ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

ADDIS ABABA INISTITUTE OF TECHNOLOGY

FINGERPRINT RECOGNITION USING HYBRID MATCHING ALGORITHM

A THESIS SUBMITTED TO THE SCHOOL OF GRADUATE STUDIES OF ADDIS ABABA UNIVERSITY IN PARTIAL FULFILLMENT FOR THE DEGREE OF MASTER OF SCIENCE IN COMPUTER ENGINEERING

BY

BINIYAM MELESE

ADVISOR

Dr. ASSEFA DAGNE

ADDIS ABABA

AUGUST, 2003 E.C

ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

ADDIS ABABA INISTITUTE OF TECHNOLOGY

FINGERPRINT RECOGNITION USING HYBRID MATCHING ALGORITHM

BY

BINIYAM MELESE

Approval by Board of Examiners

_____	_____
Chairman, Department of Electrical and Computer Engineering	**Signature**
_____	_____
Advisor	**Signature**
_____	_____
External Examiner	**Signature**
_____	_____
Internal Examiner	**Signature**

ACKNOWLEDGMENT

First and foremost I offer my thanks to God and next to saint Virgin Mary for giving me strength to go through this thesis.

I express my dippiest gratitude to my advisor Dr. Assefa Dagne for his advice, guidance and patience throughout my thesis development.

Finally, I would like to thank my parents for their unlimited support; without them I wouldn't be where I am right now.

Biniyam Melese

August, 2003 E.C

CONTENT

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

AAD	Average Absolute Deviation from the Mean
ATM	Automated Teller Machine
cn	Crossing Number
DET	Detection-Error Tradeoff
dpi	Dot Per Inches
EER	Equal Error Rate
FMR	False Matching Rate
FNMR	False Non-Matching Rate
FVC	Finger Verification Competition
gms	Genuine Matching Score
ims	Impostor Matching Score
NGRA	Number of Genuine Recognition Attempt
NIRA	Number of Impostor Recognition Attempt
ROC	Receiving Operating Curve
SE	Structural Element

ABSTRACT

Fingerprint matching is one of the most important problems in fingerprint recognition system. Generally, fingerprint matching algorithm can be classified into two: minutiae based and non-minutiae based. In minutiae based matching, ridge endings and ridge bifurcations are used as discriminative features for matching, but in non-minutiae based, features other than minutiae are used.

Hybrid approaches by combining minutiae with other non-minutiae based matching methods are used to improve fingerprint matching methods. In this paper, alignment-based elastic matching algorithm, is used for minutiae based matching and new non-minutiae method proposed which solve the shortcomings of the algorithms proposed by Jain et.al.

The experiment was done on 320 fingerprints of Fingerprint Verification Competition 2002 set-B databases and showed that the hybrid matching algorithm improved the Equal Error Rate from ~11.8 % to 7.55% than the minutiae based matching algorithm.

Keyword: Image processing, Fingerprint recognition, local orientation, local frequency, minutiae, absolute average deviation from the mean, Gabor filter.

CHAPTER ONE : INTRODUCTION

1.1 Background

Fingerprint recognition is one of the oldest methods of biometric recognition. Modern fingerprint techniques were started since 16th century. In the early 20th century, fingerprint recognition was formally accepted as valid personal identification method by law enforcement agencies and become a standard routine in forensics [7].

In biometric recognition, measurements of the body are used for recognizing a person. These measurements, called identifiers, can be behavioral (e.g. speech) or anatomical (e.g. fingerprint, face, iris) [13].

1.1.1 Biometric Systems

There are two types of biometric systems: a verification system and an identification system [7] [13]:

- In verification system, a person claim to be, let say, Mr. X is verified by taking his biometric identifier and matching it with Mr. X's pre-stored identity. The system searches only Mr. X's biometric identifier in the database which is one to one comparison. Two examples of verification systems are logging in computer and taking money from ATM machines using biometric identifiers.

- In identification system, the system takes a person biometric identifier and compares it to all identifiers in the database to find the match. Here, the identification can be done even without the knowledge of the person. An example of identification system is finding a criminal by comparing its latent fingerprint (fingerprints that where left from the crime scene equipments) with the database.

Before verification and identification are carried out, databases are created by taking a person's biometric identifiers called enrollment process. If the biometric system is fingerprint recognition, the systems can be depicted using a flow chart as follows:

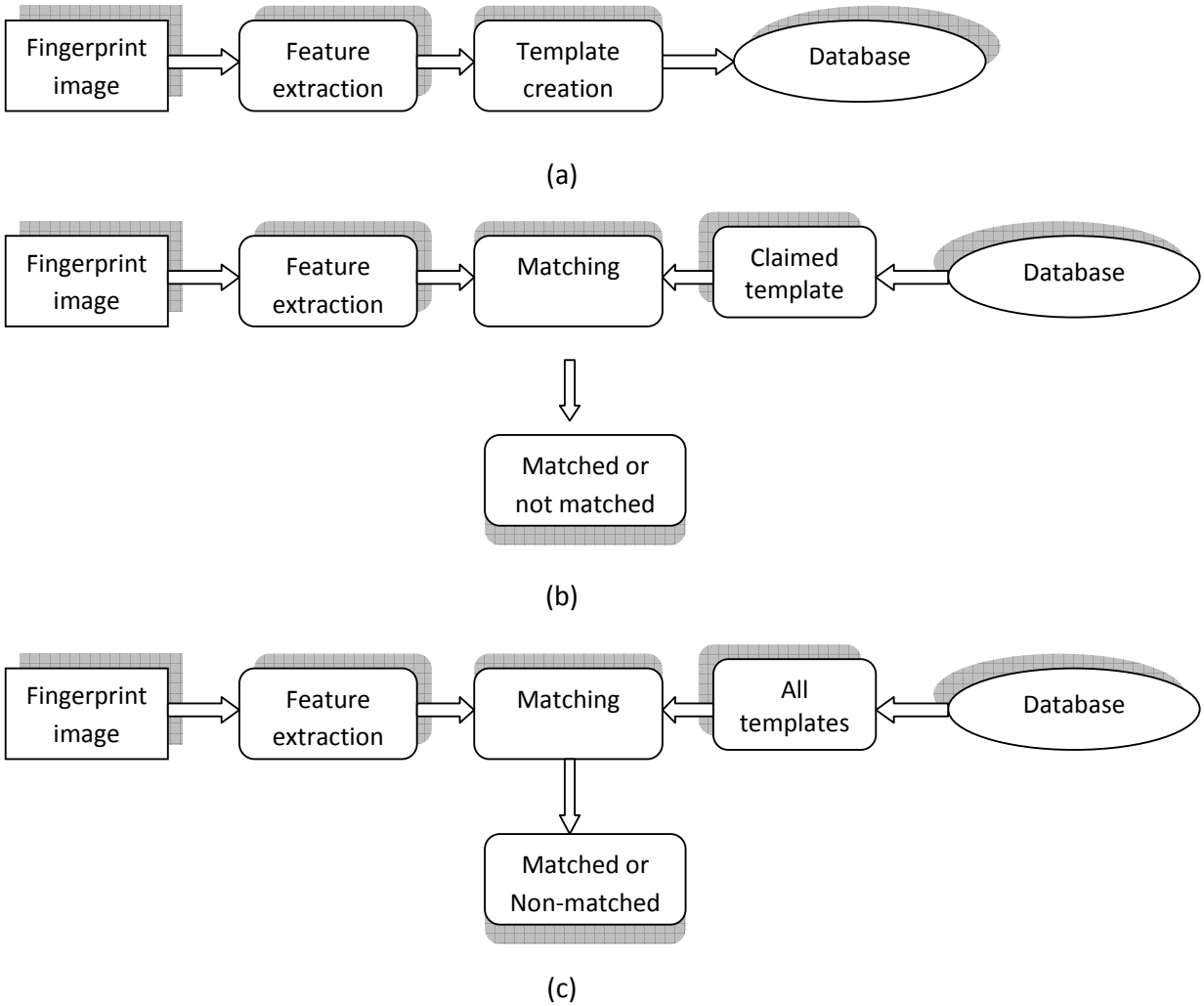


Figure 1.1 Fingerprint recognition system processes: a) enrollment, b) verification and c) identification [13].

Fingerprint recognition performance is mainly affected by fingerprint image quality and matching algorithm. For example, most of latent fingerprint images are poor quality because the area from which they extracted are dirty or not smooth. Extracting features from these poor quality images is very difficult which in turn degrade the performance of the system. Also, matching two fingerprints is very difficult. This is due to mainly the large variability in different impressions of same finger (i.e., large intra-class variations) and similarity of different impressions of different fingers (small inter-class variations). The main reason for intra-class variations are displacement, rotation, partial overlap, non-linear distortion,

pressure and skin-condition, noise and feature extraction errors. The following two figures show fingerprints that are from different fingers which look similar and fingerprints which are from same finger which may look different [13].

Figure 1.2 Fingerprints from different fingers which look similar [13].

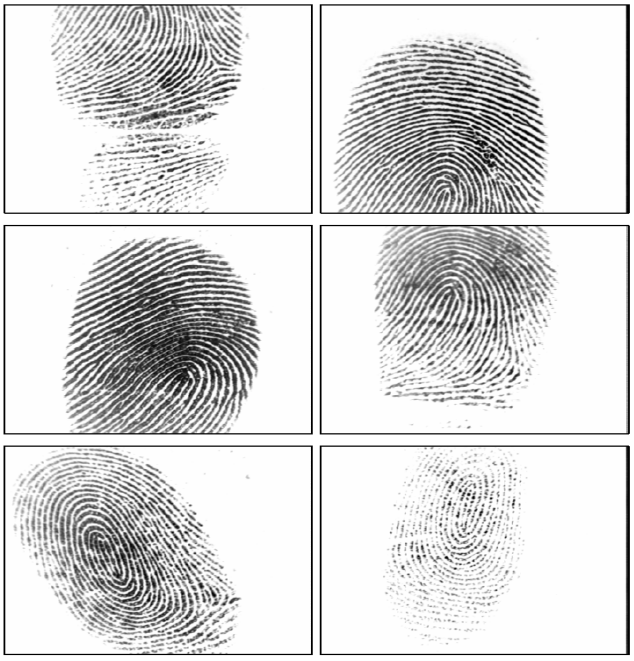


Figure 1.3 Fingerprints from same finger which look different [13].

1.2 Problem Description

Fingerprint recognition has become an interesting area of study. Some of the reasons can be the availability of low-cost fingerprint capturing devises, availability of free fingerprint image databases and increasing security issues. In the past few years, many researchers develop new fingerprint matching algorithms that have high performance than the previous ones or that create a new way to match. Other researchers try and search to hybrid the existing matching algorithms to minimize the errors.

Fingerprint matching algorithm can be generally classified into two, minutiae based and non-minutiae based. The minutiae based matching methods are the most common and widely used methods. This is because human fingerprint experts use these methods and are the accepted methods in most country court rooms for identifying a person. These matching methods are believed to have high performance than the other methods. The problem of these matching methods is extracting minutiae from poor-quality image is difficult. Also, if the number of extracted minutiae is small (e.g. small size fingerprint images) it is difficult to verify whether two fingerprint images are from the same finger or not. Therefore, a new way of matching methods is needed to help rather than to complement the minutiae based matching.

In non-minutiae based matching, features other than minutiae are used. Combining these methods with minutiae methods using hybrid techniques improves the matching performance.

1.3 Objective

General Objective

The objective of this paper is to improve the fingerprint matching by hybridizing minutiae based matching with other types of fingerprint matching methods.

Specific Objective

- To implement commonly used and appropriate algorithms for minutiae based fingerprint matching.

- To develop hybrid matching algorithm.
- To use available fingerprint image databases and to compare and evaluate each and the hybrid matching methods performance.

1.4 Methodology

Literatures are reviewed that discuss on fingerprint matching algorithms. Because the aim is to improve the fingerprint matching by hybridizing minutiae based matching with other types of fingerprint matching methods, the hybrid algorithms where taken one from minutiae based matching system and the other from non-minutiae feature based matching system. In minutiae based methods, common methods are used for extracting the minutiae features and elastic point patter matching method is used for minutiae matching in order to tolerate non-linear deformation and inexact transformations between different fingerprints. For non minutiae based matching, a new system is proposed that solve the shortcomings of filterbank based method. Then each and the hybrid algorithms are evaluated using common standard performance evaluation methods and the result is presented.

1.5 Scope

The tuning and testing processes on the hybrid matching algorithm were carried out only using FVC 2002 set-B databases which are freely available on the internet. Also, this thesis work emphasizes only on improving the accuracy of the hybrid matching algorithm (i.e. it tries to improve the equal error rate). Other performance results, like the size and speed of the hybrid matching algorithm, are not considered due to time constraint.

1.6 Thesis Organization

This thesis paper is organized into four chapters. The next chapter (chapter II) is a literature survey of the techniques used to match two fingerprints. It briefly discusses the common minutiae based and one of the non-minutiae based fingerprint matching methods. It also discusses how to evaluate fingerprint matching method. In chapter three, the proposed hybrid fingerprint matching model and the shortcomings that it solves from previous work will be presented. Finally, the performance result of the proposed matching method using standard performance evaluation method and the conclusion will be presented in chapter four.

CHAPTER TWO : LITERATURE SURVEY

2.1 Fingerprint Matching

Fingerprint matching is one part of fingerprint recognition system. It compares two fingerprints and decides whether they are from the same finger or not or it gives the degree of similarity between 1 and 0 value. Fingerprints have features that make them unique. In order to match fingerprints, we need to extract these features from a person's fingerprint. After extraction, these features represent the fingerprint images and are used in fingerprint matching.

The exterior skin of our fingertip makes a pattern, which is made up of ridges and valleys. The uniqueness of our fingerprint lies on this pattern. On an image, fingerprint ridges are dark whereas, valleys are white. The unique features can be classified in to three levels based on the view of detail [13].

Level1: When we see the fingerprint as a whole, ridges and valleys pattern generally make one of the following shapes as shown in the figure 2.1.

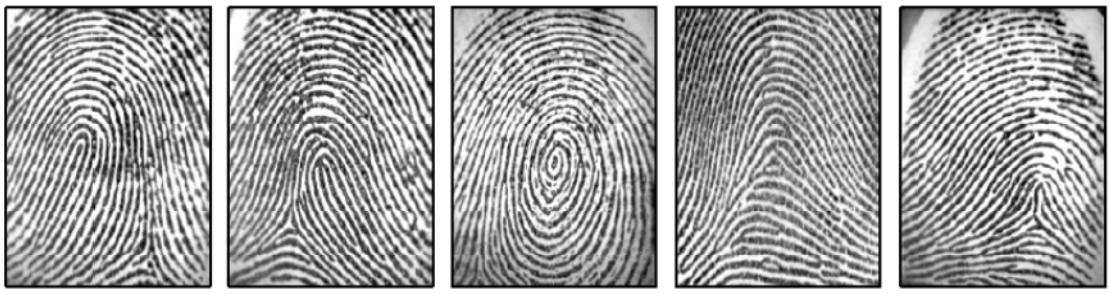


Figure 2.1 Types of fingerprints (from left to right: left loop, right loop, whorl, arch and tented arch) [13].

Even if these patterns are not very distinctive, fingerprints fall in one of the above types. These features can be used to store fingerprints with the same type in one database. This help to decrease the time consumption for searching fingerprints from the database.

In these types of patterns, there are areas the ridges make high curvature called singular regions (singularity). There are three types of singular regions; these are delta, loop and whorl as shown in the figure below. At the center of the loop and whorl regions, there is a point called core which is used to align fingerprints.

