia point, its local neighborhood information is computed with features defined based on the geometric arrangements of its neighboring minutiae points. The features proposed are provably invariant to distortions such as translation, rotation and scaling. These features are used to create an affine invariant local descriptor called an *Arrangement Vector*, which completely describes the local neighborhood of a minutiae point. To account for missing and spurious minutiae, we consider subsets of the neighboring minutiae and hashes of these structures are used in the indexing process. Experiments conducted on FVC 2002 databases show that the approach is quite effective and gives better results than the existing state-of-the-art approach using similar affine features.

We then extend our indexing framework to solve the problem of matching of two fingerprints. We extend the proposed arrangement vector by adding more features to it and making it more robust. We come up with a novel fixed-length descriptor for a minutia that captures its distinctive local geometry. This distinctive representation of each minutiae neighborhood allows us to compare two minutiae points and determine their similarity. Given a fingerprint database, we then use unsupervised K-means cluster-

ing to learn prominent neighborhoods from the database. Each fingerprint is represented as a collection of these prominent neighborhoods. This allows us to come up with a binary fixed length representation for a fingerprint that is invariant to global distortions, and handle small local non-linear distortions. The representation is also robust to missing or spurious minutiae points. Given two fingerprints, we represent each of them as fixed length binary vectors. The matching problem then reduces to a sequence of bitwise operations, which is very fast and can be easily implemented on smaller architectures such as smart phones and embedded devices. We compared our results with the two existing state-of-the-art fixed length fingerprint representations from the literature, which demonstrates the superiority of the proposed representation.

In addition, the proposed representation can be derived using only the minutiae positions and orientation of a fingerprint. This makes it applicable to existing template databases that often contain only this information. Most of the other existing methods in the literature use some additional information such as orientation flow and core points, which need the original image for computation. The new proposed binary representation is also suitable for biometric template protection schemes and is small enough to be stored on smart cards.

# Contents

# List of Figures

# List of Tables

*Chapter 1*

# Introduction

This thesis is organized in four main parts. In the first part in Chapter-1, we give a brief introduction on biometrics and fingerprints. We talk about various biometric traits such as face, iris, hand geometry etc and see why fingerprints are the most popular and widely used in biometric authentication systems around the world. Then, we have a look at the structure of a fingerprint at a global level and at a local level. We look at the functioning of a fingerprint recognition system in verification mode and in identification mode. Readers who are familiar with biometrics and functioning of fingerprint recognition systems can skip this chapter and move on to the second part. In the second part in Chapter-2, we discuss about the problem of matching two fingerprints. We talk about matching using global features such as core points, ridge structure etc and see how matching at a local level using minutiae-based features leads to a better accuracy. We do an exhaustive survey on minutiae-based local fingerprint matching techniques. Most of these techniques build local minutiae structures from invariant distances and angles in the neighborhood of each minutia. We have a look at existing local structures and their weaknesses and lay motivation for a new local minutia structure. Then in the third part in Chapter-3, we look at the problem of fingerprint identification over a large database. We propose a new minutiae structure called an *Arrangement Vector* that describes the geometric arrangement of neighboring minutiae points around a central minutia. We propose a hash-based novel indexing mechanism using arrangement vectors and show its effectiveness. In the last part of thesis (Chapter-4), we extend the Arrangement vector into a fixed length binary representation for a fingerprint and tackle the problem of fingerprint matching. We finish this thesis in Chapter-5 by summarizing our main contributions and the directions in which we can extend our work.

## 1.1 Biometrics

Recognizing people is a fundamental activity at the heart of our society and daily life. For many activities and applications, ensuring the identity and authenticity of people is a prerequisite. Biometric identification, or *biometrics*, refers to identifying people based on their unique characteristics. These distinctive unique characteristics are called biometric identifiers (or simply biometrics). Physiological biometrics, like fingerprints or hand geometry, are physical characteristics generally measured at some point in time. Behavioral biometrics, like signature , on the other hand, consist of the way some action is carried out and are learned over time. Most of the biometric identifiers are a combination of physiological and behavioral characteristics of a person. For example, fingerprints may be physiological in nature but the usage of the input device (how user places a finger over the fingerprint scanner etc.) depends on the person's behavior. Fingerprints, face, iris, retina, gait, signature and speech are few examples of such biometric identifiers [20]. Figure 1.1 shows some common biometrics used in current applications. Biometric based authentication provides many advantages over conventional methods of identification. Conventional methods of authentication rely either on *possessions* or on *knowledge*. Possessions include physical possessions such as keys, passports and smartcards. Knowledge includes pieces of information that are supposed to be kept secret like passwords and pass phrases. But both of these can be easily misplaced, lost, forged, stolen, forgotten, or shared and do not offer particularly high security. Biometric identifiers on the other hand are less likely to be stolen or shared with other people. These can also be used to provide an extra layer of security over traditional methods of authentication. For example, in many applications, both conventional and biometric based methods are combined to get better security. Passport is one such possession with face and signature biometrics.

The importance of biometrics in the modern technology era has been reinforced by the need for large-scale identity management systems whose functionality relies on the accurate determination of an individual's identity in the context of several different applications. Examples of these applications include distributing government services/products to citizens, performing remote financial transactions, crossing a border etc. The proliferation of web-based services (e.g. online banking) and the deployment of decentralized customer service centers (e.g., credit cards) have further underscored the need for reliable identity management systems that can accommodate a large number of individuals.

### 1.1.1 Selecting Biometric Identifiers

Ofcourse, the question now is what biometric identifier to use if you are designing a biometric authentication system. This is not such a simple question and the answer is very application specific. However, there are certain requirements that a good biometric identifier should satisfy in order to achieve high performance and security. These include the five requirements described by Clarke [42]

- **Universality :** Every person should have the biometric characteristic.
- **Uniqueness :** No two persons should have exactly the same biometric characteristic.
- **Permanence :** The biometric should be sufficiently invariant over a period of time.
- **Collectability :** It should be practically possible to measure the biometric with some sensing device.
- **Acceptability :** Individuals in the target population that will utilize the application should have no strong reasons to object to collection of the biometric.

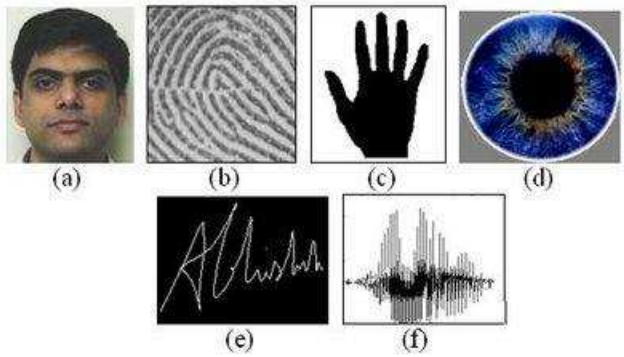



**Figure 1.1** Examples of biometric identifiers that are used for authentication purposes. Physical identifiers include face, fingerprint, hand geometry and iris while behavioral identifiers include signature and voice.

It is the combination of all these attributes that determines the effectiveness of a particular biometric and the corresponding biometric based authentication system. Also, a practical biometric system should have acceptable recognition accuracy and speed, with reasonable resource requirements, harmless to the users and sufficiently robust to various fraudulent methods.

### 1.1.2 Popular Biometric Identifiers

A number of biometric identifiers are in use in various applications (See Figure 1.1). There is no single identifier that satisfies all of these requirements absolutely. A few commonly used biometrics with their pros and cons are discussed below.

- **Face :** Face appearance is a particularly compelling biometric because it is one of the most common methods of recognition that humans use in their visual interactions. Face recognition scores very high on acceptability as the methods of acquiring face images in non-intrusive. But face as a biometric scores low in uniqueness and permeance. Also, facial features can change over the course of time. It is also a big challenge to develop face recognition techniques that can tolerate the effects of aging, facial expressions, slight variations in imaging environment and facial disguise. As a result even automated facial recognition systems require extensive human intervention.

- **Iris :** The colored part of the eye bounded by the pupil and sclera is the iris, being rich in texture it is posited to be distinctive for each person and each eye [24]. Iris based authentication systems are relatively modern and highly accurate and fast. But iris as a biometric scores very less in collectability and acceptability. The design of an iris image capture device that is convenient and unobtrusive is a real challenge. Also, capturing an iris image requires significant cooperation from the user, both to register the image of iris in the central imaging area and to ensure that the iris is at a predetermined distance from the focal plane of the camera.

- **Hand Geometry :** Hand geometry refers to the geometric structure of the human hand. Many geometric features like length and width of the fingers, aspect ratio of the palm or fingers, width of the palm, thickness of the palm etc are relatively invariant and unique to an individual. But hand geometry as a biometric scores a bit low on universality, uniqueness and permeance. On the other hand, hand geometry measurements are easily collectible and non-intrusive as compared to iris.

- **Voice :** Voice recognition (or speaker recognition) tries to identify individuals by how they sound when they speak. Voice capture is unobtrusive and voice print is an acceptable biometric in almost all societies. As a result, voice as a biometric scores very high on collectability and acceptability. But on the other hand, voice is not distinctive and unique. It gets tough to identify an individual from a large database of identities given only voice as a differentiating factor. Also, voice gets affected by a person's health, stress, emotions, age etc. And voice can easily be forged, if your mimicking skills are extraordinarily high.

- **Signature :** The way a person signs can be said to be unique to that person. Signature as a biometric scores high on collectability and acceptability. But a signature can be easily forged and score low on universality and uniqueness. Also, a person's signature need not be permanent and since this is a behavioral biometric, this can change over time.

Most of these famous biometrics score high on some grounds but then lose points when it comes to acceptability or collectability. Fingerprints score high on all of the above five requirements and hence, are used most widely in today's biometric authentication systems.

## 1.2 Fingerprints

Fingerprints are perhaps what the majority of people immediately associate with biometrics. Fingerprints have a long and interesting history of being used as a reliable biometric for identifying a person [17]. Fingerprints are popular because of their ease of capture, distinctiveness and persistence over time. Fingerprints are part of an individual's phenotype and hence are only weakly determined by genetics. Even fingerprints of identical twins are quite different in structure [47]. Fingerprints score very high on uniqueness and it is widely believed in the forensic community that no two people have identical ridge details. History of fingerprints is quite interesting.

### 1.2.1 Evolution of Fingerprinting

- There is evidence that the Chinese were aware of the individuality of fingerprints well over 5000 years ago [21]. But it was not until the late sixteenth century that the modern scientific fingerprint technique was first initiated.

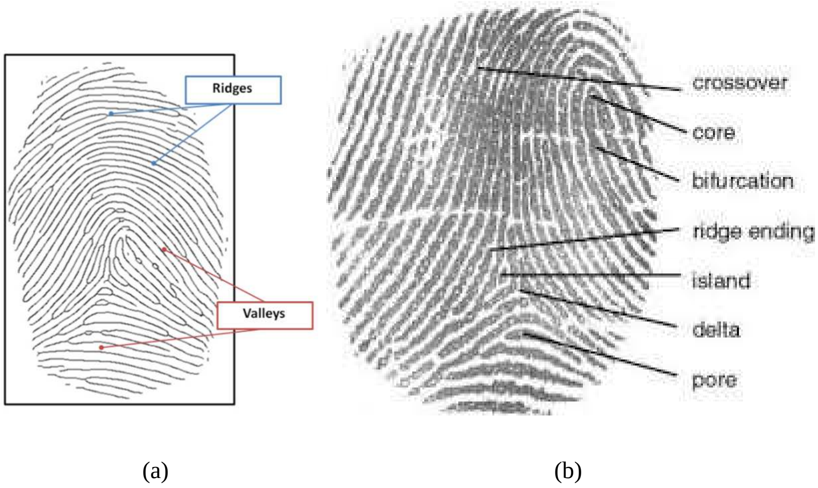



**Figure 1.2** Global and Local Structure present in a fingerprint. (a) The overall global ridge and valley structure (b) The local structure showing minutiae points and pores.

- In 1684, Nehemiah Grew, an English plant morphologist published the first scientific paper reporting his study on the ridge, furrow and pore structure in fingerprints.

- In 1823, Purkinje proposed the first fingerprint classification scheme based on nine categories.

- William Herschel (around 1859) was the first european to recognize the value of fingerprints for identification purposes.

- In 1880, Henry Fauld suggested the individuality of fingerprints based on empirical observations.

- During 1888-1892, Sir Francis Galton conducted an extensive study on fingerprints. In his works, he divided the fingerprints into three major classes and introduced the concept of minutiae features for comparing two fingerprints.

- But modern era of fingerprint recognition systems began with development of the "Henry System" of fingerprint classification during 1899-1900. This system was made by Edward Henry and his two Indian assistants in 1899 [21]. According to this system, five classes were introduced as shown in Figure 1.3. This system was adopted and refined by the FBI [13].

- In the early twentieth century, fingerprint recognition was formally accepted as a valid personal identification method and became a standard routine in forensics.

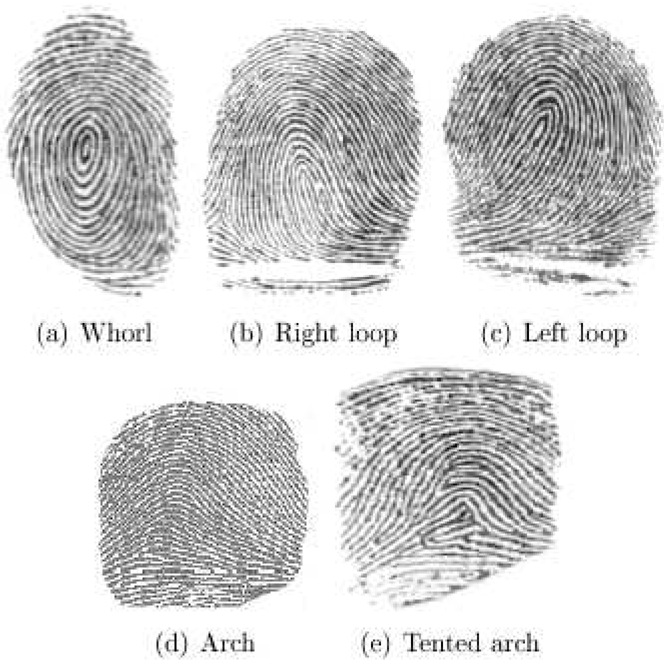



**Figure 1.3** Henry Fingerprint Classification Scheme.

### 1.2.2 Structure and Individuality of Fingerprints

Fingerprints are fully formed at about seven months of fetus development and remain the same throughout the person's life (conformance to permeance requirement). If any scars or superficial damage occur, the skin will grow back in exactly the same arrangement as at birth. They are even one of the last features to decompose after death. A fingerprint is the reproduction of a fingertip epidermis, produced when a finger is pressed against a smooth surface. As we can see in Figure 1.2, the most evident structural characteristic of a fingerprint is a pattern of interleaved ridges (the dark areas) and valleys (bright areas). This ridge-valley pattern creates the distinguishing features of a fingerprint which are identified at three levels.

At a *global* level, the ridges tend to form shapes characterized by regions of high curvature and numerous ridge terminations. These regions are called *Singularities* (see Figure 1.4) and are of three types.

1. delta (represented by the symbol $\Delta$);
2. loop (represented by the symbol $\bigcap$);
3. whorl (represented by the symbol $\bigcirc$);

Another global feature in a fingerprint is the *Core*. Core is defined as the center of the north most loop type singularity. Often the core is used as a reference point to align two fingerprints prior to matching.

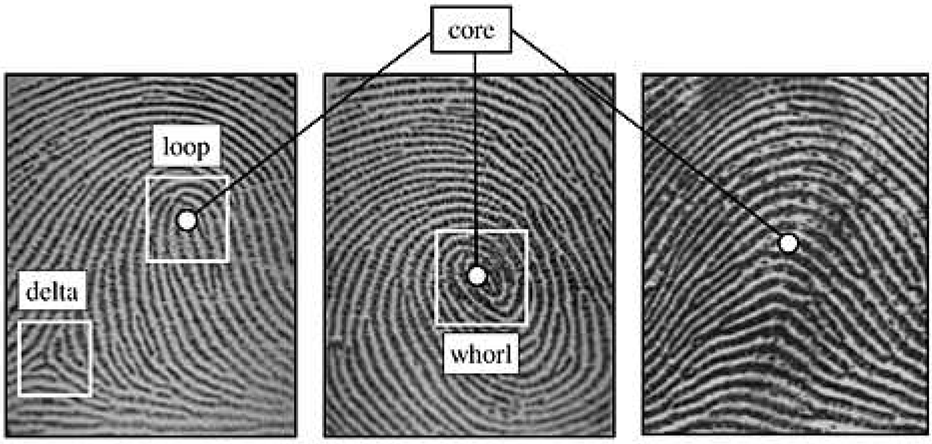



**Figure 1.4** Basic Singularities loop, whorl and delta. The core point as defined in section 1.2.2 is also shown.

The core and singularities are useful to classify fingerprints but are not discriminating enough to match two fingerprints.