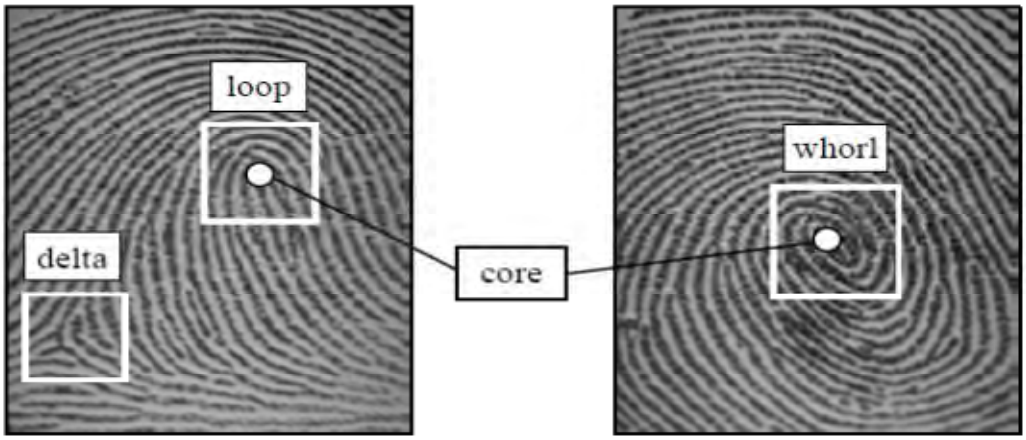


Figure 2.2 The singular regions delta, loop and whorl in right loop and whorl types of fingerprint images [13].

Level 2: The next detail level that fingerprint exhibit is called minutiae points (singular: minutia). These points are where the ridges make sudden changes like ridge ending and ridge union. Some of the types of minutiae are shown in the figure below:

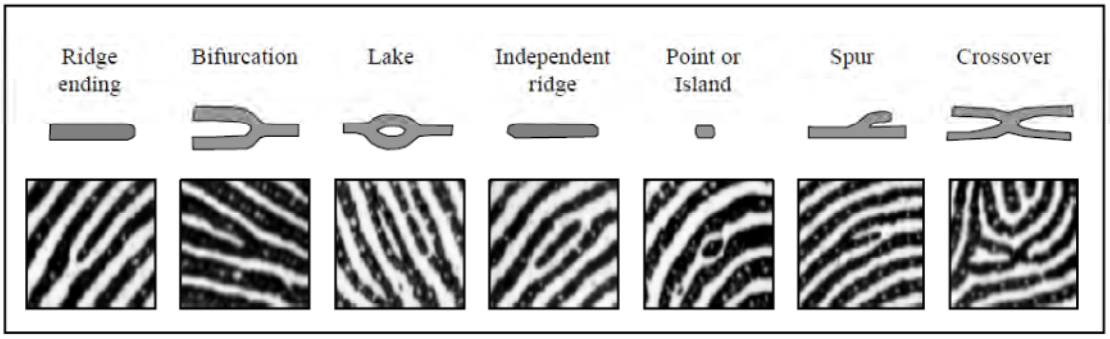


Figure 2.3 Different types of minutiae [13].

From different types of minutiae, ridge ending and bifurcation are the most common features used in fingerprint matching. These features are also used by human experts and are accepted by courts of law.

Level 3: The 3rd detail levels are sweet pores and local shape of ridges. These are very distinct and a small number is enough for comparing fingerprints. But the problem is they need high resolution scanners to be captured, so rarely used.

Figure 2.4 Sweet pores [13].

There are many methods to match fingerprints using minutiae. There are also different methods that are non-minutiae based matching. From each class, the methods used in this paper are discussed in the next sections.

2.2 Minutiae-based Matching

Minutiae are second level features as explained in pervious sections. Minutiae, or Galton's characteristics, are local discontinuities in fingerprint pattern [11]. They are the features used by human fingerprint experts and are the most widely used features in fingerprint matching. The two most minutiae types used for matching are bifurcation and termination [13].

Minutiae are extracted either directly from gray scale image or binary image. The basic process of minutiae based matching is as follows:

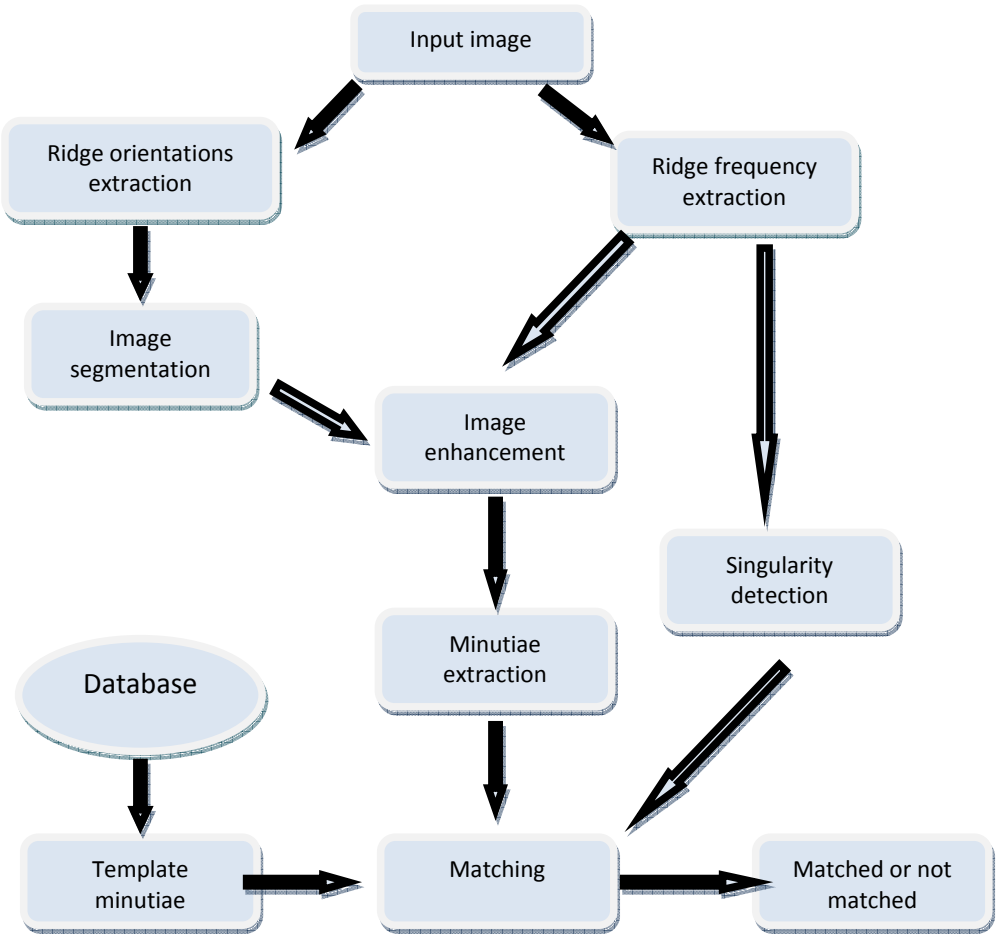


Figure 2.5 Minutiae based matching.

First from the image, orientation and frequency of ridges are extracted. From these extracted features, the input image is enhanced. After enhancement, the ridges are represented in single pixel wide lines by methods called binarization and thinning. This makes minutiae extraction simple. Finally, from the database storage a minutiae represented fingerprint image called template file is taken and matched with the extracted minutiae. For alignment of the input image and the template image, singular points that were detected from the ridges orientation or other techniques are used. The output is either matched or not or the degree of match, usually from 0 to 1 value.

2.2.1 Ridges Orientation

The local ridges orientation at a pixel $[\,x, y\,]$ is an angle θ_{xy} that the fingerprint ridges, crossing through an arbitrary small neighborhood centered at $[\,x, y\,]$, make from the horizontal axis [13].

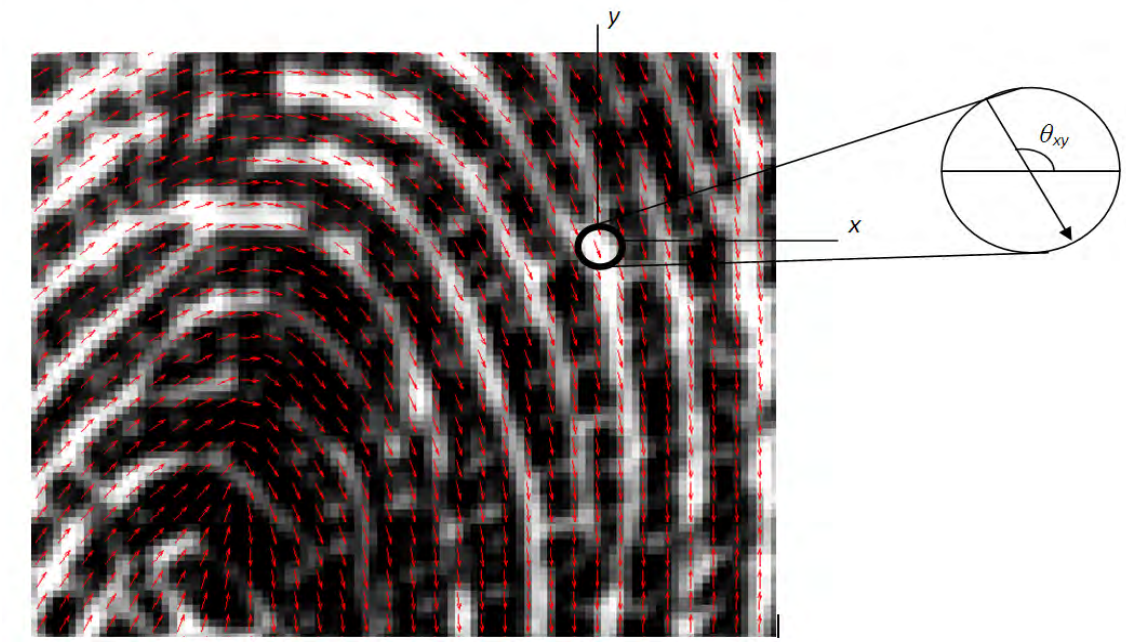


Figure 2.6 Local ridges orientation.

The common method for computing ridges orientation is the gradient based [7] [6] [8] [2]. In gradient based, the gradient of the images in x and y direction of each pixel is computed. The gradients, G_x and G_y in x and y directions, respectively, are computed as follows [3]:

$$\begin{bmatrix} G_x(x,y) \\ G_y(x,y) \end{bmatrix} = sign(G_x) * \nabla I(x,y)$$

$$= sign\left(\frac{\partial I(x,y)}{\partial x}\right) \begin{bmatrix} \frac{\partial I(x,y)}{\partial x} \\ \frac{\partial I(x,y)}{\partial y} \end{bmatrix} \quad (2.1)$$

where $I(x,y)$ represent the gray scale image and $sign(G_x)$ is the sign of the gradient in the x-direction. The gradients in polar form are computed as follows:

$$\begin{bmatrix} G_\rho \\ G_\varphi \end{bmatrix} = \begin{bmatrix} \sqrt{G_x^2 + G_y^2} \\ \tan^{-1} G_y/G_x \end{bmatrix} \tag{2.2}$$

The gradient vector is converted back to its Cartesian representation by:

$$\begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} G_\rho \cos G_\varphi \\ G_\rho \sin G_\varphi \end{bmatrix} \tag{2.3}$$

After computing the gradient vector, the square of the gradients $\begin{bmatrix} G_{s,x} & G_{s,y} \end{bmatrix}^T$ is computed:

$$\begin{bmatrix} G_{s,x} \\ G_{s,y} \end{bmatrix} = \begin{bmatrix} G_\rho^2 \cos 2G_\varphi \\ G_\rho^2 \sin 2\, G_\varphi \end{bmatrix} = \begin{bmatrix} G_\rho^2 \left(\cos^2 G_\varphi - \sin^2 G_\varphi\right) \\ G_\rho^2 (2 \sin G_\varphi \cos G_\varphi) \end{bmatrix} = \begin{bmatrix} G_x^2 - G_y^2 \\ 2G_x G_y \end{bmatrix} \tag{2.4}$$

Then the square gradients are averaged over a window w $(20\ x\ 20)$. The average squared gradient vector $\begin{bmatrix} \overline{G_{s,x}} & \overline{G_{s,y}} \end{bmatrix}^T$ is computed as follows:

$$\begin{bmatrix} \overline{G_{s,x}} \\ \overline{G_{s,y}} \end{bmatrix} = \begin{bmatrix} \sum_w G_{s,x} \\ \sum_w G_{s,y} \end{bmatrix} = \begin{bmatrix} \sum_w G_x^2 - G_y^2 \\ \sum_w 2G_x G_y \end{bmatrix} = \begin{bmatrix} G_{xx} - G_{yy} \\ 2G_{xy} \end{bmatrix} \tag{2.5}$$

In this expression,

$$G_{xx} = \sum_w G_x^2$$

$$G_{yy} = \sum_w G_y^2$$

$$G_{xy} = \sum_w G_x G_y$$

are estimates for the variances and cross covariance of G_x and G_y, averaged over the window w. Now, the average gradient direction θ_{av}, with $-\frac{1}{2}\pi < \theta_{av} \leq \frac{1}{2}\pi$ is given by:

$$\theta_{av} = \frac{1}{2} \angle \left(G_{xx} - G_{yy}, 2G_{xy}\right) \tag{2.6}$$

Where $\angle\ (x, y)$ is defined as:

$$\angle(x,y) = \begin{cases} \tan^{-1}(\frac{y}{x}), & x \geq 0 \\ \tan^{-1}(\frac{y}{x}) + \pi, & for\ x < 0\ \wedge y \geq 0 \\ \tan^{-1}(\frac{y}{x}) - \pi, & x < 0\ \wedge y \geq 0 \end{cases} \quad (2.7)$$

and the average ridge-valley direction θ, with $-\frac{1}{2}\pi < \theta \leq \frac{1}{2}\pi$, which is perpendicular to θ_{av} is given by:

$$\theta = \begin{cases} \theta_{av} + \frac{1}{2}\pi, & for\ \theta_{av} \leq 0 \\ \theta_{av} - \frac{1}{2}\pi, & \theta_{av} > 0 \end{cases} \quad (2.8)$$

The reliability (coherence) of the ridges orientation for each pixel can be computed as follows:

$$Coh = \frac{|\sum_w (G_{s,x}, G_{s,y})|}{\sum_w |(G_{s,x}, G_{s,y})|} \quad (2.9)$$

2.2.2 Ridges Frequency

The reciprocal of the distance between ridges in arbitrary small neighborhood of a pixel is called ridges frequency. This fingerprint feature is very useful in later process, especially in image enhancement. For each pixel or block of pixel, the ridges frequency is calculated from ridges that pass through them or through their near neighbors.

One method of calculating ridges frequency is by modeling the ridges and valleys as a sinusoidal shaped wave along a direction normal to the local ridges orientation [6]. Let I be the gray scale image and θ be the local orientation, and then the steps involved in local ridges frequency estimation are as follows.

1. Divide the fingerprint image $I(i,j)$ into blocks of size $w\ x\ w$ $(16\ x\ 16)$.

2. For each block centered at pixel (i,j), compute an oriented window of size $l\ x\ w$ $(32\ x\ 16)$. This window is perpendicular to the ridges.

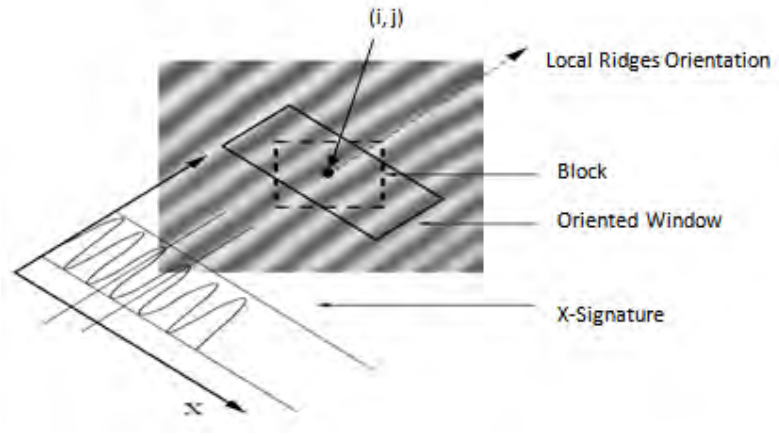


Figure 2.7 Oriented window and X-signature [6].

3. Since ridges have low and valleys have high gray scale value, the summations of the rows (called X-signature) of each columns of the oriented window make a wave.

4. The X-signature, $X\ [0], X\ [1]\ \dots\ X\ [l\ -\ 1]$, are calculated as follows:

$$X[k] = \frac{1}{w}\sum_{d=0}^{w-1} I(u,v), k = 0, 1, \dots, l-1, \tag{2.10}$$

$$u = i + \left(d - \frac{w}{2}\right)\cos\big(\theta(i,j)\big) + \left(k - \frac{l}{2}\right)\sin\big(\theta(i,j)\big), \tag{2.11}$$

$$v = j + \left(d - \frac{w}{2}\right)\sin\big(\theta(i,j)\big) + \left(\frac{l}{2} - k\right)\cos\big(\theta(i,j)\big). \tag{2.12}$$

Because of the assumption of the ridges and valleys make sinusoidal wave shape, the frequency of the wave is taken as the ridges frequency. Let $\mathcal{T}\ (i,j)$ be the average number of pixels between two consecutive peaks in the X-signature, then the frequency, $\Omega\ (i,j)$ is given by:

$$\Omega\ (i,j) = \frac{1}{\mathcal{T}\ (i,j)} \tag{2.13}$$

5. The blocks in which minutiae or singular points appear or ridges and valleys are corrupted do not form a well defined sinusoidal shaped wave. The frequency values for these blocks need to be interpolated from the frequency of the neighboring

blocks which have a well-defined frequency. The interpolation is performed as follows:

$$\Omega(i,j) = \begin{cases} \Omega(i,j), & if\ \Omega(i,j) = -1 \\ \dfrac{\sum_{u=-\frac{w_\Omega}{2}}^{\frac{w_\Omega}{2}} \sum_{v=-\frac{\omega_\Omega}{2}}^{\frac{w_\Omega}{2}} W_g(u,v) * \mu(\,\Omega(i-uw, j-vw)\,)}{\sum_{u=-\frac{w_\Omega}{2}}^{\frac{w_\Omega}{2}} \sum_{v=-\frac{w_\Omega}{2}}^{\frac{w_\Omega}{2}} W_g(u,v) * \delta(\,\Omega(i-uw, j-vw)+1\,)}, & otherwise \end{cases} \quad (2.14)$$

where

$$\mu(x) = \begin{cases} 0, & if\ x \leq 0 \\ x, & otherwise, \end{cases}$$

$$\delta(x) = \begin{cases} 0, & if\ x \leq 0 \\ 1, & otherwise \end{cases}$$

Wg is a discrete Gaussian Kernel where mean and variance is 0 and 9, respectively, and $w_\Omega = 7$ is the size of the kernel. To remove the outlier in the estimated ridges frequency, a low pass filter is used.

$$R_f(i,j) = \sum_{u=-\frac{w_l}{2}}^{\frac{w_l}{2}} \sum_{v=-\frac{w_l}{2}}^{\frac{w_l}{2}} W_l(u,v) * \Omega(i-uw, j-vw) \quad (2.15)$$

where W_l is a 2-dimensional low-pass filter with unit integral and $w_l = 7$ is the size of the filter.

2.2.3 Segmentation

The term *segmentation* is generally used to denote the separation of fingerprint area (foreground) from the image background. Separating the background is useful to avoid extraction of features in noisy areas that are often in the background [13].

Image segmentation is one of the difficult parts of fingerprint recognition system. There are many methods to segment fingerprint image from the background. Some of the methods are segmentation based on:

1. Gray scale mean,
2. Variance,
3. Coherence and

4. The magnitude of image gradients.

I found that coherence or magnitude of image gradients based segmentation to be better than the others [2] [16]. The reason is, coherence and gradients of images can be computed pixel wise and are more discriminative between the foreground and background of the image.

Ming et al. [16] proposed fingerprint segmentation method based on the magnitude of the image gradients. First, magnitude of the image gradients G are calculated for each pixel.

$$G = \sqrt{G_x^2 + G_y^2} \tag{2.16}$$

The gradient matrix is filtered with 7×7 Gaussian filter operator. Then the mean of the gradient matrix is computed which is used as threshold T.

$$T = \frac{1}{m*n} \sum_{i=1}^{m} \sum_{j=1}^{n} G(i,j) \tag{2.17}$$

where m and n are the number of rows and columns of the gradient matrix respectively.
The image is segmented as follows (see figure 2.8(b)):

$$G(i,j) = \begin{cases} 1 & if\ G(i,j) > T \\ 0 & otherwise \end{cases} \tag{2.18}$$

After the image is segemented, using morphological closing and opening operations with structural element (SE), holes are filled and isolated objects are eliminated from the segmented image (see figure 2.8(c)). Morphology is a branch of image processing which is particularly useful for analyzing shapes in images [5] [14].

$$\text{SE} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.19}$$

Finally, the border of the segmented image is smoothed using Fourier transform as follows (see figure 2.8(d)):

$$X_{fft} = fft(x), \tag{2.20}$$

$$Y_{fft} = fft(y), \tag{2.21}$$

$$X_{fft} = \begin{cases} X_{fft}, & if\ X_{fft} \leq 10 \\ 0, & otherwise \end{cases} \tag{2.22}$$

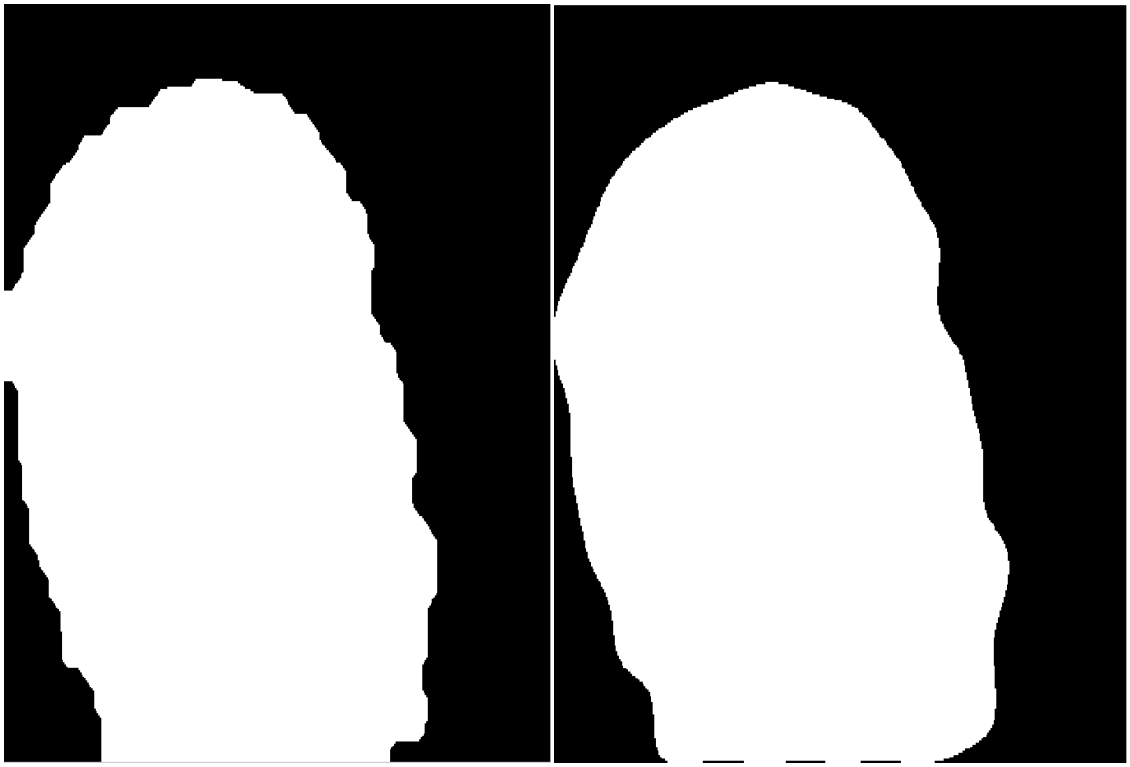
$$Y_{fft} = \begin{cases} Y_{fft}, & if\ Y_{fft} \leq 10 \\ 0, & otherwise \end{cases} \tag{2.23}$$

$$x = iff(X_{fft}) \tag{2.24}$$

$$y = iff(Y_{fft}) \tag{2.25}$$

where x and y are the cordinates of the segmented image border.

(a) (b)



(c) (d)

Figure 2.8 Different segmentation stages of the algorithm proposed by Ming et al. [45].

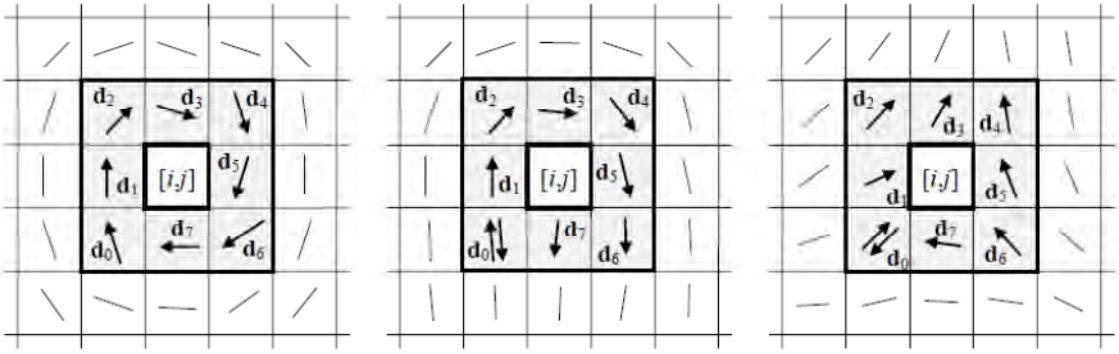
2.2.4 Singularity and Core Detection

Singular or core points are points that exhibit the maximum ridges curvature in the fingerprint image. These points are used to align two fingerprint images or for classifying the images into one of the fingerprint type when storing to or indexing from the database. Finding singular points is a difficult task especially in arc type images. One of the common singular or core detection method is Poincaré index [13]. Poincaré index is the summation of the difference of local orientation of a pair of pixels around each pixel of the local orientation image [see figure 2.9]. Let $Pc(i,j)$ be Poincaré index of pixel (i,j), then it is computed as follows:

$$Pc(i,j) = \sum_{k=0\ldots7} angle(d_k, d_{(k+1)\ mod\ 8}), \quad k = 0,1,\ldots 7 \tag{2.26}$$

where $angle(x, y)$ is defined as the difference of x and y and d_k are neighborhood pixels of pixel (i, j). The Poincaré index value assumes one of the following discrete values:

$$Pc(i,j) = \begin{cases} 0°, & if\ (i,j)\ doesnt\ belong\ to\ any\ singular\ region \\ 360°, & if\ (i,j)\ belongs\ to\ a\ whorl\ type\ singular\ region \\ 180°, & if\ (i,j)\ belongs\ to\ a\ loop\ type\ singular\ region \\ -180°, & if\ (i,j)\ belongs\ to\ a\ delta\ type\ singular\ region \end{cases} \quad (2.27)$$



$Pc(i,j) = 360°$ $Pc(i,j) = 180°$ $Pc(i,j) = -180°$

Figure 2.9 Examples of Poincaré index computations [13].

Bazen A.M. and Sabih Gerez H.S. [3] proposed a new method for calculating the Poincaré index. Summing the changes in orientation corresponds to summing the gradients of the squared of the orientation. Therefore, first they compute gradients of the squared orientation. Gradients of the squared orientation J can be directly computed from local orientation $\theta(x, y)$ as follows:

$$\begin{bmatrix} J_x(x,y) \\ J_y(x,y) \end{bmatrix} = \nabla(2 * \theta(x,y)) = \begin{bmatrix} \frac{\partial(2*\theta(x,y))}{\partial x} \\ \frac{\partial(2*\theta(x,y))}{\partial y} \end{bmatrix} \quad (2.28)$$

where $\theta(x, y)$ is the local orientation and both components of J should be calculated modulo of 2π, such that they are always between $-\pi$ and π. Then they apply the Green's theorem on the gradients of the squared orientation which state that, a closed line-integral over a vector field can be calculated as the surface integral over the rotation of this vector field.

$$\oint_{\partial A} V_x\,dx + V_y\,dy = \iint_A rot[V_x\ V_y]^T\ dxdy$$
$$= \iint_A \left(\frac{\partial V_y}{\partial x} - \frac{\partial V_x}{\partial y}\right) dxdy \quad (2.29)$$

where $x\ and\ y$ define the coordinate system, A is the area, ∂A is the contour around this area and $[V_x\ V_y]^T$ is the vector filed.

By applying the above theorem on the gradients of squared orientation for each pixel, the Poincaré index (singularity or core point) is given by:

$$Index = \sum_{\Delta x,\Delta y\ along\ \partial A}(J_x\,.\,\Delta x + J_y.\,\Delta y) = \sum_A rot[J_x\ J_y]^T\ = \sum_A(\frac{\partial J_y}{\partial x} - \frac{\partial J_x}{\partial y}) \quad (2.30)$$

Since, the Poincaré index is calculated for each pixel the area is a square of one pixel. The result is for core point 2π , for delta -2π and for other pixels zero (see figure 2.10).

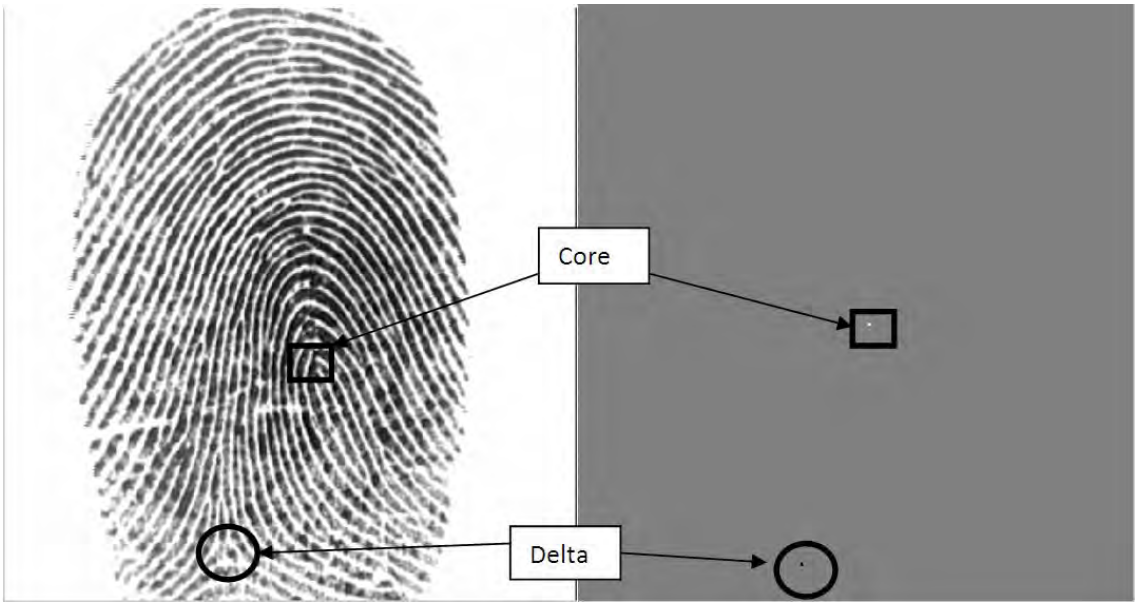


Figure 2.10 Gray scale representation of a core (2π) and delta (-2π) singularities as white and black points respectively.