At the local level, other important features, called minutiae and pores can be found in the fingerprint patterns. Minutiae points refer to the various ways that the ridges can be discontinuous. The most common minutiae points are shown in Figure 1.2. By increasing the resolution at which the fingerprint image is captured (atleast 1000 dpi), it is possible to extract very minute geometric details like scars, ridge width, breaks etc. The most important low level details are small points over the ridge line called sweat pores. These sweat pores are highly distinctive and around 20-40 pores are sufficient to recognize a person. But they are not used that much in current biometric authentication systems because the image quality and the acquisition resolution has to be very high.

## 1.3 Fingerprint Recognition Systems

A general fingerprint recognition system can be viewed as a pattern recognition system that generally offers a binary decision to a given input. The first time an individual uses a biometric systems, he/she has to go through the *enrollment* phase. During the enrollment, the fingerprints from that individual is captured and stored in the database (See Figure 1.5). In the subsequent uses, fingerprints are captured and compared with the information stored at the time of enrollment. Fingerprint recognition systems normally authenticate people in two modes :

- **Verification Mode** : In this mode, a person's identity is authenticated by comparing the captured fingerprint with the person's fingerprint image/template already stored in the database. It is a one-to-one matching process as shown in Figure 1.6.

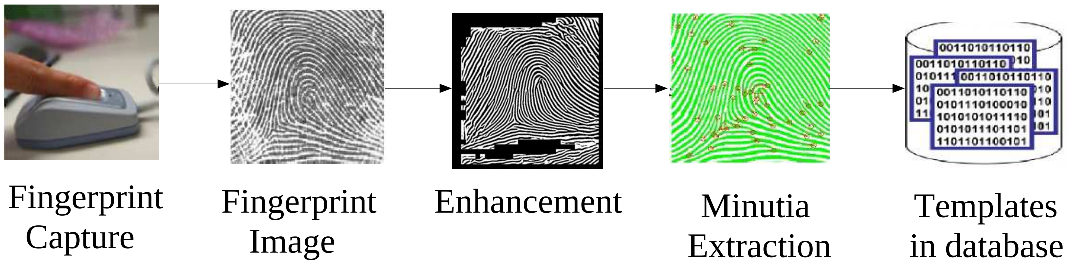



**Figure 1.5** Enrollment Process.

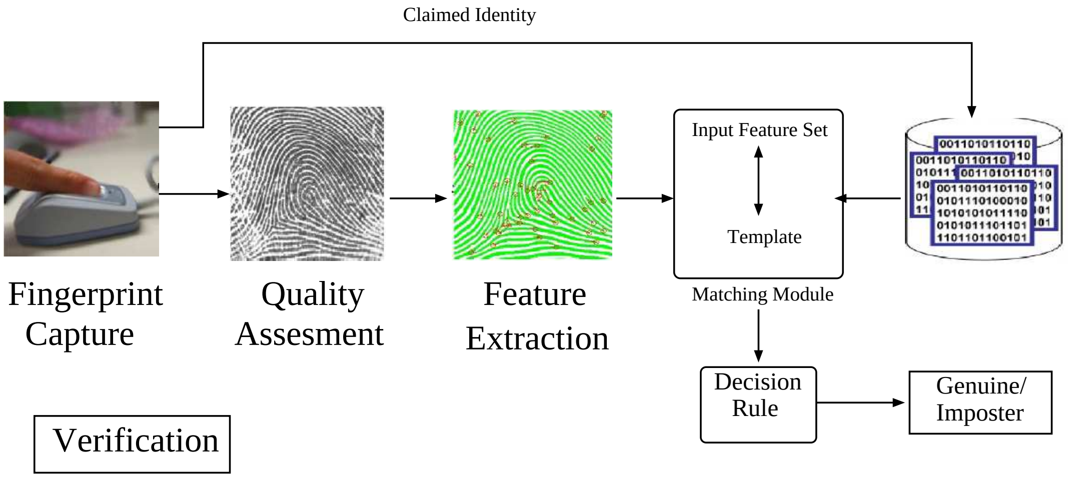



**Figure 1.6** Fingerprint Recognition System in Verfication mode.

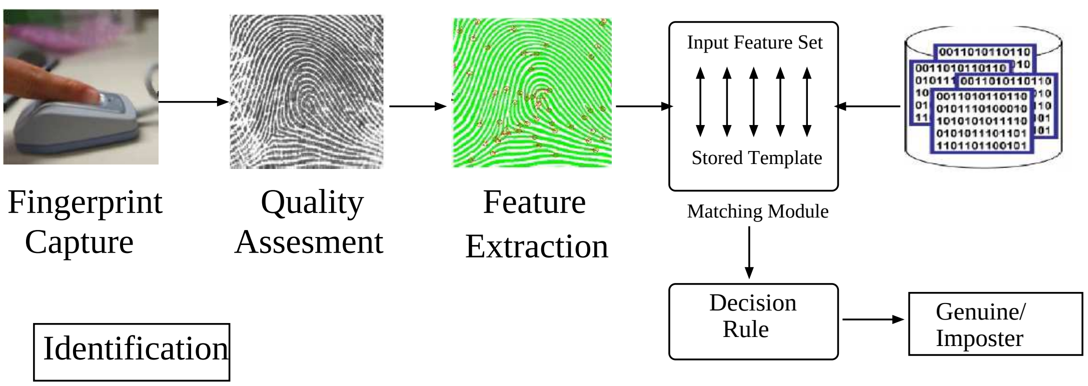



**Figure 1.7** Fingerprint Recognition System in Identification mode.

- **Identification Mode** : In this mode, the system tries to tell us whether a fingerprint query can find a match in the database or not. Therefore, in this mode the system conducts one-to-many comparison to establish an individuals identity (or fails if the subject is not enrolled in the system database) without the subject having to claim an identity. It is a one-to-many matching process as shown in Figure 1.7.

## 1.4 Accuracy Measurements

A general biometric system, irrespective of the biometric identifier being used and the mode of operation, can be viewed as a pattern recognition system that offers a binary decision to a given input. Such a system does make errors and we need a statistical way of measuring the performance of such a system. In the coming two sections, we discuss various metrics that help us in doing exactly that.

### 1.4.1 Fingerprint recognition systems in Verification Mode

Lets consider fingerprint recognition system in verification mode, when a fingerprint is presented as input to the system we expect a binary decision of the type matched/non-matched with the template already stored in the database. However, due to imperfect sensing environment, alterations in user's finger (cuts, bruises etc.), sensor faults and variations in the user's interactions with the sensor, the system may output the wrong answer. We need a framework through which we could measure the system's errors and performance. The Bayesian decision theory [44] offers all we need to measure errors in biometric systems.

The response of the system is usually a similarity score $s$ that measures the similarity between the two images (or minutiae sets). The system has to output a final binary decision based on this similarity $s$. This decision is usually by a system threshold $t$ : pairs of images (or minutiae sets) with similarity score greater than or equal to the threshold $t$ are called *matching* pairs; whereas those pairs producing score less than the threshold $t$ are called *non-matching* pairs. A similarity score is known as a *genuine score* if it is a result of matching of fingerprints of the same user. It is known as an *imposter score* if the two compared fingerprints belong to different users. A good fingerprint matching algorithm leads to good separation between genuiune scores and imposter scores.

A fingerprint verification system can make two type of errors :

- *False Match*: In this case, the system mistakes two fingerprints coming from different individuals to be a match. This is also called *false acceptance*. **False Match Rate** (FMR) is defined as the probability that an imposter scores exceeds the threshold $t$.

- *False Non-Match*: In this case, the system mistakes two different impressions of the same finger as a non-match. This is also called *false rejection*. **False Non-Match Rate** (FNMR) is defined as the probability that a genuine score falls below the threshold $t$.

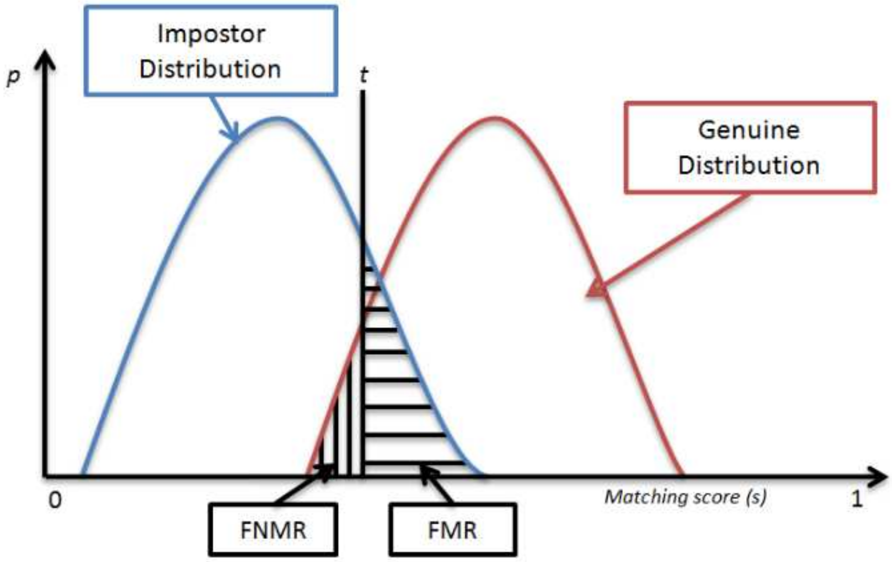



**Figure 1.8** Genuine and Imposter Distributions. Also, shown are False Non-Match Rate and False Match Rate for a given threshold (t) value

To evaluate the accuracy one must collect scores produced from a number of genuine matches (these scores form what is called the *genuine distribution*) and scores produced from a number of imposter matches (these scores form the *imposter distribution*). An example graph of FMR and FNMR over genuine and imposter distributions is shown in Figure 1.8. Both FMR and FNMR are functions of the system threshold *t*. If *t* is decreased to make the system more tolerant, the FMR increases and the FNMR decreases. If *t* is increased to make the system more secure, then FMR decreases and the FNMR increases.

There are other measures also for measuring the system's performance. If we plot a curve between FMR and FNMR, we obtain an important curve called *receiver operating characteristic* (ROC curve). Also, there are other metrics also [9] like :

- *Equal Error Rate (EER)* : denotes the error rate at the threshold t where the FMR and FNMR values are identical.
- *ZeroFMR* : is the lowest FNMR at which no false matches occur.

$$\textbf{ZeroFMR}(t) = \min_t(\textbf{FNMR}(t)|\textbf{FMR}(t) = 0) \tag{1.1}$$

- *ZeroFNMR* : is the lowest FMR at which no false non-matches occur.

$$\textbf{ZeroFNMR}(t) = \min_t(\textbf{FMR}(t)|\textbf{FNMR}(t) = 0) \tag{1.2}$$

An example of curve showing FMR(t), FNMR(t) values along with the above metrics is shown in Figure 1.9. The question still remains that for a specific system how should you set the system threshold

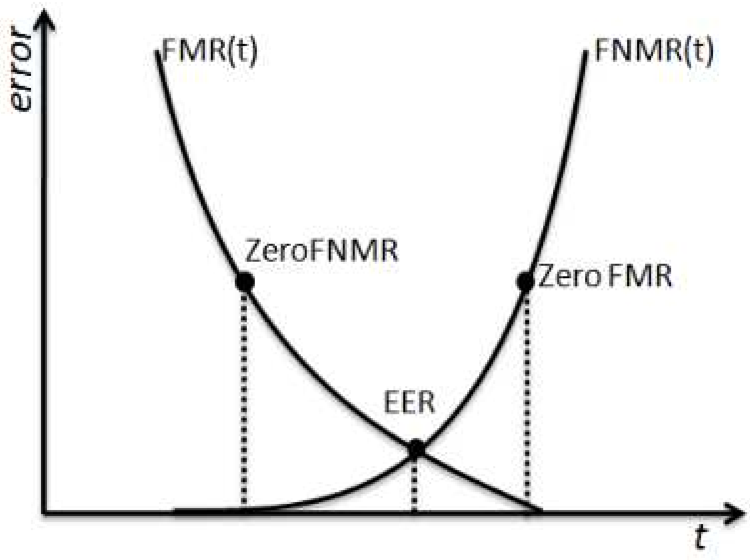



**Figure 1.9** False Non-Match Rate and False Match Rate for a given threshold (t) value. Also shown are points corresponding to ZeroFMR, ZeroFNMR and EER.

*t*, which controls the performance of the system. This is pretty much application specific. For example, some highly secure control applications like nuclear power plants etc. will need a very low FMR. In this case, *t* should be high as the primary objective is not to let in any imposters. Some legitimate users may be denied access in this case but that is a trade-off we have to accept.

### 1.4.2 Fingerprint recognition systems in Identification Mode

FMR and FNMR are used to evaluate the performance of a matching algorithm. Mainly in the Identification mode, a back-end indexing algorithm operates which narrows down the size of the database to be searched. Then the matching algorithm only works with the small part of the database which is outputted by the indexing algorithm. Hence, the goal of the indexing algorithm is to narrow down the number of hypothesis which need to be considered for subsequent matching. The output of such an algorithm is the top N hypothesis. If the query fingerprint is present in the list of top N hypothesis, we take the indexing result as the correct result. Correct Index Power (CIP) is the metric defined [7] to evaluate performance of indexing algorithms. CIP = $(N_{ci}/N_d)$, where $N_{ci}$ is the number of correctly indexed query images, and $N_d$ is the total number of images in the database. The retrieval efficiency is indicated by the *Penetration Rate*, which is the average percentage of database probed over all test fingerprints. Ideally, we would want a high CIP and a low Penetration Rate.

## 1.5 Motivation and Problem Statement

The problem we deal with in this thesis is using local minutiae structures for the purpose of fingerprint indexing and matching. We have chosen to use local minutiae information because that is more tolerant and robust against global transformations and non-linear distortions and gives better results than using global information. As we will see in the second chapter, many local matching techniques have been proposed in the literature but still the problem is far from solved one. There is still a need for more accurate, fast and efficient algorithms. We design a new minutiae-only local structure. We use that local structure for the purpose of fingerprint indexing as we will see in the third chapter. Then we extend that local structure to get a new fixed length representation for fingerprint matching. The main motivations that led us to design a new local minutiae structure and extend it to represent a fingerprint are :

- *Need for more accurate minutiae-only algorithms* : Many of the existing techniques use several extra features besides minutiae information [10]. These features are computationally expensive to compute and there is no standard definition of these features since these features are not defined in the standard minutiae template ISO/IEC 19794 [1]. We want to keep the template size as minimum as possible so that algorithm could be used on light architectures such as inexpensive smart cards etc.

- *Need for new fixed length fingerprint representation* : Very few fixed length representations have been proposed in the literature. We need a new and a more robust fixed length representation that can tolerate non-linear distortions and global transformations.

- *Suitability for template protection techniques* : Biometric template protection is an area which is rapidly growing because of its great benefits (like protection against attacks, nonreversibility, diversity etc). Many recently proposed template protection techniques [37] and [3], require a fixed length representation of the biometric identifier as the input.

## 1.6 Summary

This chapter gave a general introduction on biometrics. We had look at various biometric identifiers and we also saw some basic requirements that a biometric identifier should fulfill. Then we saw that fingerprint as a biometric fulfills almost all of the above requirements. Then we gave a brief history of how present day fingerprint authentication systems have evolved. Then we had a look the structure of a fingerprint at global and at local level. Then we discussed about functioning of a present day fingerprint authentication system. We saw that present day systems operate in two modes : verification and identification. Then we saw how to measure the accuracy and performance of such systems in both modes. Finally, we laid down the motivation for this thesis and the existing challenges that are present and need to be tackled.