2.2.5 Enhancement

Fingerprint recognition system performance is highly affected by the image quality of the fingerprint. Image enhancement in fingerprint images means, improving the clarity between ridges and valleys [13]. This is a very difficult task because in many real fingerprint images there is no clear distinction between the ridges and valleys. Rather a continuous gray-scale value change exists. Also the quality of image can be affected by finger skin-condition (e.g., wet or dry, cut, and bruises), sensor noise, incorrect finger pressure, and inherently low-

quality fingers (e.g., elderly people, manual workers) [13] which make enhancement difficult.

Most research papers use Gabor filter to enhance the fingerprint image which was proposed by Hong, Wan, and Jain [6]. The even-symmetric Gabor filter has a general form:

$$h(x, y: \emptyset, f) = exp\left\{-\frac{1}{2}\left[\frac{(x*cos\emptyset)^2}{\delta_x^2} + \frac{(y*sin\emptyset)^2}{\delta_y^2}\right]\right\}\cos(2\pi f * x * cos\emptyset) \quad (2.31)$$

where ϕ is the orientation of the Gabor filter, f is the frequency of the sinusoidal plane wave, and δ_x and δ_y are the space constant of the Gaussian envelope along x and y axes respectively (see figure 2.11).

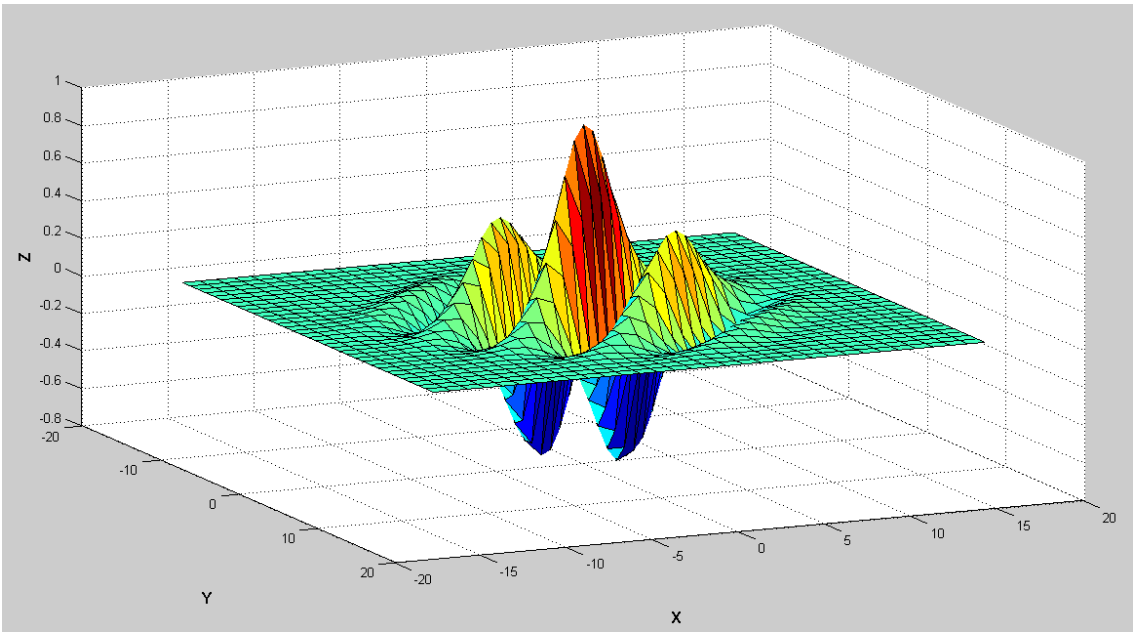


Figure 2.11 A Gabor filter with 0.2 frequency and 45^o orientation.

The space constants are set to a value of 4 based on empirical data and the frequency and orientation are set equal to the local frequency and orientation of the ridges respectively. Computing Gabor filter for each pixel (i.e. for each orientation and for each frequency) and filtering each pixel with their computed Gabor filter is time consuming. Instead, the Gabor filter is pre-computed and stored for a discrete number of frequencies and directions (see figure 2.12).

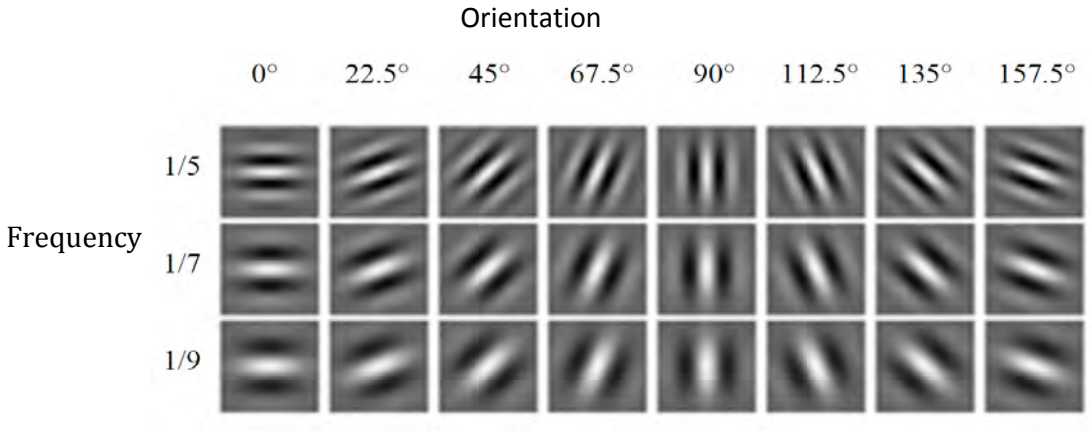


Figure 2.12 Discrete bank of Gabor filters in gray scale image representation [13].

Figure 2.13 Fingerprint image before and after Gabor filter enhancement.

2.2.6 Minutiae Detection

Most of minutiae detection methods needs that the fingerprint image ridges should be one pixel wide. To make the ridges one pixel wide, first the enhanced image must be converted to binary image. Binary image can be computed using a threshold value T by assigning the enhanced image pixels value greater than T to 1 and the remaining to 0. The binarized

image has thick ridges and valleys. Using thinning algorithm the ridges pixel is reduced to one pixel wide called thinned (skeleton) image.

After thinning, extracting minutia is very simple. Minutiae can be extracted using crossing number (cn) method. It is computed for each pixel as follows:

$$cn(p) = \frac{1}{2}\sum_{i=1\ldots8}|val(\mathrm{p}_{i \bmod 8}) - val(\mathrm{p}_{i-1})| \quad (2.32)$$

where p is a pixel in the binary image and $p_0,\ p_1, \ldots, p_7$ are an order sequence that define the neighborhood of p (see figure 2.14).

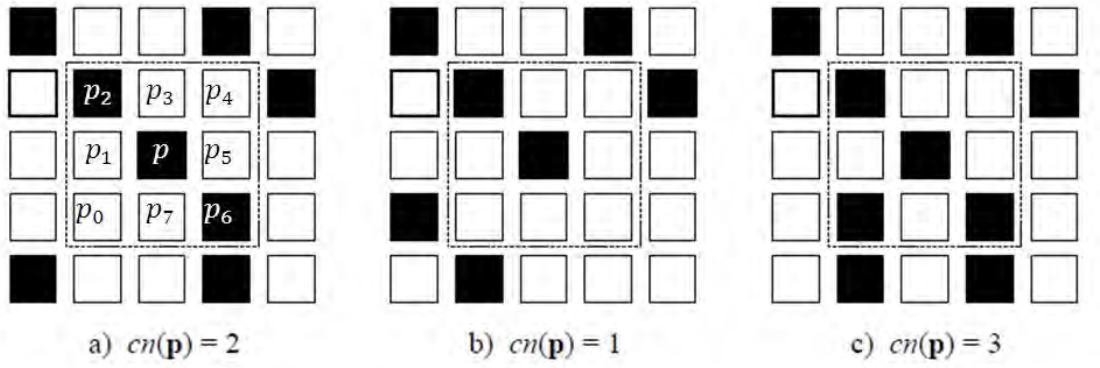


Figure 2.14 a) Intra-ridge pixel; b) ridge ending minutia; c) bifurcation minutia [13].

The crossing number of a pixel can have a value

- 2 if p is an intermediate ridge point
- 1 if p is a ridge ending
- 3 if p is point of bifurcation
- >3 if p is crossover or other complex minutia.

When the enhanced image is converted to a thinned image, most of the time only some of the rich information that the gray-scale contain are transmitted to the thinned image. This leads to false minutiae extraction or minutiae can be missed. The above problem can be reduced using pre and post processing algorithms. The task of preprocessing can be removing small ridges and filing holes while that of post processing can be removing spike on the ridges and validating the extracted minutiae through different method [9][18][19].

Sometimes minutiae are extracted with their descriptions. Descriptions of minutiae can be local orientation, ridges associated with them, number of minutiae around them within some length, etc. These descriptive features are stored along with their corresponding minutiae.

Figure 2.15 Extracted minutiae of the enhanced image shown in figure 2.13.

2.2.7 Minutiae Matching

Minutiae matching methods are a matured fingerprint matching technique that is believed to have high performance than the other matching methods [17]. There are numerous minutiae matching methods but the method proposed by Luo et al. [10] is used in this paper. Luo et al. modified the method proposed by Jain et al [7] which is alignment-based elastic matching algorithm. The algorithm tolerates non-linear deformation and inexact transformations between different fingerprints. The algorithm proposed by Lue et al. can be explained as follows:

1. Extract minutiae with their local orientation and ridges associated with them for each input and template images.

$$\mathrm{P} = ((x_1^P, y_1^P, \theta_1^P)^T, \dots, (x_M^P, y_M^P, \theta_M^P)^T) \tag{2.33}$$

$$\mathrm{Q} = \left(\left(x_1^Q, y_1^Q, \theta_1^Q\right)^T, \ldots, \left(x_N^Q, y_N^Q, \theta_N^Q\right)^T\right) \quad (2.34)$$

where P and Q denotes the input and template images minutiae sets respectively. M and N denotes the number of minutiae in the sets P and Q respectively. x and y are the coordinate values of the corresponding minutiae and θ denote the local orientation.

2. For each template minutia P_i and for each input minutia Q_j , if the types of the two minutiae are the same and the distance and angle difference of the points, taken from the associated ridges, is less than some specified value, set $rot\ [i][j]$ as the orientation difference between the input and template minutiae, otherwise set $rot\ [i][j]$ to 400. The distance and angle difference of the points are computed as follows:

$$Diff\_dist = \frac{1}{L}\sum_{i=0}^{L}|R_i(d_i) - R_t(d_i)| \quad (2.35)$$

$$Diff\_ang = \frac{1}{L}\sum_{i=0}^{L}|R_i(\alpha_i) - R_t(\alpha_i)| \quad (2.36)$$

where, L = 10 is the number of points recorded. R_i and R_t are the ridges of the input and the template minutiae respectively. d_i is the distance from point i on the ridge to the minutia and α_i is the angle between the line connecting point i on the ridge to the minutia (see figure 2.16).

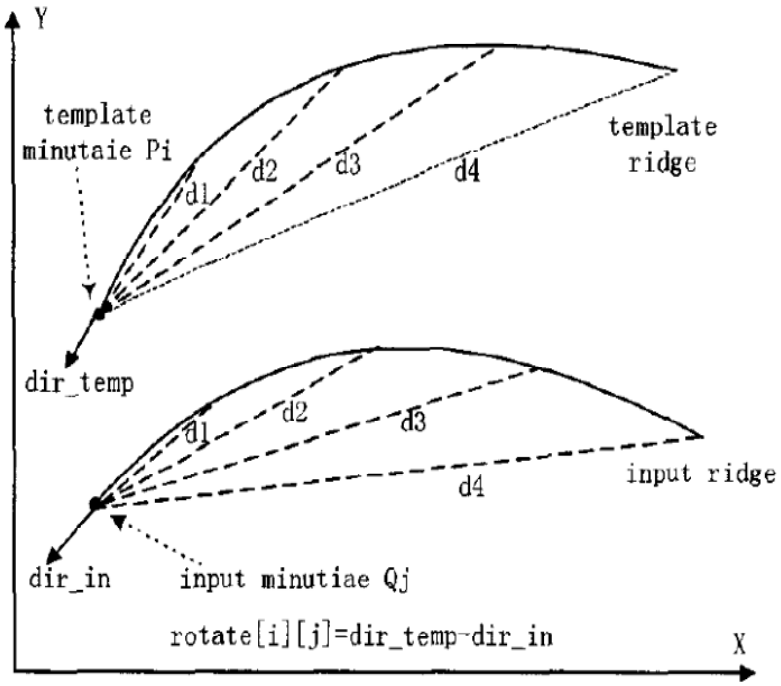


Figure 2.16 Alignment of ridges of the input and template images [10].

3. Take from each minutiae set one minutia and make them as reference points if $rot\,[i][j] < 400$. Translate the input and template minutiae sets into polar coordinate with respect to the reference minutiae P_i and Q_j. That is, for any minutia $(x_i, y_i, \theta_i)^T$, apply the following formula:

$$\begin{pmatrix} r_i \\ e_i \\ \theta_i \end{pmatrix} = \begin{pmatrix} \sqrt{(x_i - x^r)^2 + (y_i - y^r)^2} \\ \tan^{-1}\left(\frac{y_i - y^r}{x_i - x^r}\right) + rot[i][j] \\ \theta_i - \theta^r \end{pmatrix} \tag{2.37}$$

where $(x^r, y^r, \theta^r)^T$ is the coordinates and orientations of the reference minutiae and $(r_i, e_i, \theta_i)^T$ is the representation of the minutiae in polar coordinate system. r_i represent the radial distance, e_i represent the radial angle , θ_i represent the local orientation and $rot\,[i][j]$ is the orientation difference between the input and template reference points. The sorted minutiae sets, with respect minutiae i and j of the input and template, are denoted by P_i^S and Q_i^S respectively.

4. Sort the template sets in increasing order of radial angles. Compute the size of a bounding box for each minutia from the sorted template minutiae set (P_i^S). A bounding box is a box that is placed around each template minutia which is used to match the input minutia if it falls inside the box when aligned (see figure 2.17).