*Chapter 2*

# Existing Minutiae Based Structures - A Survey

In the previous chapter, we gave a brief overview of biometrics and biometric identifiers. We talked about the structure of fingerprints and functioning of fingerprint authentication systems. Then we looked at how to evaluate the performance of these systems and the need for better accuracy and performance. In this chapter, we first discuss existing fingerprint matching techniques (section 2.1). We classify these techniques as local or global in nature depending on the fingerprint features they use for matching purposes. Then we look at the weaknesses of global matching and how matching at a more local level can improve performance . Finally, we do a thorough survey of existing literature on local minutiae based matching (section 2.2). Then we look at the weaknesses of the current methods and lay motivation for a new fingerprint representation using local minutiae structures (sections 2.4 and 2.5).

## 2.1 Matching Two Fingerprints

The problem of matching two fingerprints (generally known as fingerprint verification) can be described as given two fingerprints, we need to return either a degree of similarity or a binary decision (matched/non-matched). Fingerprint Recognition at its very core is a 2D point pattern matching problem and has been researched extensively in the Patter Recognition community. Many effective solutions have been proposed in the literature, but the problem still cannot be called a fully solved one. There is still need for more accurate methods as the performance of most of the proposed algorithms is still lower than the theory estimation. Also, we need more faster algorithms as the size of the database even after filtering and indexing is quite large.

### 2.1.1 Major Problems Faced

Large intra class variability is one of the major problems that we face in matching two fingerprints. Intra-class variation refers to the large variability in different impressions of the same finger. There are many reasons that lead to such high intra-class variation :

- Global transformations: The same finger may be placed at different locations or may be rotated at different angles with respect to the sensor surface during different acquisitions. This can result in a global translation or rotation of the fingerprint area. Any good algorithm has to account for global transformation like translation, rotation, scale and shear.

- Partial Overlap: These global transformations mentioned above often cause part of the fingerprint area to fall outside the sensor's field of view, resulting in a smaller overlap between the foreground areas of the two fingerprints. In simpler terms, a lot of minutiae points present in one fingerprint may not be present in the other fingerprint. Dealing with these missing or spurious minutiae points is a major challenge for most of the fingerprint matching techniques.

- Non-linear distortions : Non-linear distortion refers to the compression or stretching of skin due to skin plasticity. This comes up as we try to map the 3D shape of a fingerprint onto the 2D surface of the sensor. The components of the force that are non-orthogonal to the sensor surface produce non-linear distortions. These distortions are quite local in nature and handling these distortions is a major open challenge. There are other reasons also which lead to these distortions [6]. These include the sensor orientation with respect to the finger, the applied pressure, the disposition of the subject, the motion of the finger prior to its placement on the sensor, the skin moisture and the elasticity of the skin. Also some users apply excessive force to create intentional elastic deformations. The effect of these non-linear distortions is quite large as shown in Figure 2.1.

- Pressure and skin condition : We would ideally want to capture the ridge structure of a fingerprint with high accuracy. For this, part of the finger being imaged has to be in uniform contact with surface. But in real life, because of pressure, dryness, skin disease, sweat, dirt, humidity etc we get a non-uniform contact and hence, a noisy image. Such noisy images result in a lot of spurious minutiae points which the algorithm has to deal with.

### 2.1.2 Existing Solutions

Many effective methods have been proposed in the literature. Depending upon the fingerprint features used, the methods can be classified as :

**Correlation-Based Matching** : These methods usually work directly on fingerprint images by superimposing the two images and computing the correlation between the corresponding pixels for different alignments. Usually a cross-correlation measure representing similarity between the two images (like sum of squared difference of intensity values) is computed. But directly computing correlation is not a good solution because :

  - Because of non-linear distortions, two global fingerprint patterns cannot be reliably correlated.

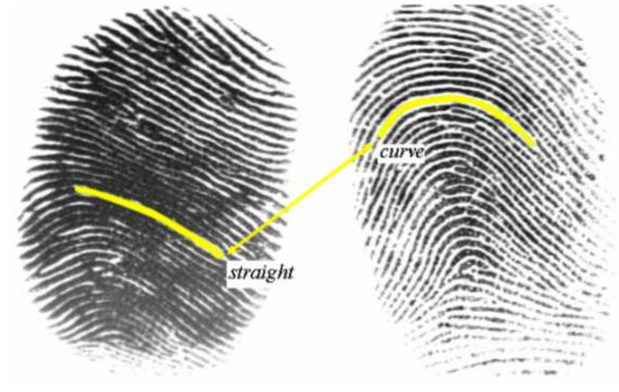



**Figure 2.1** Effect of Non-linear distortion. Shown are two different impressions of the same finger from the FVC 2004 database. The same ridge line is straight in one image and curved in the other image due to non-linear distortion.

- Skin condition and finger pressure cause the image's brightness, contrast and ridge thickness to vary significantly across different impressions of the same finger. Plain simple correlation of pixels will not give you a good result in this case.
- Pixel correlations have to be computed for many alignments. Since, the space of possible alignments is exponential with respect to the number of minutiae, correlation based methods are very expensive.

**Ridge Features Based or Global in nature** : Global features such as singular points, orientation flow around core points, Poincarè index and average ridge-line frequency represent the global pattern of ridges with uniform model. Many techniques like Tico in [33], Medina-Perez in [32], G.Ng and X.Tong in [16] and Wang in [27] use global orientation flow and frequency for matching purposes. Many methods also use spatial relationship and geometrical attributes of the ridge lines [26] and [57]. Y.He and J.Tian use global texture information present in the fingerprint in their work [57]. Unfortunately, most of the global matching algorithms are computationally demanding and lack robustness with respect to non linear distortions. Another major issue is that most of these global features are not present in the standard ISO/IEC 19794-2(2005) minutia template [1] and have to be computed separately starting with the original image. Many of above techniques [11] require prior alignment of the two fingerprint images which is computationally expensive. Since non-linear distortions make impressions of the same finger differ in terms of global structure, these techniques are not able to handle local non-linear distortions. Local minutiae based fingerprint matching methods generally outperform their global counterparts. However, global features are good for the task of fingerprint classification or can be used in conjunction with more discriminative and robust minutiae based features.

**Minutiae Based or Local in nature** : Minutiae based techniques are the most popular due to the compactness of the minutiae templates and also because these are the features that fingerprint experts look at while doing the visual inspection of fingerprints. Minutiae are extracted from the two fingerprints and stored as a set of points in 2-D space. The problem now reduces to 2D point pattern matching problem. Minutiae based matching methods can be further broken down into global and local minutiae matching methods. Global methods [11] search the space of possible transformations to find a global alignment between the two fingerprints that results in the maximum number of minutiae pairings. Hough transform based techniques fall in this category. Non-linear distortions is a major issue with these techniques. Also, the number of alignments can be exponential (w.r.t number of minutiae) and hence, these techniques are quite slow in practice. Local minutiae matching methods construct local minutiae structures around each minutia point. Then the two fingerprints are compared according to these local structures. These methods use relative distances and angles between neighboring points and the minutia point to construct the local structures. These attributes (relative distances and angles) are invariant with respect to global transformations such as translation, rotation , scale and shear . and therefore can be used for matching without any apriori global alignment. These local structures can also handle non-linear distortions better than global minutiae matching techniques. Techniques such as [35, 25, 54, 45, 15, 52] and [51] use local structures around minutiae points for the purpose of matching. These local structures are described in detail in the next section.

## 2.2 Local Minutiae Structures

Local minutiae matching consists of comparing two fingerprints according to local minutiae structures. These structures are characterized by attributes that are invariant with respect to global transformations. Local structures usually carry information about the local geometry of neighboring points around the central minutia point. Depending on how you choose the neighboring points for the central minutia, these local can be broadly classified into nearest neighbor based and fixed radius based structures.

### 2.2.1 Nearest Neighbor based structures

Nearest neighbor techniques first find out the $k$ spatially closest points with respect to the central minutia and then construct a local structure from these $k$ nearest points. In most of the cases, normal euclidean distance between points is considered for measuring distance. Many nearest neighbor based structures have been proposed in the literature.

The simplest of these structures is based on minutia pairs, where the distance between the pair and the orientation of each minutia with respect to the line connecting them can be used as invariant attributes. Another commonly used nearest neighbor structure is the minutia triplet, where relative features (distances and angles) computed from combinations of three minutiae are used for the purpose of fingerprint matching and indexing [7, 28] and [5]. A number of other nearest neighbor based

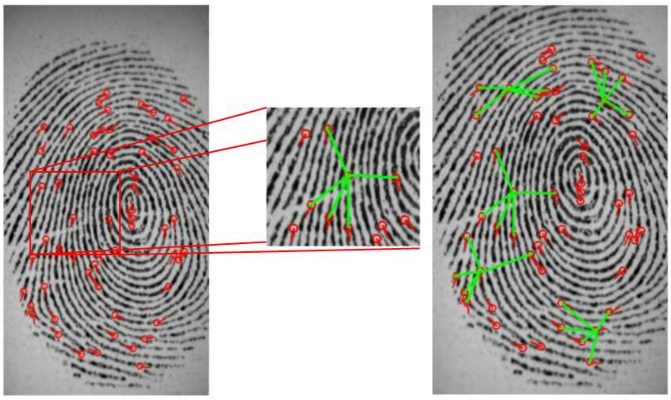


**Figure 2.2** The k-plet structure proposed by Chekkerur in [45].

local minutiae structures (with number of neighbors $> 3$) have also been proposed. Chekkerur and Govindaraju in their work [45] presented a new local structure called k-plet.

The k-plet consists of a central minutia $m_i$ and *k* of its nearest neighboring minutiae $\{m_1$,$m_2$...$m_k\}$. Each neighborhood minutia is defined in terms of its local radial co-ordinates ($\Phi_{ij}$, $\theta_{ij}$, $r_{ij}$) where $r_{ij}$ represents the euclidean distance between the minutiae $m_i$ and $m_j$, $\theta_{ij}$ is the relative orientation of minutia $m_j$ w.r.t to central minutia $m_i$ and $\Phi_{ij}$ is the direction of the edge connecting the two minutiae points. An example of k-plet structure is shown in Figure 2.2. Kwon and Yun in their work [8] propose a similar nearest neighbor based local structure called *k-directional nearest neighbor* (*k*-DNN). With central minutia as the new origin and its orientation as the new X-axis, they divide the plane into *k* slots and find the nearest neighbor for each slot. For the $r_{th}$ nearest neighbor, they encode the relative distance, angle and orientation from the central minutia as $d_r$, $\theta_r$ and $\Phi r$. An example of k-DNN structure is shown in Figure 2.3. Jiang and Yau [54] also proposed a new local structure based on relative distance, radial angles, orientation and ridge count as shown in Figure 2.4.

One main advantage of nearest neighbor based structures is that it leads to a fixed length descriptor that describes the geometric layout of these neighbors around the central minutia. This fixed length descriptor can be easily matched with other such descriptors. But one main drawback is that these structures are not tolerant enough to missing or spurious minutiae points. These structures break down when the image is too noisy or when there is very small overlap between two impressions of the same finger.

### 2.2.2 Fixed Radius based structures

In the fixed radius based structures, all the minutiae points that lie inside the sphere of given radius *R* (with central minutia as the center) for the neighborhood for that central minutia. Again, normal euclidean distance is considered for measuring distance.

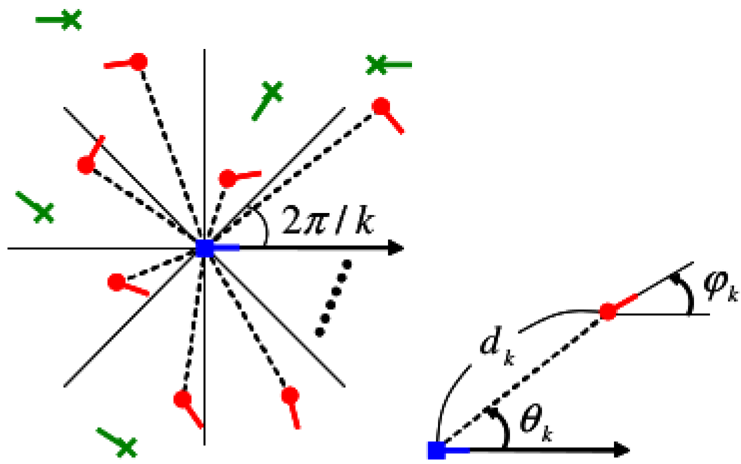
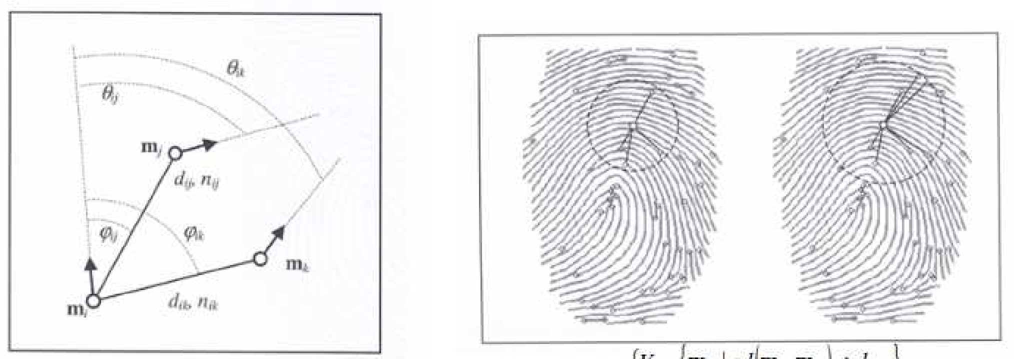



**Figure 2.3** The k-Directional Nearest Neighbor (k-DNN) structure proposed by Kwon in [8] (for k=8). The red minutiae points represent the nearest points in each slot and the green minutiae points represent the second nearest. Encoding the parameters for the nearest neighbor is also shown.

Ratha and Pandit [35] proposed a novel graph representation of a fingerprint based on fixed radius local structure as shown in Figure 2.4. They call their representation Minutiae Adjacency Graph (MAG) and present a robust and accurate matching technique for MAGs based on local structural similarity. However, like most of the fixed radius based approaches their method does suffer from border errors. Major issue is handling of minutiae that lie on the boundary of the sphere. In particular, minutiae close to the local-region border in one of the two fingerprints can be mismatched because local distortion or location inaccuracy may cause the same minutiae to move out of the local region in the other fingerprint. The technique proposed by Feng [25] does not suffer from border errors and can be considered a state-of-the-art fixed radius local matching algorithm. They deal with the border problem by considering minutiae not close to the border as *matchable* and minutiae near the border as *should-be-matchable* . Most of the fixed radius based approaches lead to a variable length descriptor (since the number of minutiae in the sphere will depend on the minutiae density around the central minutia) which is more complex to match.

### 2.2.3 Comparison of Nearest Neighbor and Fixed Radius based Structures

Nearest neighbor based techniques usually represent the neighborhood of the central minutia with the geometric arrangement of fixed number of neighboring minutiae (lets say nearest *k*). This leads to a fixed length descriptor that can be easily matched with other descriptors. Sometimes the matching can just be a simple difference or bitwise-AND of the two descriptors. Such matching can be quickly and efficiently done even on light architectures such as smart card or system-on-a-chip. On the other hand nearest neighbor based techniques cannot deal very well with missing or spurious minutiae points. Even




$$\mathbf{v}_i = [d_{ij}, d_{ik}, \theta_{ij}, \theta_{ik}, \varphi_{ij}, \varphi_{ik}, n_{ij}, n_{ik}, t_i, t_j, t_k],$$

$$S_i = \{V_i, E_i\} = \begin{cases} V_i = \{\mathbf{m}_j \mid sd(\mathbf{m}_i, \mathbf{m}_j) < d_{\max}\} \\ E_i = \{e_{ij} \mid (i, j, d(\mathbf{m}_i, \mathbf{m}_j), rc(\mathbf{m}_i, \mathbf{m}_j), \phi_{ij})\} \end{cases}$$

(a) (b)

**Figure 2.4** Local structures proposed by Jiang and Ratha. (a) The nearest neighbor based structure by Jiang encoding relative distance ($d$), orientation($\theta$), radial angles($\Phi$), ridge count($n$) and type($t$) of the neighboring point. (b) The fixed radius based structure by Ratha. Each edge towards a neighbor encodes relative distance, orientation and ridge count.

with a controlled acquisition environment, noisy fingerprint images can be acquired where different impressions of the same finger can have different number of minutiae. Fixed radius based techniques can deal robustly with such noisy images. But these approaches generally lead a variable length descriptor which is tough to match. Hence, the matching phase now becomes more complex and computationally expensive. Also, border errors as described in the earlier section have to be gracefully handled.

There are certain hybrid structures that take the advantages of both nearest neighbor and fixed radius structures and throw away their respective disadvantages. Minutiae Cylinder Code (MCC) [40] is one such descriptor that can be considered state-of-the-art in local minutiae based fingerprint matching. MCC and other hybrid structures are described in detail in the next section.

## 2.3 Minutiae Cylinder Code and other Hybrid Structures

Hybrid structures usually combine the advantages of both nearest neighbor-based and fixed-radius structures, without suffering from their respective drawbacks. Minutiae Cylinder code (MCC) [40] is the state-of-the-art in this area. MCC is a fixed-radius approach and therefore it can handle missing/spurious minutiae better than nearest neighbor-based approaches. But unlike other fixed-radius approaches, MCC outputs a fixed length descriptor for each minutia and this makes the computation of local structure similarities very simple. Infact cylinder matching is very simple and fast, it reduces to

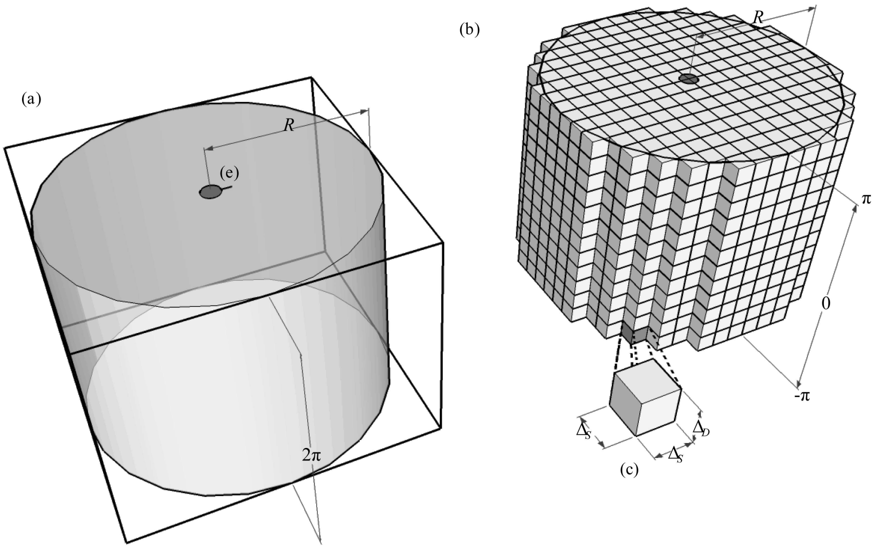



**Figure 2.5** Minutia Cylinder Code structure. (a) The main cylinder is shown enclosed in the cubiod. (b) The cylinder structure is discretized into sections and sections are divided into cells. (c) An individual cell is shown, for each cell a numerical value is calculated from its neighboring minutiae.

just a sequence of bit- wise operations (AND, XOR) that can be efficiently implemented even on very light CPUs. MCC also handles border errors and local non-linear distortions gracefully.

In MCC, the local structure for each minutia *m* is represented by a cylinder of radius R and height 2Π whose base is centered at ($x_m$,$y_m$), the 2D location of minutia *m*. The cylinder is enclosed inside a cuboid whose base is aligned according to the minutia direction $\theta_m$. The cylinder is divided into sections : each section corresponds to a directional difference in the range [-Π,Π] .The sections are discretized into $N_C$ = $N_S$ *X* $N_S$ *X* $N_D$ cells as shown in the Figure 2.5. During the creation of the cylinder, a numerical value is calculated for each cell, by accumulating contributions from minutiae in the neighborhood of the projection of the cell center onto the cylinder base. Fixed Radius ($3\sigma_S$) is used to define the radius of the neighborhood. While calculating the contributions only relative distances and directional differences are used between minutiae. The contribution of each minutia $m_t$ to a cell (of the cylinder corresponding to a given minutia *m*) depends both on:

- spatial information (how much $m_t$ is close to the center of the cell)
- directional information (how much the directional difference between $m_t$ and *m* is similar to the directional difference associated to the section where the cell lies.)

In other words, value of a cell represents the likelihood of finding minutiae that are close to the cell and whose directional difference with respect to *m* is similar to a given value. Since, the number of cells are fixed $N_C$, this leads to a fixed length descriptor that can be easily matched.

But there are some weaknesses present in the MCC structure also. Their representation of a minutiae neighborhood is not provably invariant to affine deformations. Recently, many attempts have been made at reconstructing a minutia template starting from a minutia cylinder set. Many of these attempts have been quite successful, which puts a question mark on the degree of non-reversability of MCC representation. MCC is a fixed length representation for a minutia point and not for the fingerprint. We have local similarity scores representing how well two minutiae points match. But in order to compare two fingerprints a single value (global score) denoting an overall similarity has to be obtained from these local similarities. Hence, an extra global consolidation stage is required. Also, a minutiae representation of a fingerprint cannot be applied directly in the recently developed template protection schemes such as [37] and [3], which require as an input a fixed-length feature vector representation of fingerprints. Many attempts have been made to come up with such a fixed-length representation which is invariant to global transformations but still the problem is far from solved.

## 2.4 Need for new Fingerprint Representation

As we saw in the previous section, there are certain weaknesses in most of the nearest neighbor and radius based approaches. Even state-of-the-art hybrid representation such as MCC has some shortcomings. A fixed length descriptor which completely describes the geometric structure of minutiae points present in a fingerprint is needed. Such fixed length representation would be suitable to template protection schemes. Such representation could then be intelligently quantized to get binary fingerprint representation. Such binary representation would then reduce fingerprint matching to a series to binary operations (such as bitwise-AND/XOR). This would significantly improve the matching speed and matching could then be efficiently done on small architectures such as smart cards. In Chapter 4, we propose a novel fixed length fingerprint representation and tell its benefits.

## 2.5 Summary

In this chapter, we introduced the problem of reliably matching two fingerprints. We saw that matching could be done using global features such as overall ridge structure and core points, and also using local features extracted from minutiae points. Local matching has several advantages over global matching and is much more tolerant towards non-linear distortions. Then we did a thorough literature survey of local minutiae matching and the minutiae structures that they use for matching purposes. We classified the minutiae structures as nearest neighbor based or fixed radius based. We looked at the advantages and disadvantages of both of them. Then we had a look at hybrid structures such as MCC that takes advantages of both of them. Finally, we discussed the weaknesses of these structures and gave motivation on why we would need a new minutiae structure and a new fixed length representation for fingerprints. In the next chapter, we propose a new minutiae representation called an *Arrangement Vector* and use that representation for indexing large fingerprint databases.

*Chapter 3*

# Fingerprint Indexing Based on Local Arrangements of Minutiae Neighborhoods

In this chapter, we talk about the problem of retrieving fingerprints from a very large fingerprint database. The problem is termed as *Fingerprint Indexing*. The chapter is divided into six sections. In the first section, we give the general introduction for the problem and see various real-world scenarios where the problem comes up in practice. We will look at the challenges associated with fingerprint indexing problem and some basic solutions. In the second section, we give a brief literature survey about the existing fingerprint indexing algorithms. We will have a look at their weak points and why a new fingerprint indexing algorithm is needed. In the third section, we will propose a new minutia representation called *Arrangement Vector* and use this representation for indexing purposes. The fourth section describes all the experiments conducted and detailed discussions of results. We discuss how we handle missing or spurious minutiae points which is one of the major challenges. We do detailed time analysis and look at how much time we gain when using our indexing algorithm rather than going for one-to-one matching for the entire database. Finally, we compare results with existing algorithms in the fifth section. We conclude this chapter with the sixth section in which we discuss how we can extend the proposed minutiae representation for matching purposes.

## 3.1 Searching Large Fingerprint Databases

Biometrics especially fingerprints play a major role in automated personal identification systems deployed to enhance security all over the world. Many of such automated systems have very huge underlying fingerprint database. In these large identification systems, the goal is to determine the identity of a subject from a large set of users (possibly in millions) already enrolled in the database. Many of these databases contain tens of millions of records and a single identification request can take a significant amount of time even with the modern day computing technologies. The search time becomes an important factor in the success and failure of such systems. The Aadhar project under the Unique Identification Authority of India is one such project which maintains a database of Indian citizens containing their ten fingerprints and other data. With more than one billion enrollments expected, this may lead to the biggest biometric database ever created (billions of fingerprints). Hence, there is a need for reducing the search space by narrowing down the size of the database. Ofcourse, there are many problems that come up when we try to do exactly that.

### 3.1.1 Major Problems Faced

Fingerprint Identification over a large database is still an open problem and poses many challenges. First of all the size of the database is the biggest challenge faced. In such cases, the identification typically has an unacceptably long response time. The process can be speeded up by reducing the number of comparisons that are required to be performed. Sometimes, information about age, sex, caste and other demographic data can be used to reduce the portion of the database searched. Such information is not always available in many cases. In the general case, information intrinsic to the fingerprint samples has to be used for an efficient retrieval. Also, often there are significant distortions between different impressions of the same finger making the problem even tougher. Three common approaches have been proposed in the literature for solving the problem of searching large fingerprint databases.

### 3.1.2 Possible Solutions

The three classes of solutions include :

- **Brute Fore solution** : This refers to performing a sequence of one-to-one verification with the entire database. This is, ofcourse, a time consuming solution and not at all feasible in practice. Lets take the example of the above Aadhar project. Even if one matching takes around 1 millisecond, still enrolling one indian citizen will take around 300 hours (assuming Indian population of 1.2 billion). We need something more smarter and efficient.

- **Fingerprint Classification** : Fingerprint Classification refers to the problem of assigning a fingerprint to a class in a consistent and reliable way. So basically it involves labeling each fingerprint image into one of a few known global patterns and restricting the matching of query to sample of the same class. Arch, Tented Arch, Loop (left loop and right loop), whorl and double loop are the major global patterns (or classes). But Fingerprint Classification has its own drawbacks. First of all, small inter-class variation and large intra-class variation make this a tough problem. Uneven distribution of fingerprints in different classes is another issue. Most of the fingerprints (around 90%) belong the loop and whorl patterns. So even after classification, the search space is not narrowed down by a significant amount. The number of global patterns (or classes) are quite less. Also, fingerprint classification is generally based on global features like global ridge structure, singular points [39, 38] and [4] and such features are tough to compute in a noisy image. Hence,

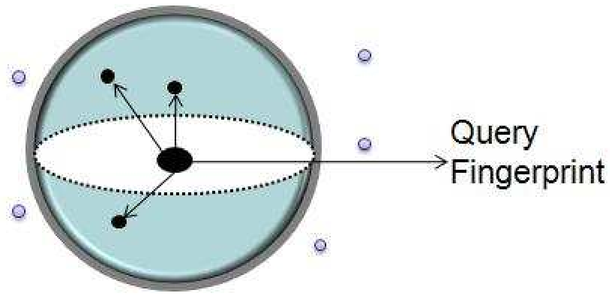



**Figure 3.1** Figure showing goal of Fingerprint Indexing. Matching is only done with vectors that are close to the query vector.

we need some different technique to narrow down the search space and indexing provides the answer.

- **Fingerprint Indexing** : This technique is generalization of the classification approach, where the database is automatically divided into a large number of possibly overlapping subsets. The indexing function predicts the subsets that need to be searched for each query image. The goal here is to find a mapping (or feature representation), that maps similar fingerprints to close points in a multi-dimensional space. So, we associate fingerprints with these multi-dimensional numerical feature vectors summarizing their main features. Retrieval is performed by matching the input fingerprint with those in the database whose corresponding vectors are close to the searched one as shown in Figure 3.1. There have been many fingerprint indexing techniques proposed in the literature.

## 3.2 Literature Survey on Fingerprint Indexing Techniques

Based upon the fingerprint features used, the existing fingerprint indexing techniques can be classified as :

**Global Representations** : Global features like average ridge-line frequency, orientation flow around core points and Poincarè index represent the global pattern of ridges with uniform model. The algorithms used in [50] and [53] belong to this category. However, these features are more suited for classification purposes and are not particularly good at handling distortions and global transformations. These techniques often require prior alignment of fingerprint images in the database and use the location of singular points. In many low quality and noisy images, it is tough to locate the singular points reliably and thus, such images are rejected in this case. These features are usually used in conjunction with more discriminative features to further narrow down the search [53].

**Orientation Flow Features** : Techniques which use features such as local ridge line orientations [39, 38] and local ridge-line frequencies [46] fall under this category. However, one disadvantage of using features obtained from orientation image is that these features are not present in the ISO standard minutiae templates and have to be computed separately starting with the original image.

**Minutiae-based Features** : Most minutiae based indexing techniques [43, 55, 23, 7], derive geometric features from sets of minutiae points that are robust in presence of rotations and translations and use hashing techniques for searching. Some techniques like [29], and [41] form complex structures from minutiae representations and use them for indexing purposes. *Minutiae Cylinder Codes* or MCC [40] was proposed recently and have been demonstrated to be a highly effective method for representing a minutiae neighborhood for the purposes of fingerprint matching as well as indexing. While the MCC representation of a minutiae neighborhood is not provably invariant to affine deformations, the regularizations performed during the computation make them very robust. In this work, we try to ensure affine-invariance of the minutiae neighborhood features and explore their effectiveness for the purposes of indexing.

**Other Features** : Features such as Fingercode [4] and SIFT-based features [56] use wavelet responses to encode local textures. Some techniques also try to combine different types of features to improve the results [5]. There are also techniques which are based on match scores [2] and hash functions [48].

Minutiae-based fingerprint indexing schemes generally give better results than other techniques. Indexing based on Minutia Cylinder Code (MCC) can be considered as state-of-the-art in the area of fingerprint indexing. But MCC representation of minutiae neighborhood is not provably invariant to affine deformations. We have developed a new representation for a minutia point that is provably invariant to affine deformations and made it applicable directly to the existing minutiae based templates. The representation does not require detection of singular points or prior alignment of the templates. Unlike MCC, we have even avoided the use of minutiae orientations to make the method applicable to the widest variety of existing templates. We also propose a way of handling missing or spurious minutiae points.

## 3.3 New Minutiae Representation : Arrangement Vector

The atomic unit of our representation is a fixed-length descriptor for a minutia that captures its distinctive neighborhood pattern in an affine-invariant fashion. This distinctive representation of each minutiae allows us to compare two minutiae points and determine their similarity irrespective of the global alignment.

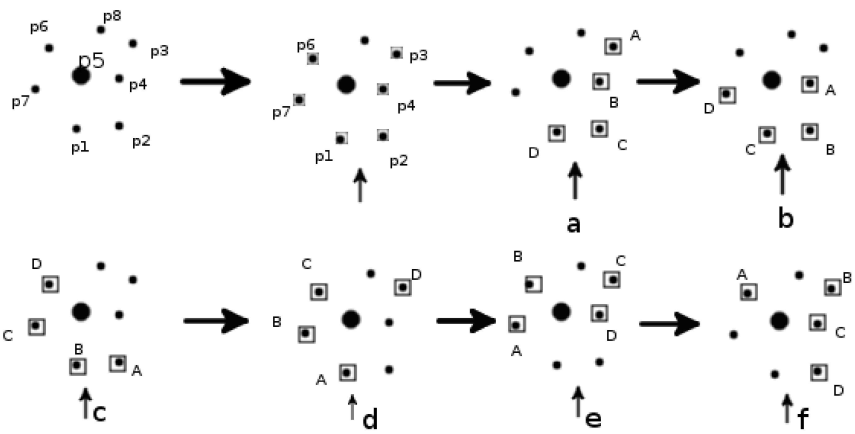



**Figure 3.2** Process of creating the Arrangement Vector for minutia p5. In step1, we find the nearest n minutiae of p5 (n=7). In step2, we take all combinations of m points out of those n points. In subsequent steps, we take four points A,B,C,D and calculate invariants a,b,c,d,e,f (see Section 3.3.1). *abcdef* is the required vector that describes the arrangement of p3,p4,p2,p1,p7 and p6 around p5.

### 3.3.1 Process of Calculating Arrangement Vector

The process of calculating the arrangement vector for a minutia *p5*, shown in Figure 3.2, is as follows:

- We calculate the nearest $n$ neighbors of minutia *p5*. In Figure 3.2, let $n = 7$, and the nearest minutiae are p1,p2,p3,p4,p6, and p7. We then enumerate all combinations of m points of the above $n$ ($\binom{n}{m}$ combinations).
- For each combination, we arrange the $m$ points in clockwise order. Now, we describe the local geometry of these $m$ points around the minutia p5. As shown in Figure 3.2, let m=6, and let p3, p4, p2, p1, p7 and p6 be the m minutiae arranged in clockwise order. With four points denoted as A, B, C, D, we calculate the following invariant features for indexing :

**Ratio of Areas** : The first feature $\varphi$ is the ratio of the areas of the triangles formed by minutiae triplets $A, B, C$ and $A, B, D$.