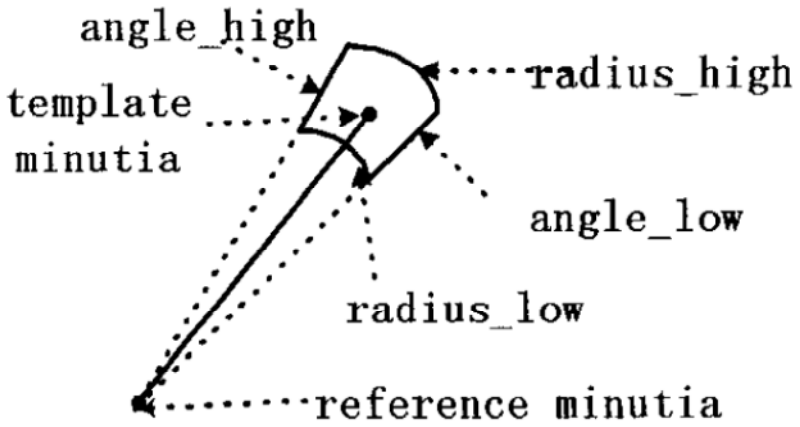


Figure 2.17 Bounding box [10].

The bounding box size attached to a template minutia with radius r is computed as follows:

$$radius\_size = \begin{cases} r\_small, & if\ r\_size < r\_small \\ r\_size, & if\ r\_small < r\_size < r\_large \\ r\_large, & if\ r\_size > r\_large \end{cases} \quad (2.38)$$

$$r\_size = \frac{r}{\alpha} \quad (2.39)$$

$$angle\_size = \begin{cases} a\_small, & if\ a\_size < a\_small \\ a\_size, & if\ a\_small < a\_size < a\_large \\ a\_large\ , & if\ a\_size > a\_large \end{cases} \quad (2.40)$$

$$a\_size = \frac{\beta}{r^2} \quad (2.41)$$

where $\alpha = 5$ and $\beta = 5$ are predefined constants, $angle\_size = angle\_high - angle\_low$ and $radius\_size = radius\_high - radius\_low$. The value of $angle\_size$ and $radius\_size$ change for each minutiae. If the radius of the template

minutia is larger, then its bounding box will have a larger $radius\_size$ and a smaller $angle\_size$ (see figure 2.18).

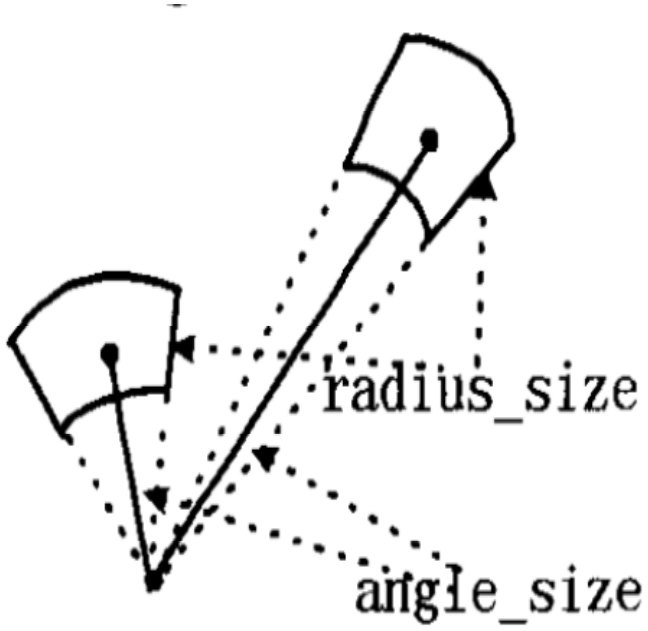


Figure 2.18 Changeable size bounding box [10].

5. For each minutia of the sorted template, each sorted input minutia is checked whether they fall inside the bounding of the template minutia or not. If an input minutia falls inside the bounding box then a score will be given to the reference minutiae pair of the input and the template. Let l denote the l^{th} point and k denote the k^{th} point in the sorted template and input minutiae set respectively. The condition whether an input minutia k fall inside the template minutia l bounding box can be computed as follows:

$$condition = \begin{cases} true, & if \begin{cases} radiu\_low[k] < \left(r_l^p - r_k^q\right) < radius\_high[k] \\ angle\_low < \Delta e < angle\_high[k] \\ \Delta\theta < \varepsilon \\ rot[k][l] < 400 \end{cases} \\ false, & otherwise \end{cases} \tag{2.42}$$

$$\Delta e = \begin{cases} a, & if\ \left(a = \left(e_L^P - e_k^Q + 360\right)\ mod\ 360\right) < 180 \\ a - 180\ , & otherwise \end{cases} \tag{2.43}$$

$$\Delta\theta = \begin{cases} a, & if\ \left(a = \left(\theta_L^P - \theta_k^Q + 360\right)\ mod\ 360\right) < 180 \\ a - 180, & otherwise \end{cases} \quad (2.44)$$

where $\varepsilon = 5°$ is a predefined constant.

6. Repeat the algorithm from step 3 for all minutiae of the input and template set. The maximum score from all the reference minutiae points is taken as the matching score.

2.3 Non-Minutiae Feature-based Matching

In non-minutiae feature based matching, features other than minutiae are used for matching like local orientation, local frequency, sweet pores, ridges shape etc. Correlation based matching is also one of the non-minutiae method which align the fingerprint images to see whether their ridges and valleys are aligned or not. Most of the other non-minutiae techniques use the same method as correlation-based method. They align and determine whether the features are matched or not [4] [17] [13].

Some of the reasons why we need other than minutiae feature is

1. Other features can be used together with minutiae to improve the efficiency.
2. Extracting minutiae from extremely poor quality fingerprint image is difficult.
3. When the fingerprint image is small, non-minutia based matching method can perform better [13].

2.3.1 Filterbank-Based Fingerprint Matching

One of the non-minutiae based matching which was proposed by Jain A.K, Prabhakar.S, Hong.L and Pankanti.S is Filterbank-Based Fingerprint Matching [8]**.** In filterbank based matching there is no minutiae extraction. Features are extracted directly from the gray scale fingerprint image. The first step for feature extraction is the determination of the region of interest around a reference point (core point). They define the reference point as the point of maximum curvature of the concave ridges in fingerprint images (see figure 2.19).

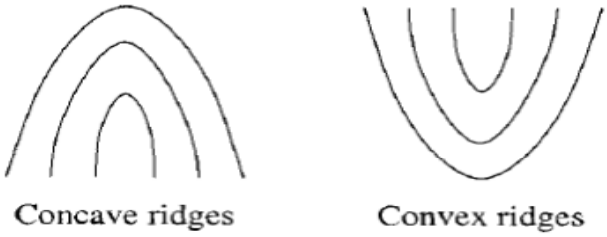


Figure 2.19 Concave and convex ridges in a fingerprint image when the finger is positioned upright [8].

To find the reference point, local orientation is computed for each pixel of the gray scale image and the sine component of the local orientation is taken. Then, for each pixel, the sine component is integrated over two different regions (see figure 2.20) and the value of their difference is assigned to the corresponding pixel. The regions designed and applied iteratively to capture the maximum curvature. The maximum value after a number of iteration will be used as a reference point.

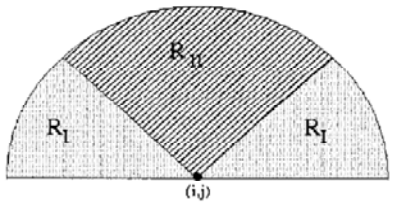


Figure 2.20 Regions for integrating the sine component for pixel (i,j) [8].

After determining the region of interest, the region is tessellated into concentric circles made up of 80 sectors (see figure 2.21). If $I(x, y)$ denotes the gray scale image of $m\ x\ n$ size then the sectors are computed using the radius r and angle γ as follows,

$$S_i = \{(x, y) | b(t_i + 1) \leq r < b(t_i + 2), \gamma_i \leq \gamma < \gamma_{i+1}, 1 \leq x \leq m, 1 \leq y \leq n\} \quad (2.45)$$

where

$$t_i = i\ div\ k$$

$$\gamma_i = (i\ mod\ k) \times \left(\frac{2\pi}{k}\right)$$

$$r = \sqrt{(x - x_c)^2 + (y - y_c)^2}$$

$$\gamma = \tan^{-1} \frac{(y - y_c)^2}{(x - x_c)^2}$$

$b = 20$ is the width of each band, $k = 16$ is the number of sectors considered in each band, and $i = 0 \ldots (B\ x\ k - 1)$, where $B = 5$ is the number of concentric bands considered around the reference point for feature extraction. Then each sector is normalized separately.

Figure 2.21 Reference point (x), the region of interest, and 80 sectors superimposed on a fingerprint [8].

The third step is filtering the tessellated region using Gabor filter in 8 directions. Filtering is done in the spatial domain using a 33 x 33 size mask. The frequency of the Gabor filter is set to the average ridges frequency which is 0.1 and the direction θ to (0^o, 22.5^o, 45^o, 67.5^o, 90^o, 112.5^o, 135^o, and 157.5^o) with respect to *x*-axis. Then for each sector of each 8-direction filtered images, the average absolute deviation from their mean (AAD) is computed.

Let $F_{i\theta}(x, y)$ be the θ-direction filtered image for sector S_i. Now, $\forall\, i \in \{0,1, \ldots, 79\}$ and $\theta \in \{0°, 22.5°, 45°, 67.5°, 90°, 112.5°, 135°, 157.5°\}$, the feature value $V_{i\theta}$ is the average absolute deviation from the mean defined as:

$$V_{i\theta} = \frac{1}{n_i}(\sum F_{i\theta}(x, y) - P_{i\theta}) \tag{2.46}$$

Where, n_i is the number of pixels in S_i and $P_{i\theta}$ is the mean of pixel values of $F_{i\theta}(x, y)$ in sector S_i. The Euclidean distance between these features of the two fingerprint images to be matched is used as matching score. The Euclidean distance is small for different fingerprint images that are from the same finger than that are from different fingers.

All the steps described above can be shown in diagram as shown below:

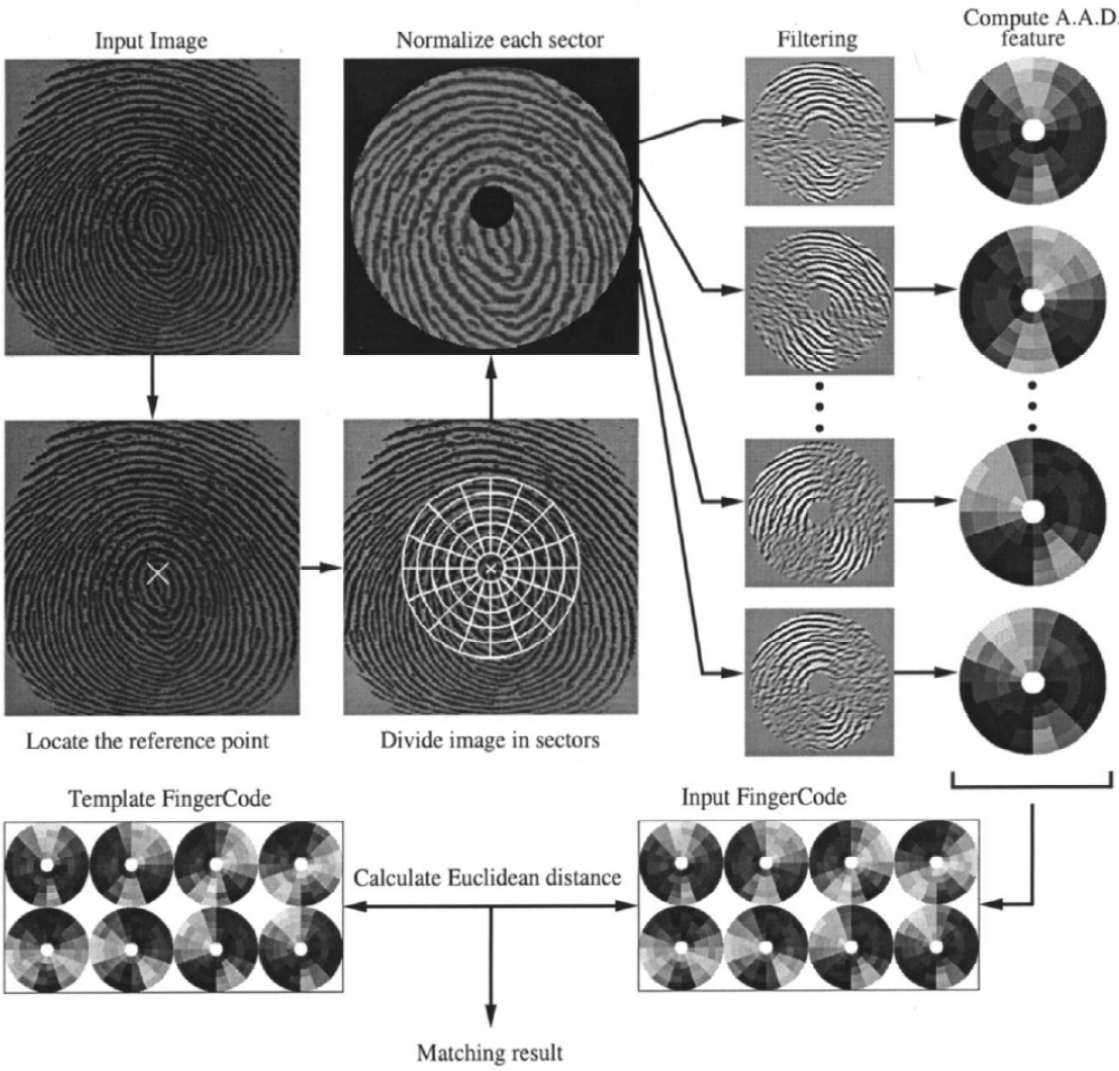


Figure 2.22 Filterbank based fingerprint matching system [8].

The authors tested the above matching system on two databases and the result of the algorithm on one of the databases is shown below:

Threshold value	False Acceptance Rate (%)	False Reject Rate (%)
30	0.10	19.32
35	1.07	7.87
40	4.59	2.83

Table 2.1 Result of filterbank based fingerprint matching system [8].

Jain et al. [8] conclude that the matching system has a verification accuracy which is only marginally inferior to the performance of a state-of-the-art minutiae-based fingerprint matcher which is already matured. Therefore, it can be developed more and also can be used with minutiae based matching system to improve the efficiency.

After the proposal of the above new state-of-the-art matching system, many other papers try to modify and improve the performance. The shortcoming of the above matching system is

1. Finding the core point which is difficult especially for arc type fingerprints. Jain et al [8] said that, about their proposed algorithm for finding the core point, “although this successfully detects the reference point in most of the cases, including double loops, the present implementation is not very precise and consistent for the arch type fingerprints”. Also, the method for singularity extraction by Poincare indexing described in section 2.2.4 fails to extract the core point if it is near the boundary of the fingerprint image.