**Ratio of Lengths of Largest Side** : The second feature $\lambda$ is the ratio of the lengths of the largest side of the triangles formed by minutiae triplets $A, B, C$ and $A, C, D$.

**Ratio of median and minimum angles** : The third and fourth features $\alpha_1$ and $\alpha_2$ are the ratios of the median and minimum angles of the triangles formed by minutiae triplets $A, B, C$ and $A, C, D$.

These features are invariant to affine transformations [7] and remain unchanged even when the fingerprint is translated, rotated, scaled or sheared. A weighted combination of these features is computed to get one final invariant value that describes the local arrangement of these $m$ points. By sliding the points to regard A, B, C and D in clockwise rotation, m such invariants are calculated (i.e a,b,c,d,e and f in Figure 3.2). Thus, *abcdef* represents an arrangement vector for minutia p5.

### 3.3.2 Enrolling a Fingerprint

The above vector depends on the initial choice of A, B, C and D points and is not rotation invariant. To achieve rotation invariance, we use cyclic permutations of this vector. Cyclic permutation of these m invariants give us m vectors (i.e *abcdef, bcdefa, cdefab, defabc, efabcd, fabcde)*. Each vector is considered for hashing and a hash value is calculated from it by Equation 3.1. In the equation, *v* is the $m$ length arrangement vector, *H_size* is the size of the hash table and *k* is the level of quantization of the invariant. This means that the quantized value is in the range [0,k]. The minutia ID, fingerprint ID along with the arrangement vector is stored in the corresponding hash bin. Separate Chaining technique is applied to resolve collisions that occur when two vectors map to the same hash bin. Summary of the offline Enrollment stage is shown in Algorithm 1. The complete enrollment pipeline is shown in Figure 3.3.

$$H_{index} = \left( \sum_{i=1}^{m} v[i] \cdot k^i \right) mod\ H_{size} \tag{3.1}$$

**Algorithm 1** Enrollment Algorithm

INPUT $\rightarrow$ Entire Fingerprint Database db, n, m, k
OUTPUT $\rightarrow$ Model Hash Table
**for all** fingerprint image *fp* in *db* **do**
  **for all** minutia $p$ in *fp* **do**
    N $\rightarrow$ nearest n neighbors of minutia p
    L $\rightarrow$ list of all possible combinations of m points
    **for all** combination of m points in L **do**
      find arrangement vector *v*
      C $\rightarrow$ list of all cyclic permutations of *v*
      **for all** vector $v'$ in list C **do**
        calculate $H_{index}$ from *v'* using eq.1
        register item (Fingerprint ID of *fp*, Minutia ID of $p$, Arrangement vector *v'* ) using $H_{index}$
      **end for**
    **end for**
  **end for**
**end for**

### 3.3.3 Querying the Index

For each minutia $p'$ in a query image and for each combination of $m$ points around that minutia, we calculate its arrangement vector $v''$ as described earlier. The hash value of $v''$ is computed, and the corresponding list of fingerprints that contain a similar minutiae neighborhood is obtained from the hash table. Each minutia in the query fingerprint casts a vote for each fingerprint in its candidate list. Finally, a list of top N fingerprints with the maximum votes is returned as the output of the indexing algorithm. Summary of the on-line indexing stage is given in Algorithm 2. The complete identification pipeline is shown in Figure 3.4.

**Algorithm 2** Indexing Algorithm

INPUT $\rightarrow$ Query image im, n, m, k, N
OUTPUT $\rightarrow$ List of top N fingerprints sorted by number of votes received
**for all** minutia $p'$ in *im* **do**
  N $\rightarrow$ nearest *n* neighbors of minutia $p'$
  L $\rightarrow$ list of all possible combinations of m points
  **for all** combination of m points in L **do**
    find arrangement vector *v"*
    calculate $H_{index}$ from *v"* using eq.1
    lookup Hash Table with $H_{index}$ and retrieve the corresponding list
    **for all** $item$ in the retrieved list **do**
      **if** $v''$==$item$.Arrangement vector **then**
        Increment vote count for FingerprintID corresponding to $item$
      **end if**
    **end for**
  **end for**
**end for**
Sort all fingerprints according to vote counts in descending order
Output list of top N as the indexing result

## 3.4 Experiments and Discussions

The experiments were conducted on the four *FVC 2002* databases: DB1, DB2, DB3 and DB4. Each database contains 800 fingerprints from 100 users (8 impressions per user). For each user, the first 4 impressions were placed in the gallery to build the hash table while the remaining 4 impressions were used as probes. Experiments were conducted with different values of n, m and k. The best results were observed for n=6, m=5 and k=28 and $H_{size} = 1000000$. Accuracy and efficiency are two main indicators of the retrieval performance. In the experiments, the accuracy is denoted by *Correct Index Power (CIP)* where CIP = $(N_{ci}/N_d)$, $N_{ci}$ is the number of correctly indexed probe images, and $N_d$ is the number of images in the database. The retrieval efficiency is indicated by the *Penetration Rate*, which is the average percentage of database probed over all test fingerprints. Ideally, we would want a high CIP and a low Penetration Rate.

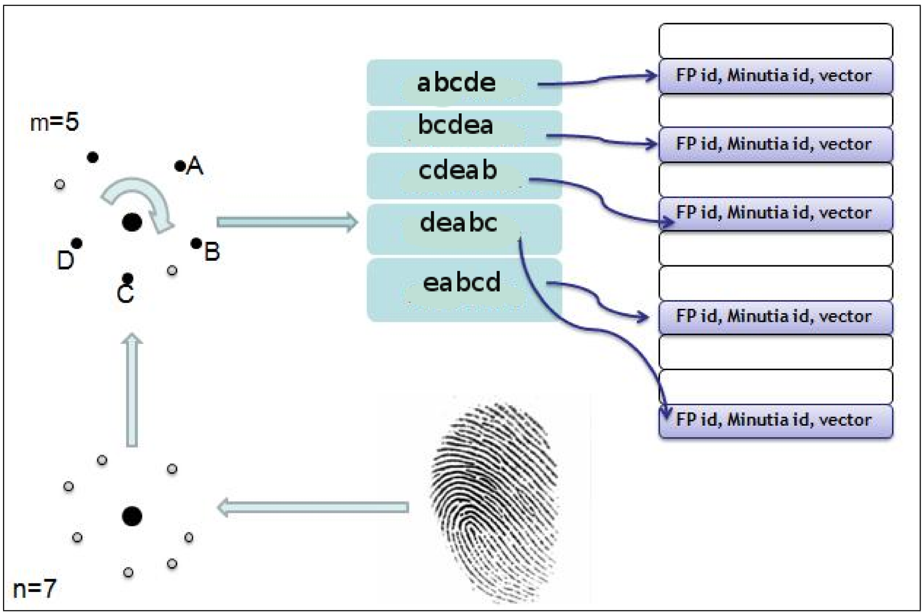



**Figure 3.3** Process of enrolling a user in the hash table.

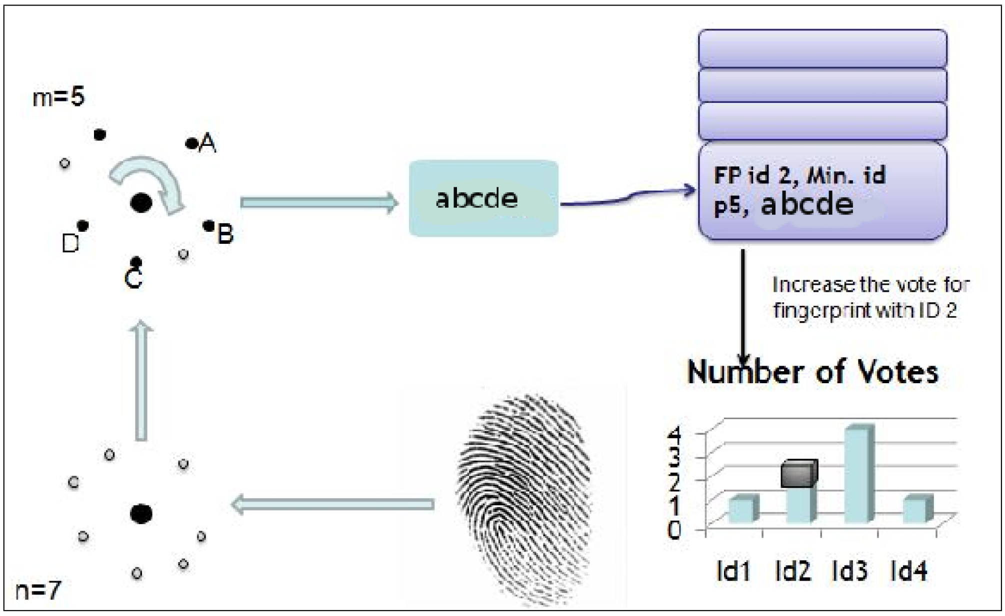
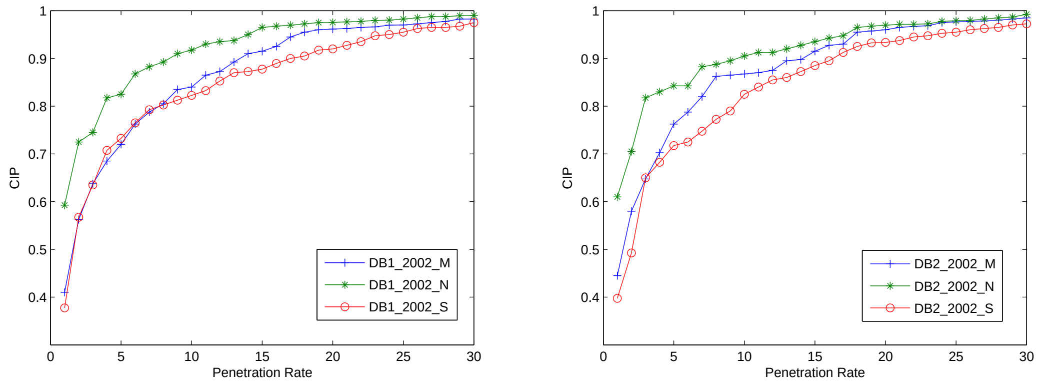



**Figure 3.4** Process of identifying a user given a query fingerprint image.




(a) (b)

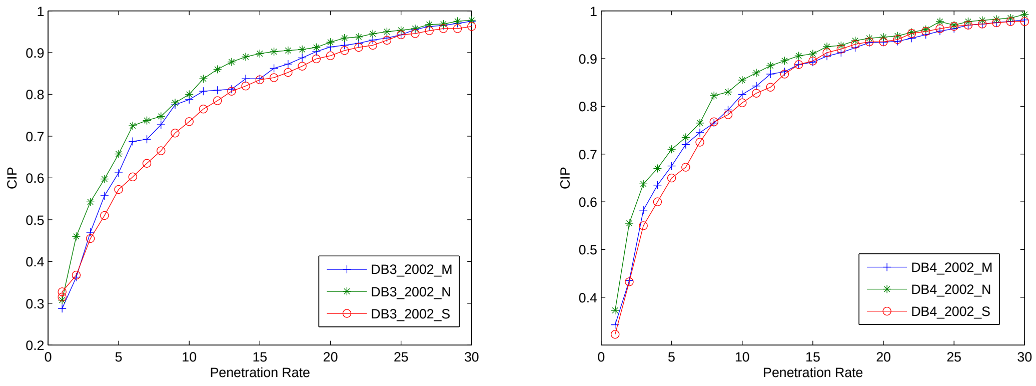
**Figure 3.5** Results on FVC 2002 databases in case of 20% missing minutiae data(M), original minutiae (N) and 20% spurious minutiae (S). (a) For FVC 2002 DB1 (b) For FVC 2002 DB2.

### 3.4.1 Dealing with missing and spurious minutiae

Handling the case of missing or spurious minutiae is a major challenge for minutiae based indexing techniques [43], [55], [23], [7]. We deal with this problem by first choosing nearest *n* neighbors for a minutia and then out of these *n*, we choose all possible combinations of *m* minutiae points. This rule of ***choosing m out of n neighbors*** helps us to deal with missing and spurious minutia. Experiments were done with datasets having 20% spurious minutiae and 20% missing minutiae. Minutiae were removed and added randomly to the database. For each experiment, the following three cases were considered: datasets in their original form; datasets with 20% spurious minutiae; and datasets with 20% missing minutiae. As the plots show (Figures 3.5,3.6), even removing or adding 20% extra minutiae did not affect the low penetration rates at the hit rate of greater than 97%. This shows that the scheme is able to handle low quality noisy images, where there are lots of missing or spurious minutiae points.

### 3.4.2 Dealing with Distortions

Handling the case of non-linear distortions and transformations is a major challenge for indexing algorithms that use global features [50] and [53]. We deal with this problem by using features, like ratio of sides, angles and areas, which are invariant to geometric transformations like rotation, translation, scaling and shear [7]. It is known that non-linear distortion happening in local minutiae structures is small enough to be ignored compared with the much larger global non-linear distortion [58]. The arrangement vector, the proposed local minutia structure, can tolerate non-linear distortion as indicated by the experimental results.




(a) (b)

**Figure 3.6** Results on FVC 2002 databases in case of 20% missing minutiae data(M), original minutiae (N) and 20% spurious minutiae (S). (a) For FVC 2002 DB3 (b) For FVC 2002 DB4.

| **Hit Rate** | **Minutiae Quadruplets** [36] | **Proposed Algo.** |
|---|---|---|
| 60% | 6.8 | **1** |
| 70% | 8.68 | **1.9** |
| 80% | 10.5 | **3.9** |
| 90% | 15 | **8.6** |
| 95% | 17.6 | **14** |
| 100% | **21.5** | 57 |

**Table 3.1** Average penetration rates using the proposed and quadruplets [36] approaches at various Hit Rates. The database used was FVC 2002 DB1.

| **Hit Rate** | **Minutiae Quadruplets** [36] | **Proposed Algo.** |
|---|---|---|
| 60% | 6.31 | **2.8** |
| 70% | 8.15 | **4.14** |
| 80% | 11.8 | **8** |
| 90% | 17.89 | **13.5** |
| 95% | 22.89 | **17.5** |
| 100% | **27.89** | 60 |

**Table 3.2** Average penetration rates using the proposed and quadruplets [36] approaches at various Hit Rates. The database used was FVC 2002 DB1 with 20% minutiae removed randomly.

## 3.5 Comparison of Results

The proposed algorithm is compared with the quadruplet based indexing algorithm in [36], which is also a minutiae based indexing algorithm. We have selected this work for comparison as, to the best of our knowledge, it has the highest accuracy among the affine invariant representations that have been proposed till date. The Minutiae Cylinder Code uses a richer representation of the minutiae neighborhoods and performs better in practice. However, their features have only translation invariance and rotation invariance is achieved by orienting the cube using minutiae orientation. Other invariances are not considered. While explicit invariances are not necessary for practical systems (as indicated by MCC results), we prefer to use them as they provide a theoretically sound basis for future analysis of errors.