2. If the core point is near the boundary of the fingerprint image, the tessellated region of interest will include areas that are not part of the finger print image. This will deteriorate the efficiency (see figure 2.23)

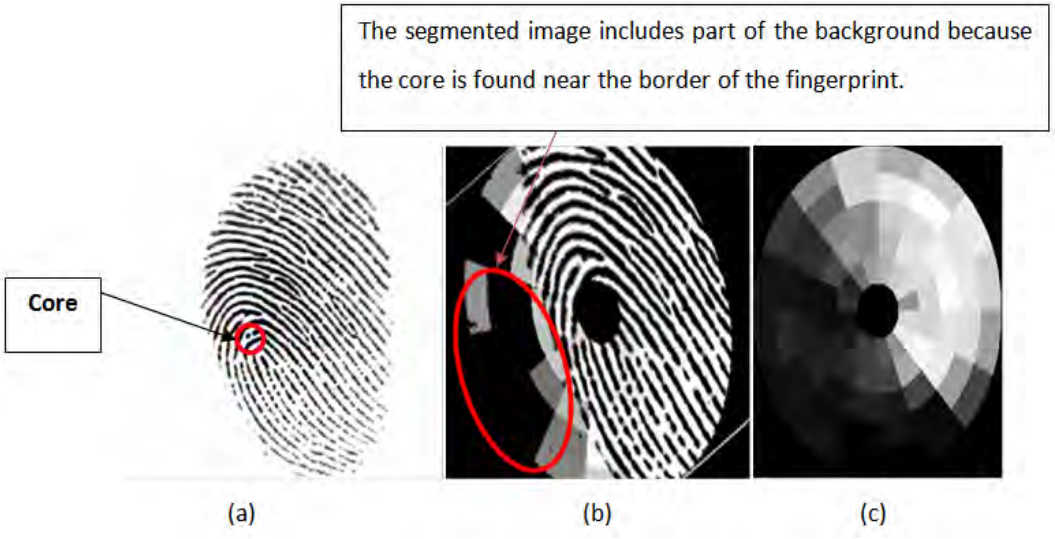


Figure 2.23 a) Fingerprint image, b) segmented area around the core and c) A.A.D feature of the segmented area.

Ross et al. [17] proposed another method to overcome the above shortcomings. First before feature extraction, the fingerprint images are enhanced using the method as described in section 2.2.5. Then the entire image is filtered with 8-direction Gabor filter. Each of the eight filtered images is partitioned into 16 x 16 square pixels which is twice the size of the ridges distance. After that, the variance of each square pixel is computed. Therefore, there will be eight variance images called feature map. Finally, the Euclidean distance between these feature maps, of the two fingerprint images to be matched, is used as the matching sore. In the above proposed method, it solves the problem of finding reference point (core point) and use the entire image which also solve the second problem. The authors aligned the two fingerprint image feature maps using the affine transformation (rotation and translation) parameters determined from the minutiae based matching system. This feature map based matching was hybrid with minutiae based matching and improves the equal error rate from ~14% to ~4.5%. The shortcoming of the above method is in the alignment process. Taking affine transformation from the minutiae based method makes the feature map based matching system dependant on the first one.

2.4 Fingerprint recognition system performance evaluation

Even if fingerprint recognition system is a matured biometric recognition it certainly makes wrong decisions. Some of the reasons are [13]:

- Information limitation: for example due to poor quality finger print images.
- Representation limitation: i.e. features used to represent the fingerprint image.
- Invariance limitation: i.e. images of different fingerprint may look similar.

The following method is common for evaluating performance of fingerprint recognition system which is adapted from FVC 2002 report [12]. Let the database contain 8 samples of 10 different fingers. $\boldsymbol{Fs}_{ij}$ is the j^{th} fingerprint sample of the i^{th} fingerprint and its corresponding template $\boldsymbol{Ts}_{ij}$.

1. The templates $\boldsymbol{Ts}_{ij}$ are computed from the corresponding fingerprint samples and stored in another database.

2. For each finger, each sample is matched against each other (i.e. the samples must be from the same finger). If $\boldsymbol{Ts}_{ij}$ is the template it will be matched with fingerprint $\boldsymbol{Fs}_{ik}$ ($i < k \leq 8$) and the scores called genuine matching scores (**gms**$_{ik}$) are stored. The number of genuine recognition attempt (**NGRA**) is $\frac{7\times8}{2} \times 10 = 280$.

3. The first samples from each fingerprints matched against each other (i.e. the samples are from different fingers). If $\boldsymbol{Ts}_{i1}$ is the sample template it will be matched with fingerprint image $\boldsymbol{Fs}_{ki}$ ($i < k \leq 10$) and the scores called impostor matching scores (***ims***$_{ik}$) will be stored. The number of impostor recognition attempt (**NIRA**) is $\frac{9\times10}{2} = 45$.

4. The **FMR**(t) (False Match Rate, also sometimes called False Acceptance Rate) and **FNMR**(t) (False Non-Match Rate, also sometimes called False Rejection Rate) curves are computed for threshold t ranging from 0 to 1. Given a threshold *t,* **FMR**(t) denotes the percentage of **ims**$_{ik} \leq$ t (i.e percentage of the number of samples that are matched even if they are from different fingers), whereas **FNMR**(t) denotes the percentage of **gms**$_{ijk}$ < t (i.e. percentage of the number of samples that are not matched even if they are from the same finger).

$$\mathbf{FMR}(t) = \frac{\{\boldsymbol{ims}_{ik} | \boldsymbol{ims}_{ik} \geq t\}}{\mathbf{NIRA}}, \tag{2.47}$$

$$\mathbf{FNMR}(t) = \frac{\{\boldsymbol{gms}_{ijk} | \boldsymbol{gms}_{ijk} < t\}}{\mathbf{NGRA}} \tag{2.48}$$

5. A **ROC** (Receiving Operating Curve) or **DET** (Detection-Error Tradeoff) is given where **FNMR** is plotted as a function of **FMR**; the curve is drawn in log-log scales for better comprehension.

6. The Equal Error Rate EER is computed as the point where **FNMR**(t) = **FMR**(*t*); in practice the score distributions are not continuous and a crossover point might not exist. In this case, the interval [EER_{low}, EER_{high}] is used.

CHAPTER THREE : MODEL BULIDING

3.1 The Proposed System

In this paper, a hybrid matching algorithm is proposed. The first part of the hybrid matching algorithm is a minutiae matching algorithm which is described in section 2.2. The other matching algorithm is a new proposed method that solves the shortcomings of the filterbank based matching method, which was also described in section 2.3.1. The general hybrid matching algorithm can be depicted using the flow chart as follows:

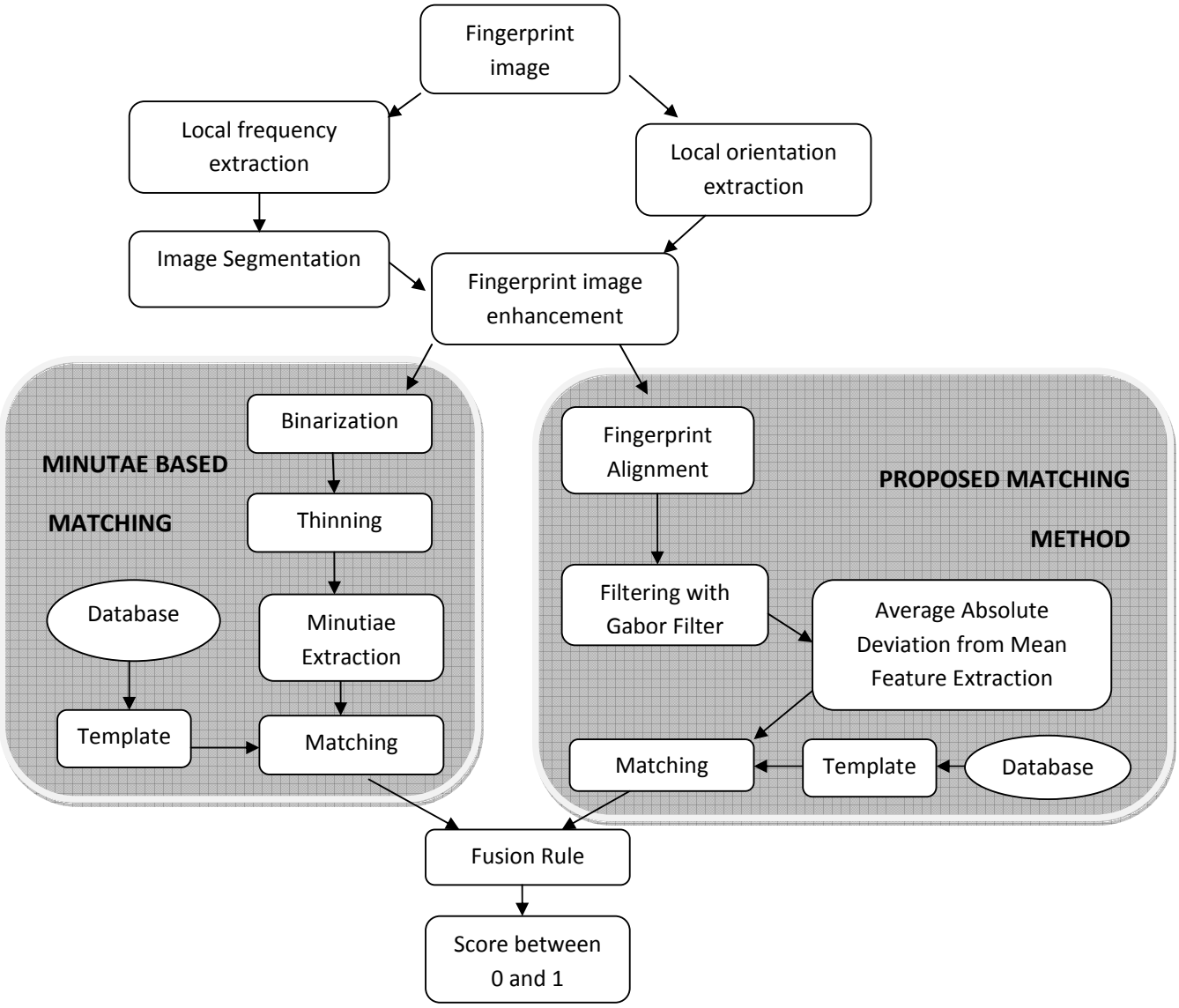


Figure 3.1 Flow chart of the proposed system.

Both matching algorithms use the enhanced image. The local frequency extraction, local orientation extraction, image segmentation, enhancement method and the process in minutiae matching methods are described in section 2.2.

Figure 3.2 Fingerprint images from input and template database.

The steps of the proposed matching method are given below:

1. In the fingerprint alignment process, divide the local orientation image (see figure 3.3) into square blocks $(w \times w)$. Compute the mean value and assign to the corresponding block. If the local orientation image size was $m\ x\ n$, the new computed mean orientation image will have $ceil(\frac{m}{w}) \times ceil(\frac{n}{w})$ size (see figure 3.4). Where, the function $ceil(x)$ round x to the next integer. For each pixel of the input and template mean orientation images, align the two mean images and find a pixel pair that gives maximum orientation matches within the overlapped region.

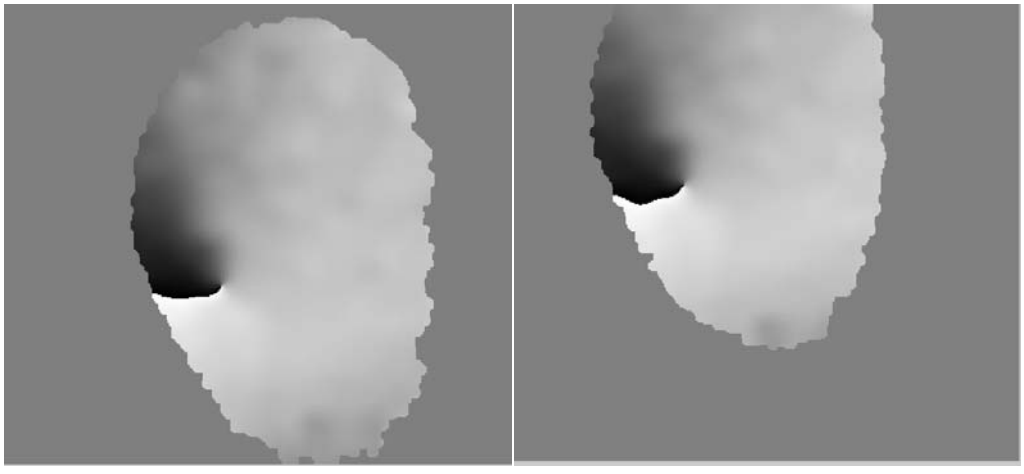


Figure 3.3 Local orientation images of the input and template images shown in figure 3.2.

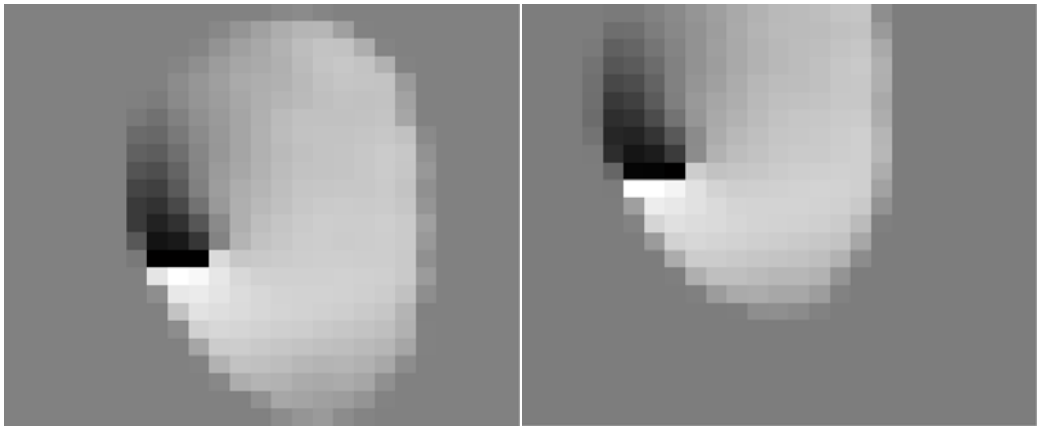


Figure 3.4 Mean orientation images of the input and template local orientation images shown in figure 3.3.

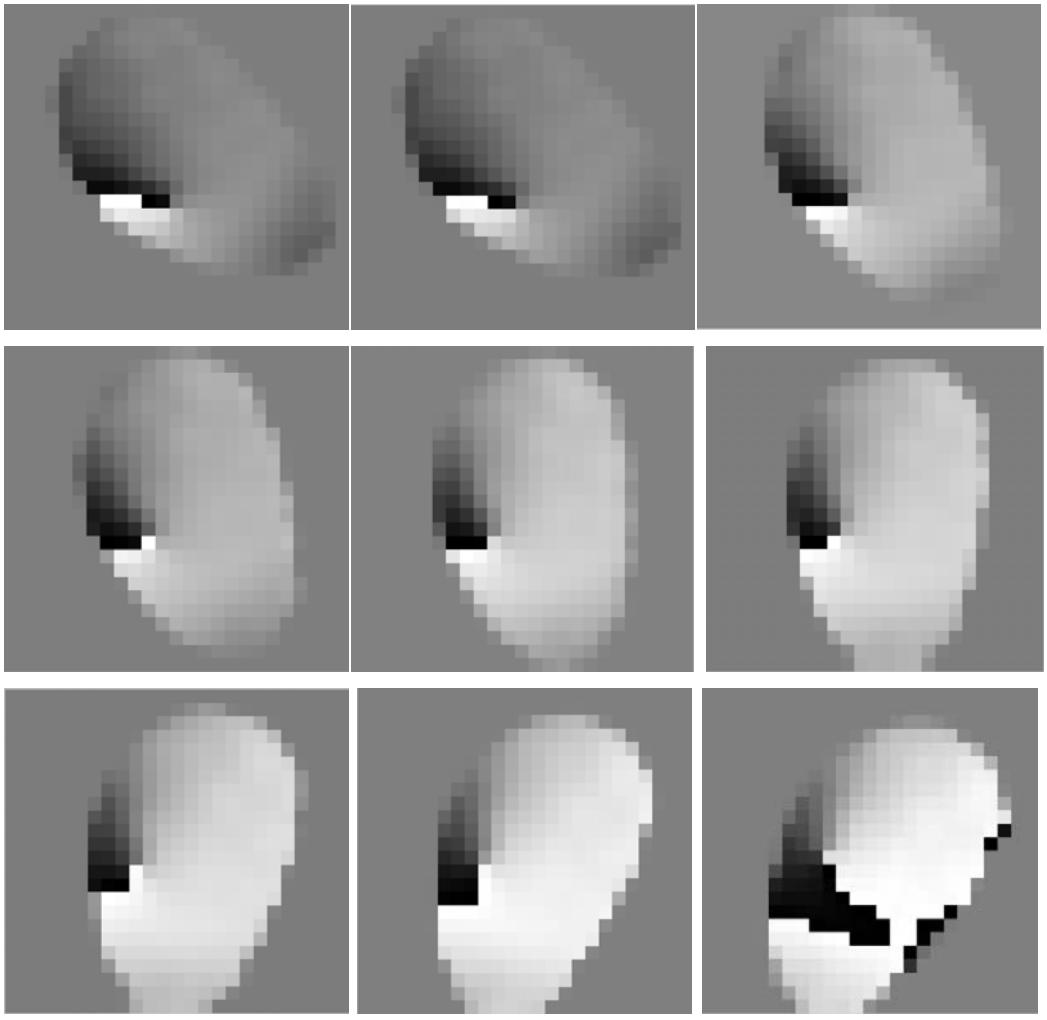


Figure 3.5 Mean orientation images of the input image after the input image rotated in nine different directions.