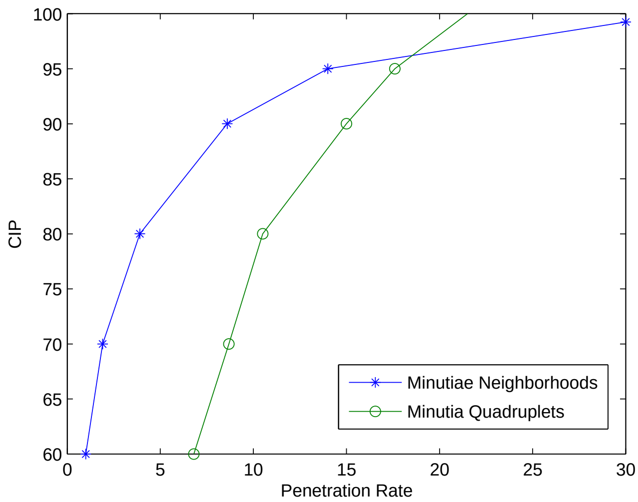
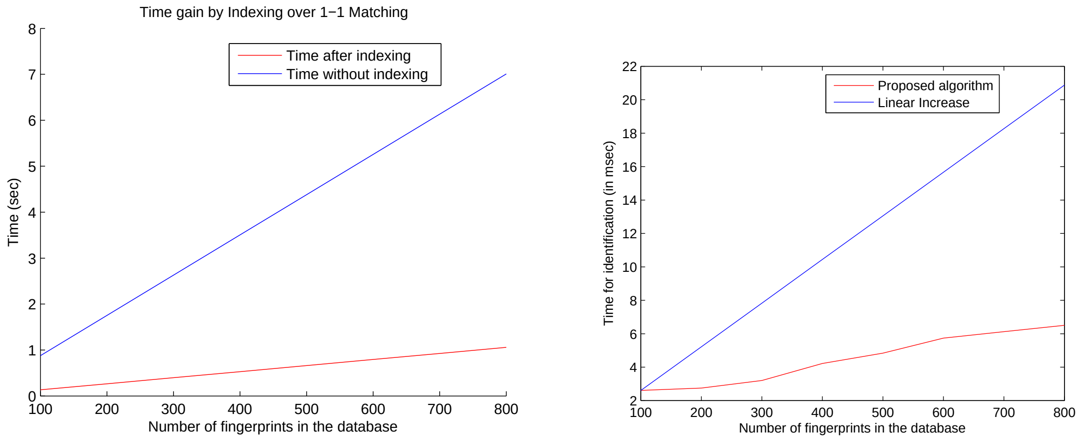



**Figure 3.7** Comparison of the proposed and quadruplets [36] approaches on FVC 2002 DB1.

The evaluation protocol was based on [36], which uses Hit Rate as a measure of correct indexing. Hit Rate is defined as CIP*100 in percentage. As the results in Table 3.1 and Table 3.2 show, the Hit Rate of the proposed algorithm is better than that reported in [36] at lower penetration rates. Also, our algorithm handles missing minutiae points better than the one proposed by Ross in [36]. However, the quadruplet based algorithm performs better than ours for 100% Hit Rate. Figure 3.7 compares the Hit Rate vs Penetration graphs of both the algorithms. The tests were run on FVC 2002 DB1 database.

## 3.6 Time Analysis

The major benefit of using a fingerprint indexing algorithm is that it reduces the number of expensive one-to-one matches resulting in significant reduction in the overall time for identification. To find out the reduction in time we get by applying our indexing algorithm, we carried out experiments with the FVC 2002 Db1 database. To find the time required for each identification test, each image in DB1 was




(a) (b)

**Figure 3.8** Time analysis done for FVC 2002 DB1. (a) Shows the gain in time obtained by indexing over 1:1 matching with entire database. (b) Shows the sub-linear scaling of our proposed algorithm with the increase in database size.

used as a probe for identification against a gallery of other fingerprints. The gallery size was kept at 100, 200, 300 and increased till 800. This helped us to check how our algorithm scaled with increase in size of the database. This is a very critical factor, as any good indexing algorithm should scale well when the database size is increased. A probe image is considered to be correctly classified, if at least one impression from its class is present in the list returned by indexing algorithm.

### 3.6.1 Time Gained by Indexing

In the first experiment, we observed the time we gained by using our indexing algorithm. Given a probe image, we first calculate the time required for performing one to one comparisons with the entire database. The comparisons were made using the FBI approved *nfis* matching algorithm [31] which takes **8.76 msec** for one matching. Then we applied our indexing algorithm to get a top $N$ list and matched the probe with the top N list only (for 97% CIP). We calculated the time required for this task. We compared both the times for a database of size 100, 200, 300, 400, 500, 600, 700 and 800 fingerprint images. We observed a significant gain in time when we apply our fingerprint indexing algorithm over the naive entire one-to-one matching. The results are plotted in Figure 3.8 and the time values are shown in Table 3.3.

| Size of the database | Time without Indexing | Time with Indexing |
|---|---|---|
| 100 | 0.876 | **0.134** |
| 200 | 1.752 | **0.265** |
| 300 | 2.628 | **0.397** |
| 400 | 3.504 | **0.529** |
| 500 | 4.380 | **0.661** |
| 600 | 5.256 | **0.794** |
| 700 | 6.132 | **0.925** |
| 800 | 7.008 | **1.057** |

**Table 3.3** Showing the time benefit (in sec.) we obtain on indexing the database over 1:1 matching without indexing

### 3.6.2 Scalability of the proposed algorithm

In the second experiment, we observed the time required by our indexing algorithm to output the top *N* list sorted by votes. We experimented with different database sizes and noted down the time taken. We observed that the time increase was sub-linear with the increase in database size as shown in Figure 3.8.

## 3.7 Summary

In this chapter, we tackled the problem of searching through large fingerprint databases known as fingerprint indexing problem. Then an extensive literature survey was done on the existing fingerprint indexing methods. These methods were classified according to the features they use and we discussed the weaknesses present in this method. Then we proposed a new representation for a minutia point using only the locations of the points and no other high level features such as orientation flow and directional field were used. This makes the proposed approach applicable to a wide variety of existing templates. Then we showed results of experiments on FVC 2002 databases. We conducted experiments with missing or spurious minutiae points and showed that our algorithm was robust to appearance and disappearence of points. Then we compared our results with the best indexing method in the literature using similar features. Our approach led to better results and was more tolerant towards missing minutiae. Then finally, we did a detailed time analysis of our algorithm and showed that our algorithm scales sub-linearly with the increase in database size. Now, in the next chapter we extend our proposed minutiae representation for constructing a fixed length representation of a fingerprint image. Then we used the proposed fixed length representation for fingerprint matching purpose.

*Chapter 4*

# Learning Minutiae Neighborhoods: A New Binary Fingerprint Representation

In the last chapter, we proposed a novel local minutiae based nearest neighbor structure called *Arrangement Vector* that describes the geometric layout of neighboring points around a central minutia point. We used that structure for a hash based fingerprint indexing method and showed the usefulness of the structure. In this chapter, we will extend that structure to handle the problem of matching two fingerprints. The problem, with existing solutions and major challenges, was described in Chapter-2. The technique proposed in this chapter, relies on the creation of high dimensional structure space based on minutiae neighborhoods. The k-means clustering technique is then applied to partition this structure space into multiple clusters. The center of these clusters represent the learned neighborhoods from the entire dataset. Fingerprints are then visualized as collection of these neighborhoods and a fixed length binary representation for the fingerprint is then generated. The matching of two is then reduced to simple computation of the hamming distance between the two binary vectors representing the fingerprints. The experiments performed on the FVC 2002 and FVC 2004 databases show the effectiveness of the proposed approach.

## 4.1 Representing a Fingerprint

Although what we get from a fingerprint sensor is usually a grayscale image of some resolution, only a few fingerprint matching or indexing algorithms work directly on the grayscale image. Before the matching stage, most of the algorithms have a pre-processing or a feature extraction stage where useful information is extracted from the fingerprint. And then this information is used, instead of directly superimposing or corelating the two grayscale images. Based on features extracted and stored, the traditional fingerprint representation schemes can be classified as :

- **Global Features based Representation** : The global approach to fingerprint representation is typically used for fingerprint indexing. These representations include global ridge-line frequency, core points, orientation images and singular points. These features represent the global pattern of

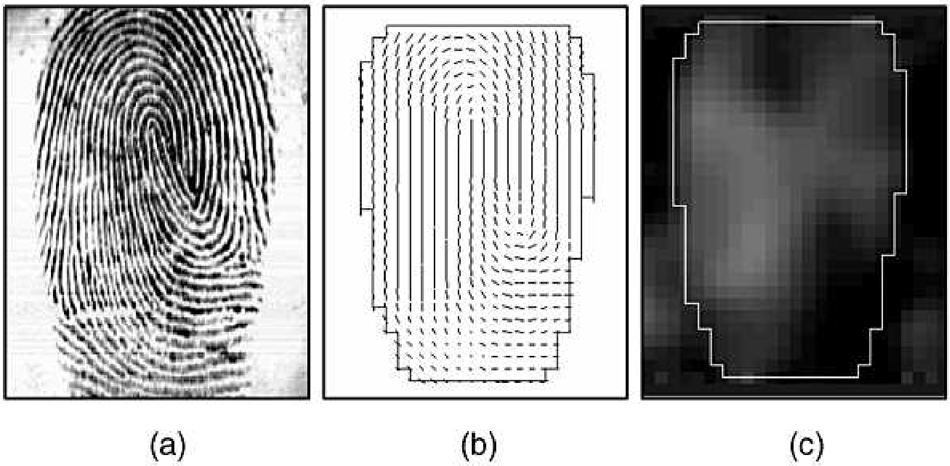



**Figure 4.1** Global Representations of a fingerprint. (a) Original grayscale image (b) Corresponding orientation image (c) Corresponding frequency map where lighter regions correspond to higher ridge frequency.

the ridges in the fingerprint. An example of orientation image and the corresponding frequency map is shown in Figure 4.1. One disadvantage of these representations is that they cannot be easily extracted from poor quality fingerprints. Also, these representations do not offer good individual discrimination and are not good at handling distortions. Further, the indexing efficiency of existing global representations is not very good due to a small number of categories that can be effectively identified and a highly skewed distribution of the population in each category.

- **Local Features based Representation** : The local approach refers to representing the fingerprint in the terms of minutiae sets, local ridge orientations and local ridge frequency. Ross in his work [2], uses representative local fingerprint patterns to construct a feature vector. These local representations have evolved from intuitive system design geared for fingerprint experts who visually match fingerprints. These local representations are quite distinctive and generally outperform their global counterparts. Minutiae based representations are most commonly used as they are compliant with most of the existing fingerprint suppliers and databases. One disadvantage of such representations is that they suffer from misalignment problem and require a preliminary registration step. Also, minutiae extraction is not that simple and requires stages such as binarization, thinning of the grayscale image and post processing to remove false minutiae.

- **Combination of Local and Global** : Certain schemes have been proposed which combine the local and global information present in a fingerprint. Fingercode [19] proposed by Jain, utilizes both local and global ridge descriptors and texture information. The features are extracted by measuring the responses of radial image sectors to a gabor filterbank. Sha [30] proposed a better version of fingercode as described in Section 2. Benhammadi [12] also proposed a new representation called oriented minutiae codes based on minutiae texture maps. They use the response of eight gabor filters to generate the codes.

Existing representations are surveyed in Section 2. Depending on the length of the feature vector constructed, the traditional fingerprint representation schemes could also divided into *fixed-length* and *variable-length* representation schemes.

### 4.1.1 Fixed length representation

The length of such representations are independent of the number of minutiae points present in the image. Generally, in this scheme each fingerprint in the database is represented by a fixed length (usually binary) feature vector. Various methods have been proposed to transform a fingerprint image (or a minutiae set) into a fixed-length quantized feature vector. Fingercode [19] encodes the local and global texture around the core of the fingerprint. Tuyls in his work [37] proposed a novel quantization algorithm to get fixed length representation based on local orientation of ridges. Xu [18] explained the construction of a feature vector of floats via the spectral representation of a minutiae set. Julien Bringer in their work [22], transform a minutiae set into a fixed-length quantized feature vector by matching small minutiae vicinities (or neighborhoods) with a set of representative vicinities. One major advantage of fixed length vectors representing fingerprints is that the matching stage becomes very fast. It just reduces to a simple series of binary operations (AND, XOR) or calculating the hamming distance between the two vectors which is very quick. Also, since the length of the vectors is fixed, the vectors are easy to match and most of the times don't require any prior alignment. Also, many template protection schemes like [37] and [3] require a fixed length vector as input. In short, fixed length representations are easier to match and are suitable for the recent biometric template protection schemes.

### 4.1.2 Variable length representation

The most widely used and a classical representation for fingerprints is based on minutiae set which is an unordered set of characteristic points (ridge endings and bifurcations). Fingerprint is stored as a collection of minutiae points. The length of such representations depend upon the density of minutiae points in the fingerprint. A good quality fingerprint contains between 60 to 80 minutiae, but different fingerprints have different number of minutiae. Matching now requires registering of minutiae sets of different sizes which is computationally expensive and tough to intuitively visualize. Also, variable length representations are tough to store on smart cards and are not at all suitable for biometric template protection schemes. And ofcourse, a major challenge when designing such variable length representation is how to deal with the insertion and deletion of minutiae. However, variable length representations are usually more tolerant towards non-linear distortions.

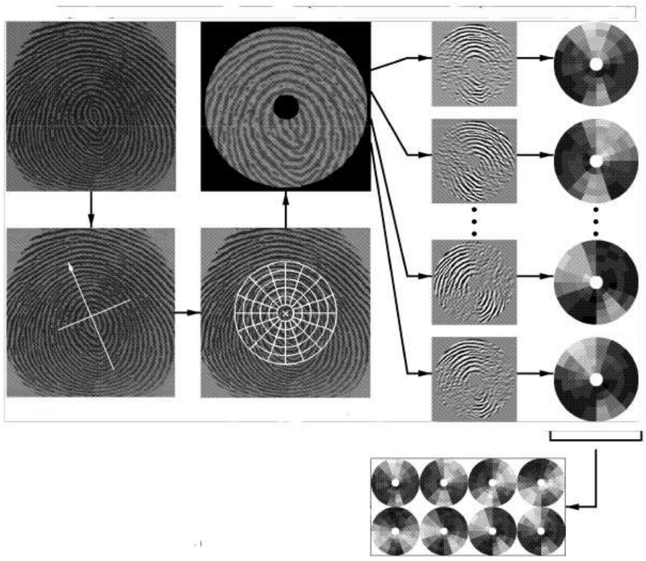


**Figure 4.2** Process of generating Fingercode. First the core point is detected. Then image is tessellated around the core point into sectors. Then each sector is normalized and is filtered with bank of Gabor filters. The response is concatenated to get a fixed length fingercode.

## 4.2 Popular representations and their drawbacks

Many representations have been proposed in the literature. Fingercode[19, 30] is a fixed *640 byte* representation that makes use of both the overall global ridge pattern and the local ridge characteristic. The fingercode is extracted by tessellating the image around the core point. The feature vector consists of an ordered collection of texture descriptors from various sectors of the tessellation. The texture descriptors are obtained by filtering each sector with eight oriented gabor filters and then computing the Average Absolute Deviation of the pixel values in each cell. The features are concatenated to get the fingercode as shown in Figure 4.2. The disadvantage of fingercode is that it requires the core point to be accurately located which in itself is a difficult problem. Tico [34] proposed a *48 byte* length representation using Digital Wavelet Transform (DWT) features. Amornraksa [49] proposed a *24 byte* representation using the Digital Cosine Transform (DCT) features. However, one drawback of transform-based representations is that they are not rotation invariant and rotation has to be handled explicitly. This was handled by Xu in his work [18], in which he proposed a spectral minutiae representation based on Fourier-Melin transform. By representing minutiae as a magnitude spectrum, he transforms a minutiae set into a fixed length feature vector that does not need registration to compensate for translation, rotation and scaling. But still the scheme is not very robust to non-linear distortions and missing/spurious minutiae.

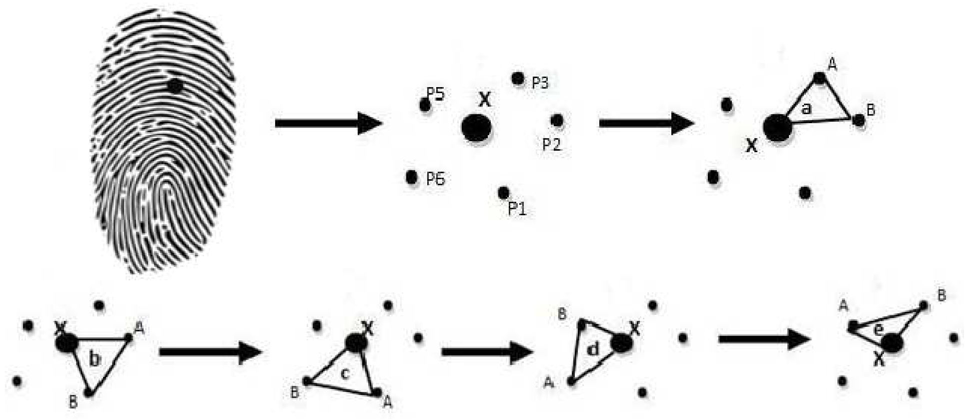



**Figure 4.3** The process of creating the local structure for minutia X. In step1, we find the nearest n minutiae of X. In subsequent steps, we take two points A,B and calculate invariants a,b,c,d,e (see Section 4.3). *a*bcde is the required structure that describes the local neighborhood of central minutia X.

### 4.2.1 Need for new Representation

Most of the fixed length representations described above either cannot handle global transformations like rotation and translation or are not tolerant towards small local non-linear distortions. This implies that the accuracy of matching using the quantized feature vector representations still is very low as compared to classical minutiae based matching. Also, most of these schemes find it very tough to handle missing/spurious minutiae points. We really require a fixed length (binary prefered) representation that is tolerant towards these distortions, can handle missing/spurious minutiae, is suitable for template protection schemes, small enough to be stored on smart cards and has a minutiae-only construction so that it can be applied to existing databases. In the next section we propose a new local minutiae structure that captures the complete geometry of neighboring points around a central minutia. This local structure is an extension of *Arrangement Vector*(defined in the last chapter). We have added relative orientation information into the Arrangement Vector to make it more robust. We then use *k-means* clustering to cluster this high dimensional space of local structures. From this we get *k* cluster centers, which correspond to the *k* most prominent neighborhood structures learned from the fingerprint database. Then every fingerprint in the database is expressed as a collection of these cluster centers to get a fixed-length (of length *k*) representation for a fingerprint.

## 4.3 Proposed Local Structure

Our local structure is a fixed-length descriptor for a minutia that captures the geometry formed by its neighboring points around that minutia. Such geometry is quite distinctive for a particular minutia and

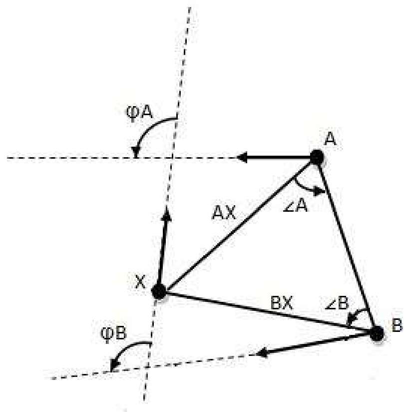



**Figure 4.4** The geometric features computed from $\Delta$AXB. Relative distances AX and BX, Relative Orientations $\phi_A$ and $\phi_B$ and angles $\angle$B and $\angle$A

remains the same even when the image is transformed. This distinctive representation of each minutiae allows us to compare two minutiae points and determine their similarity.

The process of calculating the local structure for a minutia (X), shown in Figure 4.3, is as follows:

- We calculate the nearest $n$ neighbors of minutia X based on their euclidean distances from X. In Figure 1, let $n = 5$, and the nearest minutiae are p1,p2,p3,p5 and p6.

- We arrange the $n$ points in clockwise order. This is because the clockwise order of minutiae points remains unchanged even when the fingerprint image is rotated, translated, scaled or sheared.

- Now, we describe the local geometry of these $n$ points around the minutia X. As shown in Figure 4.3, let n=5, and let p3, p2, p1, p6 and p5 be the n minutiae arranged in clockwise order. With two points marked as A, B we calculate the following geometric features from $\Delta$AXB as shown in Figure 4.4:

  **Relative Distances** : We calculate the euclidean distances between points X and A,B. The first feature is the ratio of these relative distances.

  **Relative Orientation** : We calculate the orientations of points A,B with respect to the central minutia X (relative orientation of A is the $\phi_A$ - $\phi_X$, where $\phi_A$ is the orientation of minutia A). The second feature is the ratio of these relative orientations.

  **Angles of $\Delta$AXB** : The next features we use the angles $\angle$XBA and $\angle$XAB of the $\Delta$AXB. The third feature is the ratio of these angles.

- These features are provably invariant to geometric distortions [7] and remain unchanged even when the fingerprint is translated, rotated, scaled or sheared.

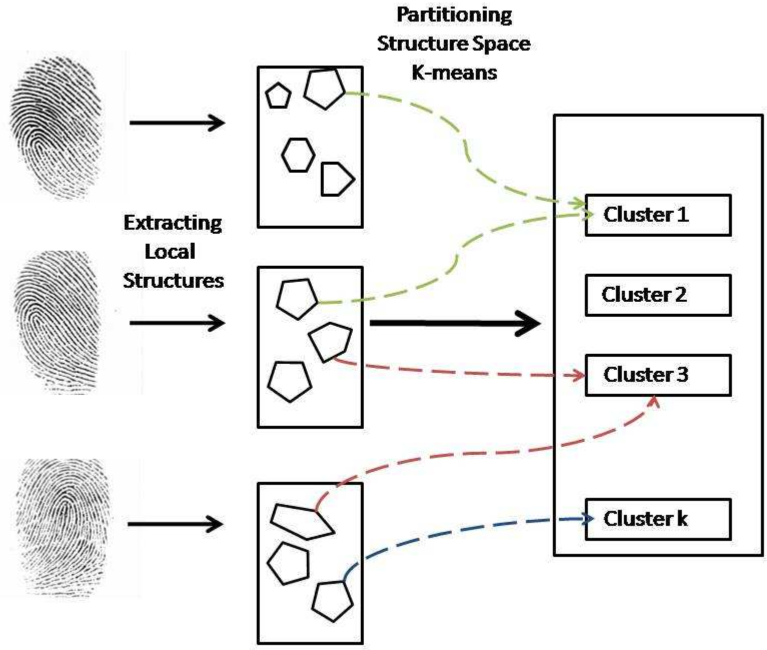



**Figure 4.5** Populating the *n*-dimensional structure space. The local structures are extracted from each fingerprint in the database. Then the structure space is partitioned into *K* clusters via the k-means algorithm.

A simple average of these features is computed to get one final invariant value (*a* as shown in figure 4.3) that describes contribution of $\Delta$AXB in the arrangement of these $n$ points around the minutia X. By sliding the points A, B in clockwise rotation, $n$ such invariants are calculated ( i.e a,b,c,d and e in Figure 4.3). Thus *abcde* is the local structure of length *n* that describes the geometric layout of these n points around our central minutia X. The structure *abcde* depends upon the initial choice of points A, B and is not invariant to rotations. To achieve rotation invariance, we use cyclic permutations of this structure. All *n* cyclic permutations of *abcde* (i.e *bcdea*, *cdeab*, *deabc*, *eabcd* and *abcde*) are calculated and stored in a list. So now we have many $n$ dimensional feature vectors, where each vector represents a minutiae neighborhood. Now we use *k-means* to cluster this n-dimensional space as shown in Figure 4.5.

### 4.3.1 Partitioning the Structure Space and Representing a Fingerprint

Each fingerprint now can be represented as a set of minutiae neighborhoods. Each neighborhood is generated from only minutiae points and is characterized by a *n* length feature vector. This n-dimensional feature vector can be viewed as a single point in n-dimensional hyperspace. Thus, each finger will have a collection of points (pertaining to all neighborhoods it contains) residing in this hyperspace. Given a set of training fingerprint images, an unsupervised learning algorithm *K-means* is

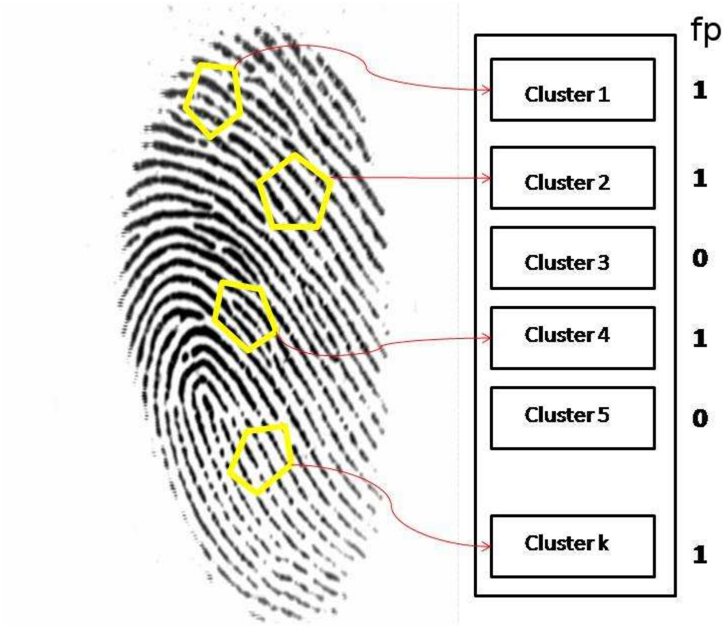



**Figure 4.6** A fingerprint image represented in terms of representative neighborhoods. Given a image, we extract all the neighborhoods and map them to the nearest cluster. *fp* is the *K* length binary representation of the fingerprint.

then applied to cluster this hyperspace. This results in *K* clusters $c_1$,$c_2$,$c_3$......$c_K$ where each cluster represents set of similar neighborhoods. The centroid of each cluster $c_j$, represented by $m_j$ can be seen as the mean representative neighborhood for that set of neighborhoods that map to $c_j$. So, in essence, $m_1$,$m_2$,$m_3$.......$m_K$ are the most prominent neighborhoods learned by our algorithm. Any fingerprint now can be represented in terms of these representative neighborhoods. When a new fingerprint comes, we extract all the neighborhoods from that and map each neighborhood feature vector to its nearest cluster center as shown in Figure 4.6. So, now each fingerprint is a binary feature vector **fp** of length *K* where $fp_i$ tells whether a neighborhood similar to $m_i$ is present in the fingerprint or not. So, we now visualize fingerprints as a collection of neighborhoods rather than a grayscale image or minutiae sets.

## 4.4 Fingerprint Similarity Measure

Now given two binary vectors *fp1* and *fp2*, representing the two fingerprints, a formula based on simple bitwise operations on the two vectors will give a measure of number of similar neighborhoods present in them. Thus, simple bit-oriented coding can now be used as a measure for fingerprint similarity. Similarity *s* between two binary vectors, *fp1* and *fp2* is calculated by using 3 different metrics.

The first metric used is the *L2-norm* of the XOR of the two vectors . *L2-norm* is the square root of the number of one bits in the vector. The second metric used is the *L0-norm* of the XOR of the two vectors. *L0-norm* gives the number of one bits in the vector. The third metric used is *L0-norm* of the bitwise AND of the two vectors. Equation 4.1 gives the XOR similarity between the two vectors, where $||fp||$ is L2 norm for *xor-L2* metric measure and L0 norm for *xor-L0* metric measure. Equation 4.2 gives the AND similarity used for *and* metric measure. A test run on FVC 2002 db1 showed that L2 XOR (*xor-L2*) similarity measure gave the best results (Figure 4.7) and was used for remaining experiments.

$$s(fp1, fp2) = 1 - (\|fp1\ XOR\ fp2\|)/(\|fp1\| + \|fp2\|) \tag{4.1}$$

$$s(fp1, fp2) = (\|fp1\ AND\ fp2\|)/(\|fp1\| + \|fp2\|) \tag{4.2}$$

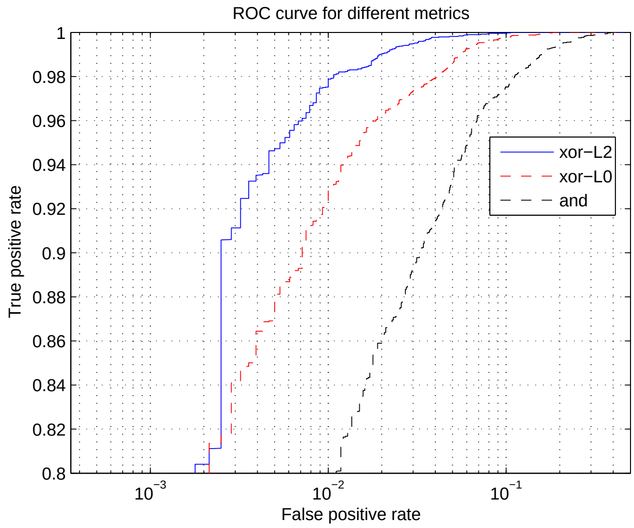



**Figure 4.7** ROC curve for comparing the accuracy achieved by the three different metrics. *xor-norm* is the L2 norm of the XOR. *xor* is the L0 norm of the XOR and *and* is the L0 norm of the AND of the two vectors.

## 4.5 Experiments and Results

Experiments were conducted on FVC 2002 db1, db2, db3 and FVC 2004 db1 and db2 databases. Each database consists of 800 impressions from 100 different fingers, 8 impressions per finger. The minutiae were extracted using the standard NIST MINDTCT algorithm[14]. The performance evaluation protocol used in FVC 2002 (same as in [9]) has been adopted. Experiments were done for different values of *k* and *n*. The best results were obtained for cluster size of 1000 (i.e k=1000) and neighborhood size of 5 (i.e n=5). A total of 14,000 genuine matches (2800 per database) and 24,750 imposter matches

(4950 per database) were done. The ROC curves with different number of clusters have been plotted below. It was observed that the accuracy increased with increase in number of clusters upto an extend and then it started decreasing gradually after 1000 clusters as shown in the plot below. The ROC curves and the genuine-imposter class distribution curves for the FVC datasets are shown on the next page (refer to Figures 4.10, 4.11 and 4.12). The results have been compared with spectral minutiae representation [18] and binary representation through minutiae vicinities [22] (see Figure 4.9) . These are the two major fixed-length quantized fingerprint representations in the literature.

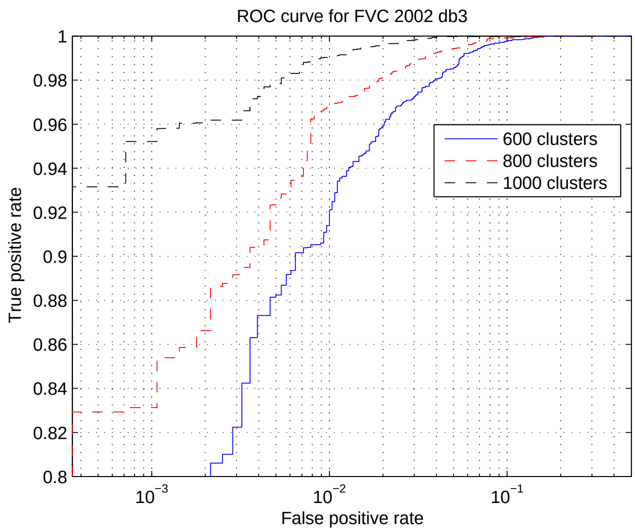



**Figure 4.8** ROC curve for FVC2002 db3 database showing the increase in accuracy with the increase in number of clusters.

## 4.6 Summary

We proposed a novel binary fixed-length representation for a fingerprint constructed from minutiae-only features. We captured the local geometry around a minutia point into our local arrangement structure. We then applied unsupervised learning to learn prominent minutiae neighborhoods from the database. A fingerprint was then represented as a collection of neighborhoods resulting in a fixed 1000-length binary representation. The matching of two fingerprints is then reduced to a sequence of bitwise operation which is very quick. Experiments conducted of FVC 2002 and 2004 databases showed the effectiveness of our representation as compared with the major fingerprint representations existing in the literature. Our representation is tolerant towards distortions, can be stored easily on light architectures such as smart cards and is suitable for biometric template protection schemes.