2. Repeat the above procedure first by rotating only the input image in 9 different directions $(40°, 30°, 20°, 10°, 0°, -10°, -20°, -30°, -40°)$ (see figure 3.5). And choose the pixel pair and direction that give the maximum orientation match. Using these values, align the two fingerprint images (see figure 3.6).

Figure 3.6 Alignment of the input and template images shown in figure 3.2.

3. In filtering with the Gabor filter process, enhance the input and the template fingerprint images as discussed in section 2.2.5.

4. Using two threshold values $T1$ and $T2$ $(T1 < T2)$, pixels of the enhanced images that have value greater than $T2 = 1100$ are assigned to 1 and less than $T1 = 1000$ are assigned to 0 (see figure 3.7).

Figure 3.7 Enhanced images of the input and template images which have a value between 1 and 0.

5. In average absolute deviation from mean feature extraction process, first filter each image using 8 different directed (

   Gabor filters with mean local frequency as discussed in section 2.3.1. This will result 8 filtered images for each input and template images (see figure 3.8 for the input image).

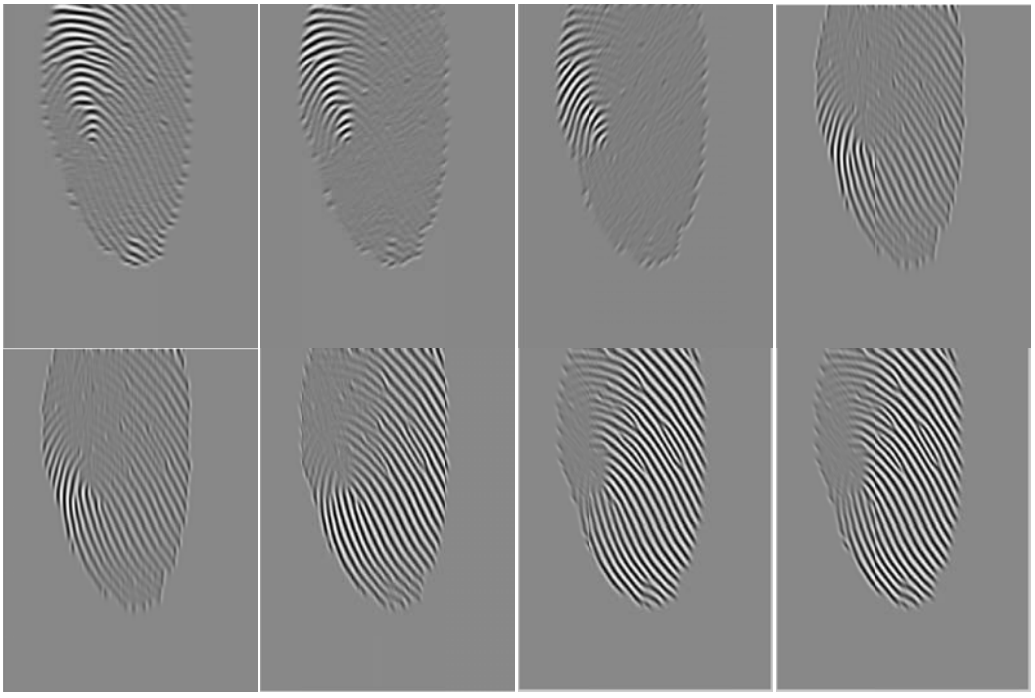


Figure 3.8 the filtered images of the input image shown in figure 3.2 in 8 different directions by Gabor filter.

Then divide these images into square blocks $w\ x\ w$. Compute the average absolute deviation from the mean (AAD) and assign to the corresponding blocks.

$$AAD_j = \frac{1}{wxw}\sum_{i=1\ldots w\times w}\left|I_{ij}(x,y) - M_j\right| \quad (3.1)$$

Where $I_{ij}(x,y)$ is the i^{th} pixel of j^{th} block, M_j & AAD_j are the mean and AAD of the j^{th} block of the image respectively (see figure 3.9 for the input image).

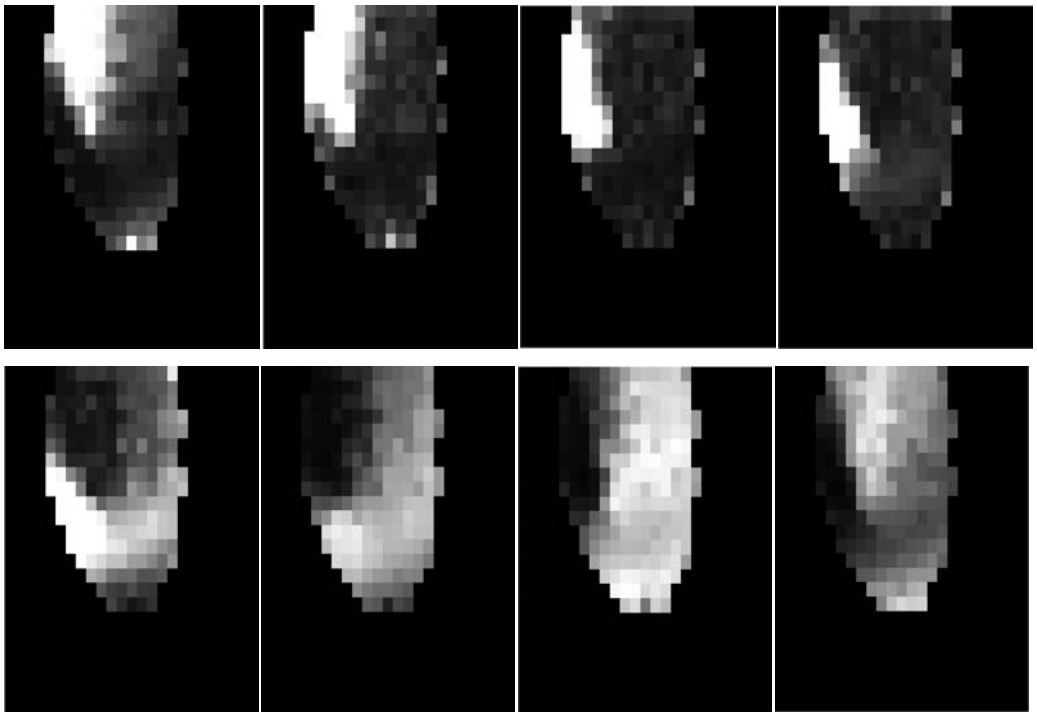


Figure 3.9 the average absolute deviation from the mean (AAD) feature images of the input image shown in figure 3.2.

6. In matching process, compute the Euclidian distance between AAD images of the input and the template. Fingerprints that are from the same finger have small distance than those from different fingers.

Rather than searching all the direction, rotating the input image in small number of different direction is sufficient. Also, dividing the local orientation image into blocks and using the

mean of the blocks, reduces the time consumption by the algorithm than using the entire image. The blocks size was 16x16 square pixels and can align two fingerprints with accuracy within this square region (see figure 3.10).

(a) (b)

(c)

Figure 3.10 a) Input image b) template image c) alignment of the two images.

Mean values are not discriminative as minutiae. This is because the same ridges that are found in different impression of the same finger do not have equal gray scale values. This makes it difficult to compare AAD directly, that are from the gray scale images. Therefore, to avoid the problem after the enhancement, the ridges are assigned to be near 0 and valleys to be near 1 (step 5).

To capture the local characteristics, the enhanced images are filtered with Gabor filters and blocks of the filtered images are used rather than the entire images.

The shortcomings of the filterbank based method and also the problem of one of the modified methods (feature map based method) was discussed in section 2.3.1. The above proposed method solves the discussed shortcomings:

1. By aligning the input and template images using the local orientations rather than using the core point as in paper [8] and using affine transformation taken from minutiae based matching as in paper [17].
2. It uses the entire overlapped region of the two images rather than small part of the image as in paper [8].
3. The AAD features are made more discriminative by assigning the ridges value near 0 and valleys near 1.

CHAPTER FOUR : RESULT AND CONCLUSION

4.1 Result

The fingerprint images were taken from Fingerprint Verification Competition (FVC) 2002 databases of set B, which are freely available. The type of sensor used to capture and the number of the fingerprint images in the FVC 2002 databases of set B are shown below [12].

	Technology	Scanner	Image Size	Set B (wxd)	Resolution
DB1	Optical	Identix TouchView II	388x374	10x8	500 dpi
DB2	Optical	Biometrika FX2000	296x560	10x8	569 dpi
DB3	Capacitive	Precise Biometrics 100 SC	300x300	10x8	500 dpi
DB4	Synthetic Generator	SFinGE v2.51	288x384	10x8	500 dpi

Table 4.1 Scanners/technologies used for the collection of FVC 2002 databases [12].

First, each matching algorithms parameters were tuned using database 1 of set B which consist 80 fingerprint images (i.e. 8 fingerprint samples of 10 fingers). Then, all the set B databases were used for testing the matching algorithms (i.e. 8 fingerprint samples of 40 fingers, which is a total of 320 fingerprint images).

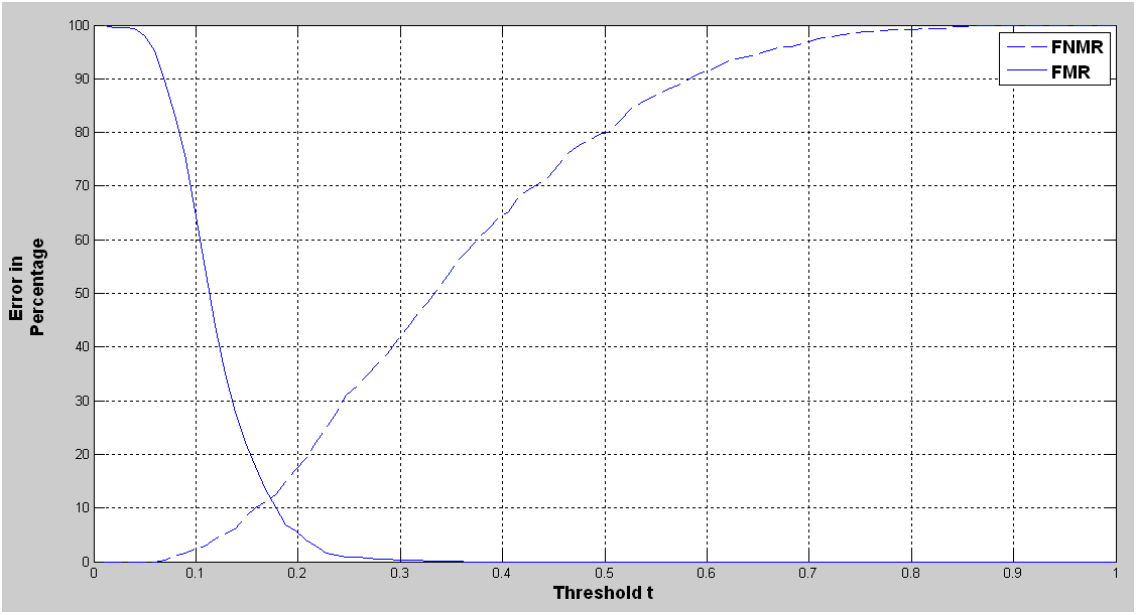
The experiment was conducted on a computer which has Intel®Core™2 Duo CPU @2.93GH and 1.96 GB RAM, running Microsoft Windows XP Professional Version 2002 with Service pack 3. The software used for programming the code is MATLAB R2010a.

The detailed methods for performance evaluation were discussed in section 2.4. Here, only the result of each fingerprint matching methods will be discussed.

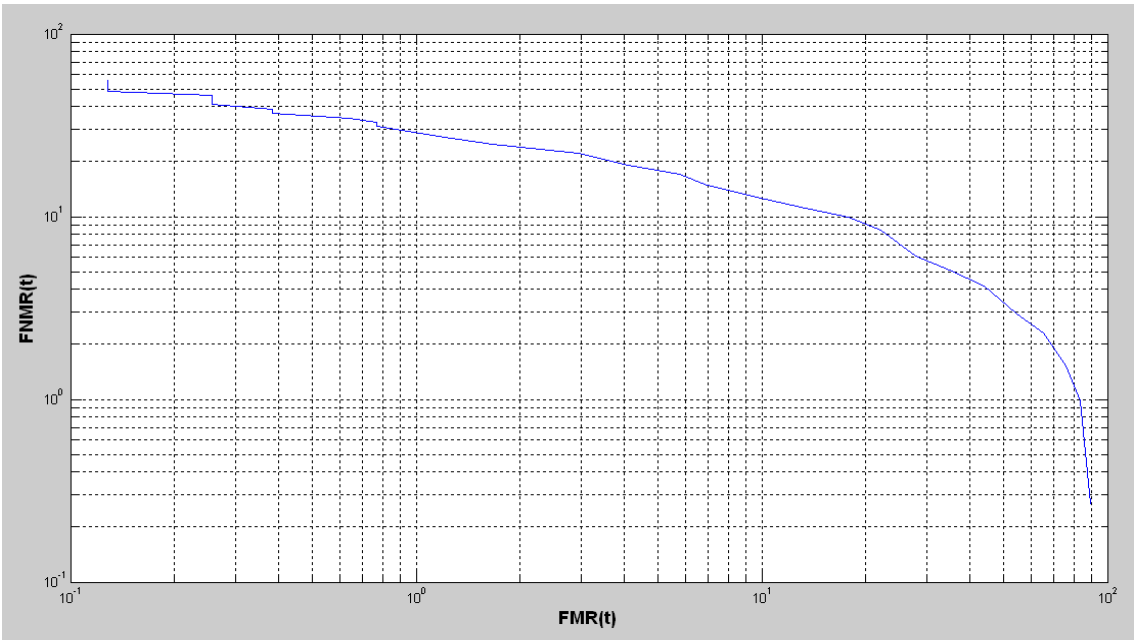
The False Matching Rate (FMR) and False Non-Match Rate (FNMR) curves with a given threshold t and the Detection-Error Tradeoff [DET] curve of the minutia based matching algorithm are shown in figure 4.1. FMR curve show that, for a given threshold t, the percentage of number of fingerprints that are matched even if they are from different

fingers. The other curve FNMR shows that, the percentage of fingerprints that are not matched even if they are from the same finger.