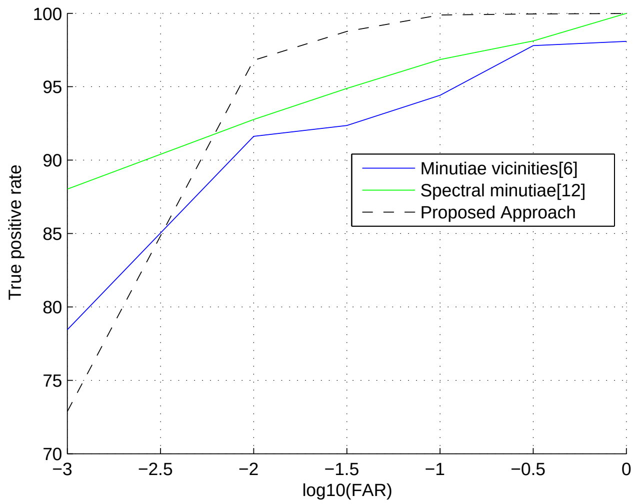
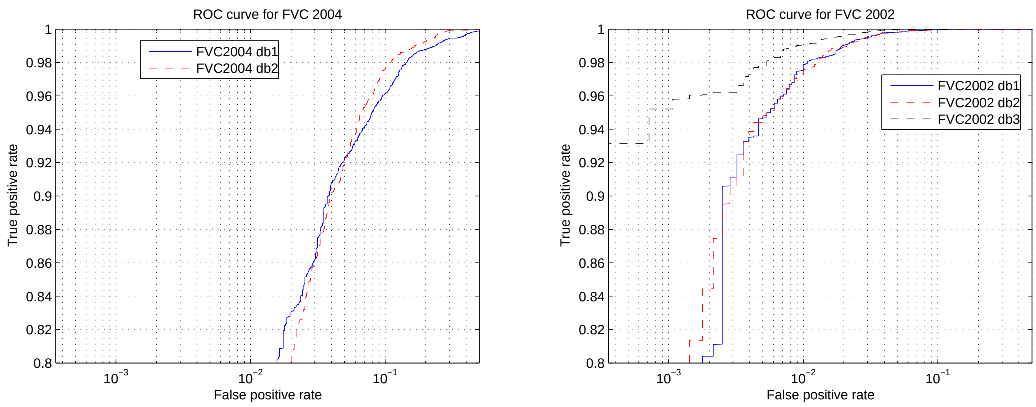


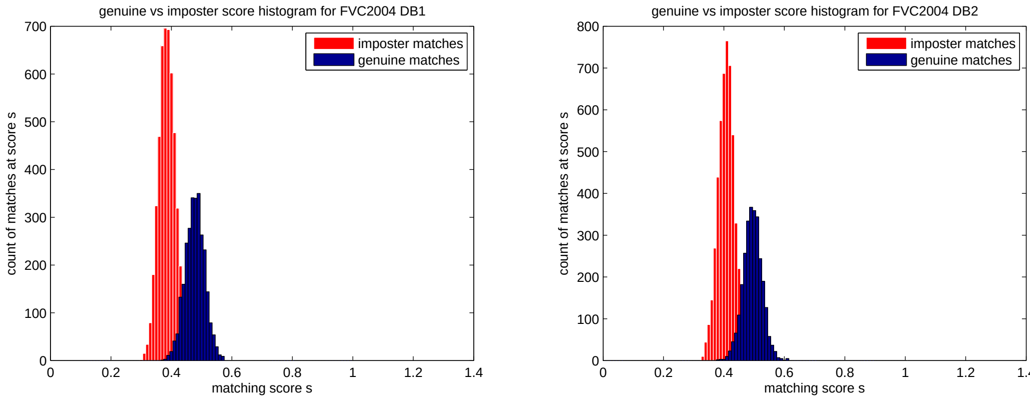

**Figure 4.9** Comparison of the proposed approach with spectral representation[18] and minutiae vicinities [22]. The comparison is done with proposed representation based on 1000 clusters on FVC 2002 db2.




(a)

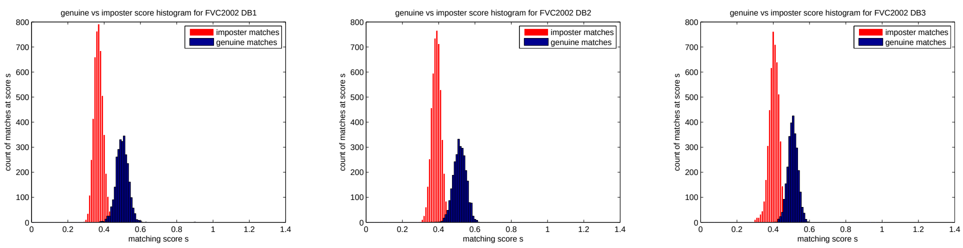
(b)

**Figure 4.10** ROC curves for FVC databases. (a) ROC curve for FVC 2004 db1,db2. (b) ROC curve for FVC 2002 db1,db2,db3.




(a) (b)

**Figure 4.11** Genuine and imposter class distribution curves for FVC 2004 databases.




(a) (b) (c)

**Figure 4.12** Genuine and imposter class distribution curves for FVC 2002 databases.

# Chapter 5

## Conclusions and Future Work

The major problem we try to solve is the use of local geometric information present around a minutia point for the purpose of fingerprint indexing and matching. Many existing methods also use such information in the vicinity of a minutia point for the purpose of indexing. However, they also use extra information such as orientation flow and core points. We use only the spatial location of neighboring points to come up with a new local minutiae structure called an *Arrangement Vector*. This allows us to maintain compatibility with all possible templates, while keeping the storage requirements to a minimum. The arrangement vector is invariant to distortions and can handle the partial overlap problem of fingerprint acquisition. We use this structure to come up with a hash based indexing algorithm to speed up large scale fingerprint retrieval. We propose the **choose m out of n rule** to handle missing or spurious minutiae. Then we extend the arrangement vector by adding minutiae orientation information to it. We include more robust features to the arrangement vector and use it to solve the problem of matching of two fingerprints. In the end, we apply unsupervised clustering to find the common minutiae neighborhoods. We represent each fingerprint as a collection of neighborhoods rather than a grayscale image or a minutiae set. This results in a fixed length binary representation of the fingerprint, which is then used for matching purposes. Experiments were carried out on the publicly available datasets from FVC 2002 and FVC 2004 databases for comparison with existing methods.

The goal of any representation is to capture as much of the distinctive information available in a fingerprint, while discarding the variations between multiple impressions of the same finger. The experiments we conducted showed that our representation was able to capture most of the geometric details around minutiae points and is invariant to distortions. The matching is then reduced to bitwise operations between the vectors representing the two fingerprints. We compared our results with two major quantized fingerprint representations [18, 22] from the literature and demonstrated our method to be more accurate in most practical situations.

One way in which we could extend our work is to improve the robustness and distinctiveness of the arrangement vector. Possible features that can be included for this purpose include ridge count between neighboring minutiae, relative minutiae orientation with respect to X-axis, type of minutia, and quadruplet features[36]. Another possible extension is to use deep learning algorithms to learn what a fingerprint really is. For this approach, we require massive amounts of labeled fingerprint data, which is becoming available with large scale identification systems. Also, we could use our minutia representation in conjunction with similar representations such as MCC and k-plet. This combined representation could outperform any of the single representations. We could also use our proposed binary representation for a fingerprint for the purpose of template protection and use it in the research area of biometric cryptosystems.

# Related Publications

- Akhil Vij and Anoop M. Namboodiri, “Fingerprint Indexing Based on Local Arrangements of Minutiae Neighborhoods”, in Proceedings of the IEEE Compter Society Conference Computer Vision and Pattern Recognition Workshops (CVPRW) 2012
- Akhil Vij and Anoop M. Namboodiri, “Learning Minutiae Neighborhoods : A New Binary Representation for Matching Fingerprints”, in Asian Conference on Pattern Recognition, ACPR 2013 (under review)

# Bibliography


[1] I. 19794-2:2005. Information technology–biometric data interchange formats–part2: Finger minutiae data. 2005.

[2] A.Gyaourova and A.Ross. A novel coding scheme for indexing fingerprint patterns. *Proc. Intl. Workshop on Statistical Pattern Recognition*, 2008.

[3] A.Jeuls, P.Tuyls, T.Kevenaar, and B.Skoric. Fuzzy commitment. *Security with Noisy Data-On Private Biometrics, Secure Key Storage and Anti-Counterfeiting*, pages 45–56, 2007.

[4] A.K.Jain, S.Prabhakar, and L.Hong. A multichannel approach to fingerprint classification. *IEEE TPAMI*, 21(4):348–359, 1999.

[5] A.Ross and R.Mukherjee. Augmenting ridge curves with minutiae triplets for fingerprint indexing. *SPIE Conference on Biometric Technology for Human Identification*, 2007.

[6] A.Ross, S.Dass, and A.K.Jain. A deformable model for fingerprint matching. *Pattern Recognition*, 38(1):95–103, 2005.

[7] B.Bhanu and X.Tan. Fingerprint indexing based on novel features of minutiae triplet. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 25:616–622, 2003.

[8] D.Kwon, I.D.Yun, D.H.Kim, and S.U.Lee. Fingerprint matching method using minutiae clustering and warping. *Proc. Int'l Conf. Pattern Recognition*, 4, 2006.

[9] D.Maio, D. Maltoni, R. Cappelli, J. Wayman, and A. Jain. Fvc 2000:fingerprint verification competition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24(3):402–412, 2002.

[10] D.Maio, D. Maltoni, R. Cappelli, J. Wayman, and A. Jain. Performance evaluation of fingerprint verification systems. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 28(1):3–18, 2006.

[11] E.Zhu, J.Yin, and G.Zhang. Fingerprint matching based on global alignment of multiple reference minutiae. *Pattern Recognition*, 38:1685–1694, 2005.

[12] F.Benhammadi, M.N.Amirouche, H.Hentous, K.Beghdad, and M.Aissani. Fingerprint matching from minutiae texture maps. *Pattern Recognition*, 40(1):189–197, 2004.

[13] FBI. U.s. department of justice, washington, d.c. 20402. *The Science of Fingerprints, Classification and Uses*, 1984.

[14] M. Garris, C. Watson, R. McCabe, and C. Wilson. User's guide to nist fingerprint image software(nfis). *Technical Report NISTIR 6813. National Institute of Standards and Technology*, 2002.

[15] G.Parziale and A.Niel. A fingerprint matching using minutiae triangulation. *Lecture Notes in Computer Science*, pages 241–248, 2004.

[16] G.S.Ng, X.Tong, X.Tang, and D.Shi. Adjacent orientation vector based fingerprint minutiae matching system. *Proc. Int'l Conf. Pattern Recognition*, pages 528–531, 2004.

[17] H.C.Lee and R.E.Gaensslen. Advances in fingerprint technology. *CRC Press, Boca Raton, FL*, 1994.

[18] H.Xu, R. Veldhuis, A.M.Bazen, T.Kevenaar, T.Akkermans, A.Ton, and B.Gökberk. Fingerprint verification using spectral minutiae representations. *IEEE Transactions on Information Forensics and Security*, pages 397–409, 2009.

[19] A. Jain, S. Prabhakar, L. Hong, and S. Pankanti. Fingercode: a filterbank for fingerprint representation and matching. *IEEE Computer Vision and Pattern Recognition*, 2:187–193, 1999.

[20] A. K. Jain and D. Maltoni. *Handbook of Fingerprint Recognition*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2003.

[21] J.Bery. The history and development of fingerprinting. *H.C.Lee and R.E.Gaensslen editors, Advances in Fingerprint Technology*, pages 1–38, 1994.

[22] J.Bringer and V.Despiegel. Binary feature vector fingerprint representation from minutiae vicinities. *IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, pages 1–6, 2010.

[23] J.D.Boer, A.M.Bazen, and S.H.Gerez. Indexing fingerprint databases based on multiple features. *ProRISC Workshop on Circuits, Systems and Signal Processing*, pages 300–306, 2001.

[24] J.Dougman. Recognizing persons by their iris patterns. *Biometrics: Personal Identification in a Networked Society*, 1999.

[25] J.Feng. Combining minutiae descriptors for fingerprint matching. *Pattern Recognition*, 41:342–352, 2008.

[26] J.Feng, Z.Ouyand, and A.Cai. Fingerprint matching using ridges. *Pattern Recognition*, 39:2131–2140, 2006.

[27] J.Qi and Y.Wang. A robust fingerprint matching method. *Pattern Recognition*, 38:1665–1671, 2005.

[28] K.Choi, D.Lee, S.Lee, and J.Kim. An improved fingerprint indexing algorithm based on the triplet approach. *AVBPA*, pages 584–591, 2003.

[29] X. Liang, A. Bishnu, and T. Asano. A robust fingerprint indexing scheme using minutia neighborhood structure and low-order delaunay triangles. *IEEE transactions on Information Forensics and Security*, 2(4):721–733, 2007.

[30] L.Sha, F.Zhao, and X.Tang. Improved fingercode for filterbank-based fingerprint matching. *International Conference on Image Processing*.

[31] M.Garris, C.Watson, R.McCabe, and C.Wilson. User's guide to nist fingerprint image software(nfis). *Technical Report NISTIR 6813. National Institute of Standards and Technology*, 2002.

[32] M.Medina-perez, A.G.Rodriguez, and M.G.Borroto. Improving fingerprint matching using an orientation-based minutia descriptor. *Iberoamerican Conference on Pattern Recognition, CIARP*, 2009.

[33] M.Tico and P.Kuosmanen. Fingerprint matching using an orientation-based minutia descriptor. *IEEE TPAMI*, 25(8):1009–1014, 2003.

[34] M.Tico, P.Kuosmanen, and J.Saarinen. Wavelet domain features for fingerprint recognition. *Electronic Letters*, 37(1):21–22, 2001.

[35] N.K.Ratha, V.D.Pandit, R.M.Bolle, and V.Vaish. Robust fingerprint authentication using local structural similarity. *IEEE Workshop Applications of Computer Vision*, pages 29–34, 2000.

[36] O.Iloanusi, A.Gyaourova, and A.Ross. Indexing fingerprints using minutiae quadruplets. *IEEE Compter Society Conference Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 127–133, 2011.

[37] P.Tuyls, A.Akkermans, T.Kevenaar, A.Bazen, R.Veldhuis, and G.Schrijen. Practical biometric authentication with template protection. *AVBPA*, pages 436–446, 2005.

[38] R.Cappelli, A.Lumini, D.Maio, and D.Maltoni. Fingerprint classification by directional image partitioning. *IEEE TPAMI*, 21(5):402–421, 1999.

[39] R.Cappelli, D.Maio, and D.Maltoni. A multi-classifier approach to fingerprint classification. *Pattern Analysis and Applications*, 5:136–144, 2002.

[40] R.Cappelli, M.Ferrara, and D.Maltoni. Minutia cylinder-code: A new representation and matching technique for fingerprint recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 32(12):2128–2141, 2010.

[41] R.Cappelli, M.Ferrara, and D.Maltoni. Fingerprint indexing based on minutia cylinder-code. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 33(5):1051–1057, 2011.

[42] R.Clarke. Human identification in information systems: Management challenges and public policy issues. *Information Technology and People*, 7(4):6–37, 1994.

[43] R.Germain, A.Califano, and S.Colville. Fingerprint matching using transformation parameter clustering. *IEEE Computational Science and Engineering*, 4:42–49, 1997.

[44] R.O.Duda, P.E.Hart, and D.G.Stork. Pattern classification (2nd edition). *Wiley-Interscience*, 2000.

[45] S.Chikkerur, A.N.Cartwright, and V.Govindaraju. K-plet and coupled bfs: A graph based fingerprint representation and matching algorithm. *Lecture Notes in Computer Science*, 3832:309–315, 2006.

[46] S.Lee, Y.Kim, and G.Park. A feature map consisting of orientation and inter-ridge spacing for fingerprint retrieval. *Proc. of AVBPA*, 2005.

[47] S.Pankanti, S.Prabhakar, and A.K.Jain. On the individuality of fingerprints. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 805–812, 2001.

[48] S.Tulyakov, F.Farooq, and V.Govindaraju. Symmetric hash functions for fingerprint minutiae. *Proc. Intl. Conference on Pattern Recognition and Image Analysis*, 2, 2005.

[49] T.Amornraksa and S.Tachaphetpiboon. Fingerprint recognition using dct features. *Electronic Letters*, 42(9):522–523, 2006.

[50] T.Lui, G.Zhu, P.Hao, and C.Zhang. Fingerprint indexing based on singular point correlation. *Proc. Int'l Conference on Image Processing*, 2005.

[51] W.Xu, X.Chen, and J.Feng. A robust fingerprint matching approach:growing and fusing of local structures. *Lecture Notes in Computer Science*, 4642:134, 2007.

[52] X.Chen, J.Tian, J.Yang, and Y.Zhang. An algorithm for distorted fingerprint matching based on local triangle feature set. *IEEE Trans. Information Forensics and Security*, 1(2):169–179, 2006.

[53] X.Jiang, M.Liu, and A.Kot. Fingerprint retrieval for identification. *IEEE TIFS*, 1:532–542, 2006.

[54] X.Jiang and W.Y.Yau. Fingerprint minutiae matching based on the local and global structures. *Poc. Int'l Conf. Pattern Recognition*, 2:6038–6041, 2000.

[55] X.Liang, T.Asano, and A.Bishnu. Distorted fingerprint indexing using minutia detail and delaunay triangle. *Proc. Int'l Symp. Voronoi Diagrams in Science and Eng.*, 2006.

[56] X.Shuai, C.Zhang, and P.Hao. Fingerprint indexing based on composite set of reduced sift features. *Proc. Int'l Conf. Pattern Recognition*, pages 1–4, 2008.

[57] Y.He, J.Tian, L.Li, Y.He, and X.Yang. Fingerprint matching based on global comprehensive similarity. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 28(6):850–862, 2006.

[58] Z.M.Kovacs-Vajna. A fingerprint verification system based on triangular matching and dynamic time warping. *IEEE TPAMI*, 22(11):1266–1276, 2000.