The point where FMR and FNMR curves meet is called Equal Error Rate (EER). As it can be seen from the figure 4.1 (a), the EER is $\cong 11.8\%$ at a threshold $t \cong 0.17$. If we take a new threshold greater than t, the rate of matched fingerprints decreases even though they are from different fingers while the rate of non-matched fingerprints increases even if they are from the same finger. Therefore, EER is the point where the error is minimized for both curves. The tradeoff between FMR and FNMR can be see clearly in DET curve, if we plot FMR against FNMR for different threshold t using log-log scales as shown in figure 4.1 (b).



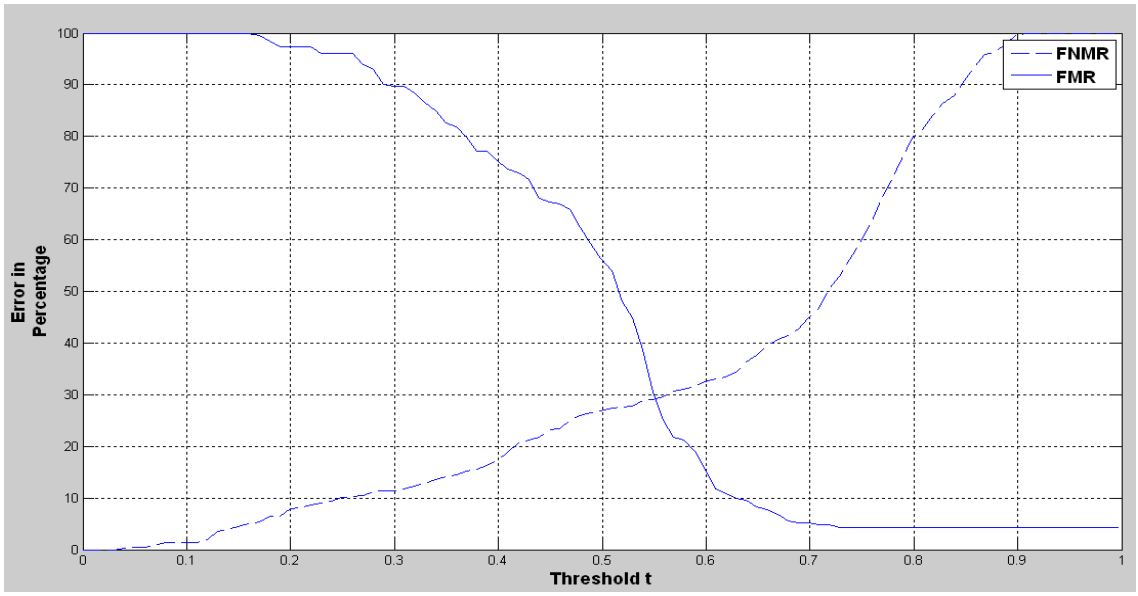
(a)



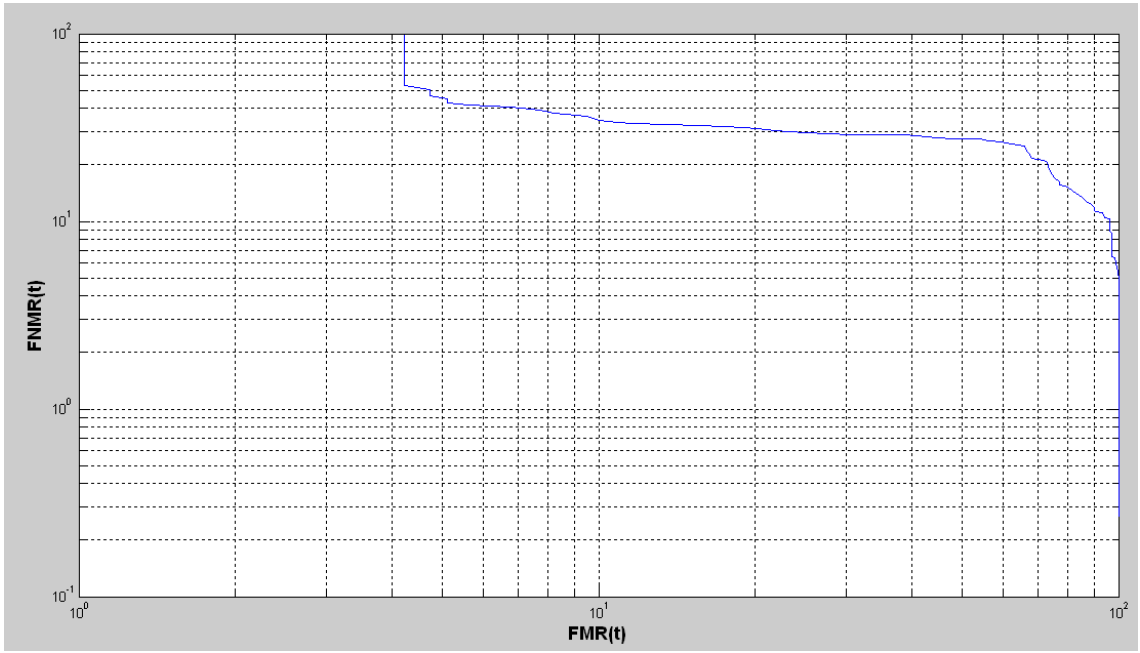
(b)

Figure 4.1 (a) The FMR and FNMR curves and (b) The DET curve of minutiae based matching algorithm.

The FMR, FNMR and the DET curves, for filterbank based matching with EER $\cong 29.2\%$ at $t \cong 0.55$ and also for the proposed matching algorithm with EER $\cong 21.2\%$ at $t \cong 0.81$, are shown in figure 4.2 and 4.3 respectively.

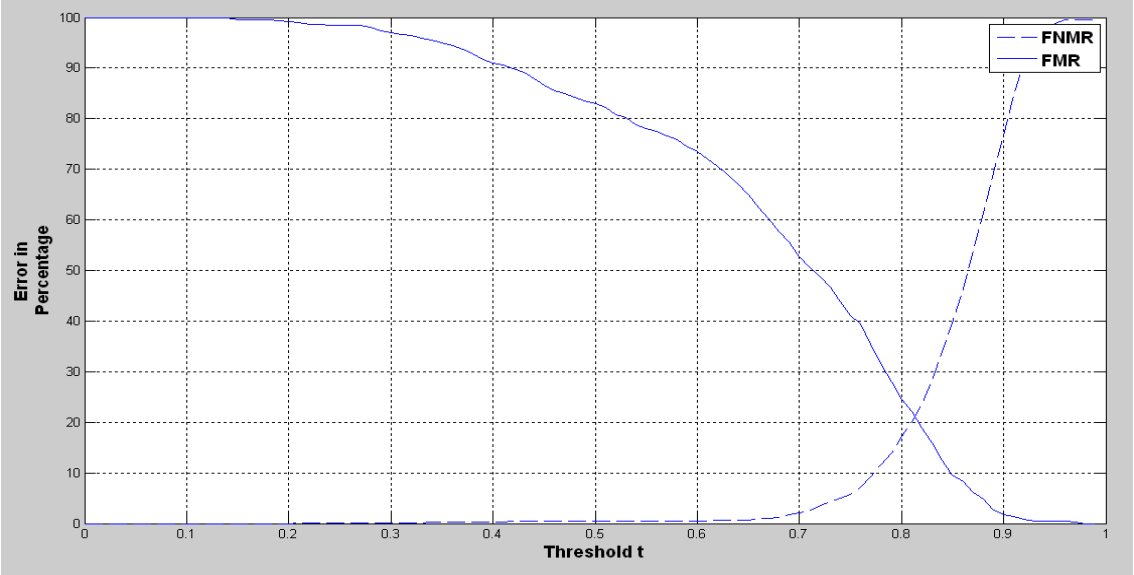


(a)

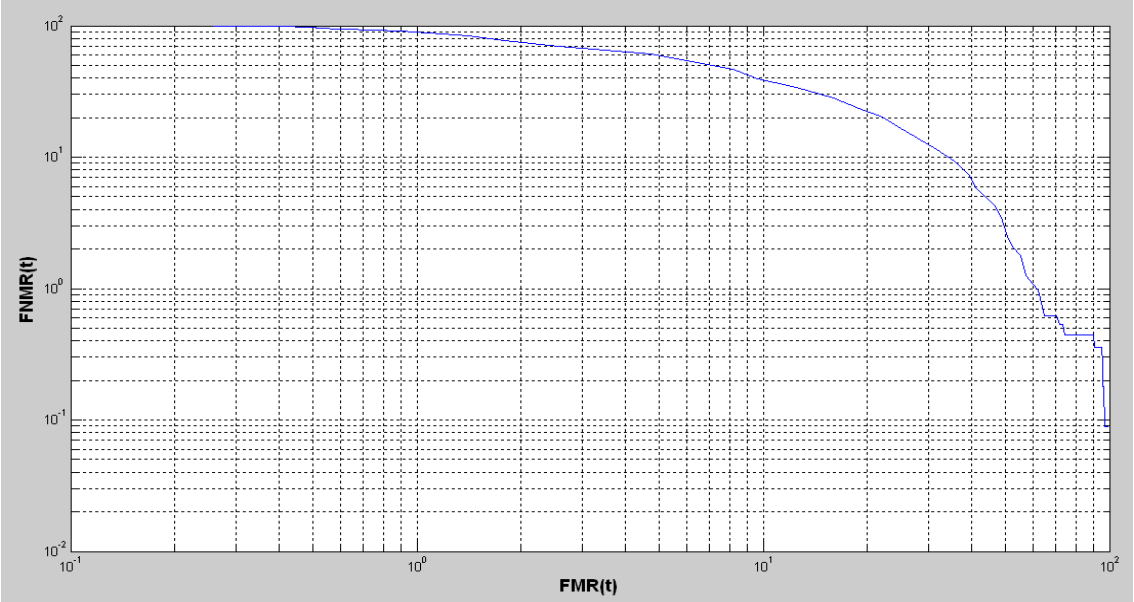


(b)

Figure 4.2 (a) The FMR and FNMR curves and (b) the DET curve of filterbank based matching algorithm.



(a)

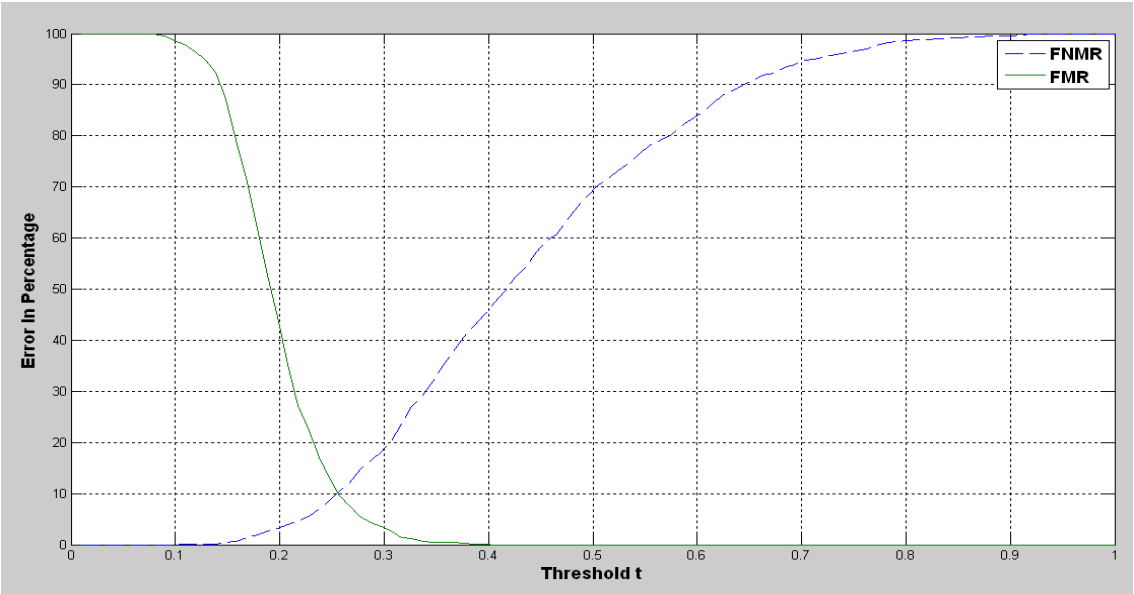


(b)

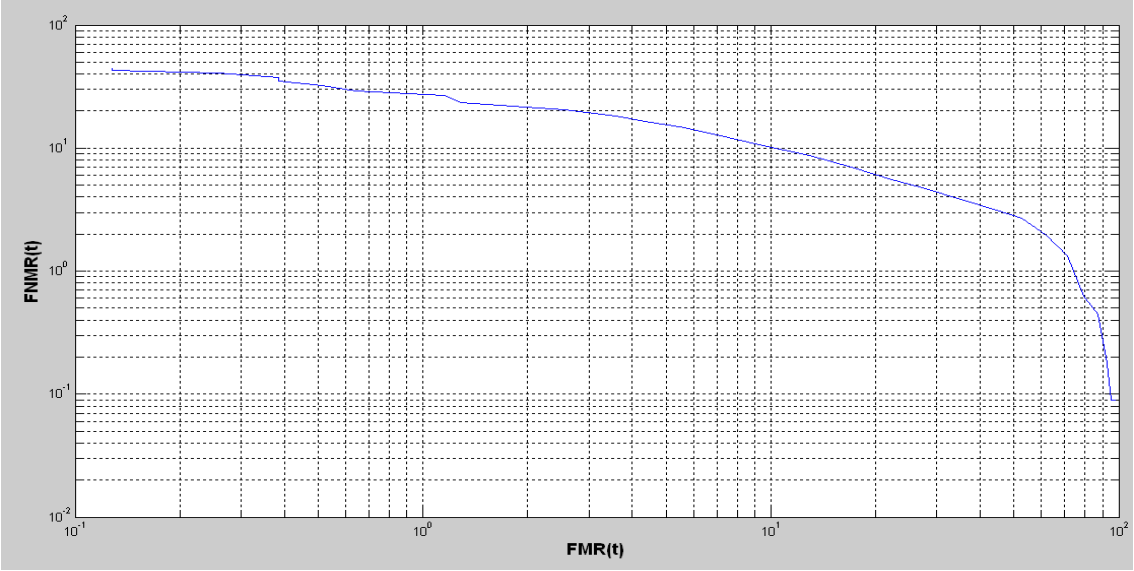
Figure 4.3 (a) The FMR and FNMR curves and (b) The DET curve of the new proposed matching system.

In hybrid matching algorithms, the decision is made by combining the similarity scores of the individual matching algorithms using fusion rules, such as max and sum rules [1]. In this paper, the fusion rule $S = \alpha S_M + (1-\alpha)S_{NM}$ is used [17]. Where, S_M and S_{NM} are the minutiae and non-minutiae based matching similarity scores.

The FMR, FNMR and the DET curves, for the hybrid minutiae-filterbank based matching with EER $\cong 10.12\%$ at $t \cong 0.25, \alpha \cong 0.81$ and also, for the hybrid minutiae-proposed algorithm having EER $\cong 7.55\%$ at $t \cong 0.40, \alpha \cong 0.63$ are shown in figure 4.4 and 4.5 respectively.

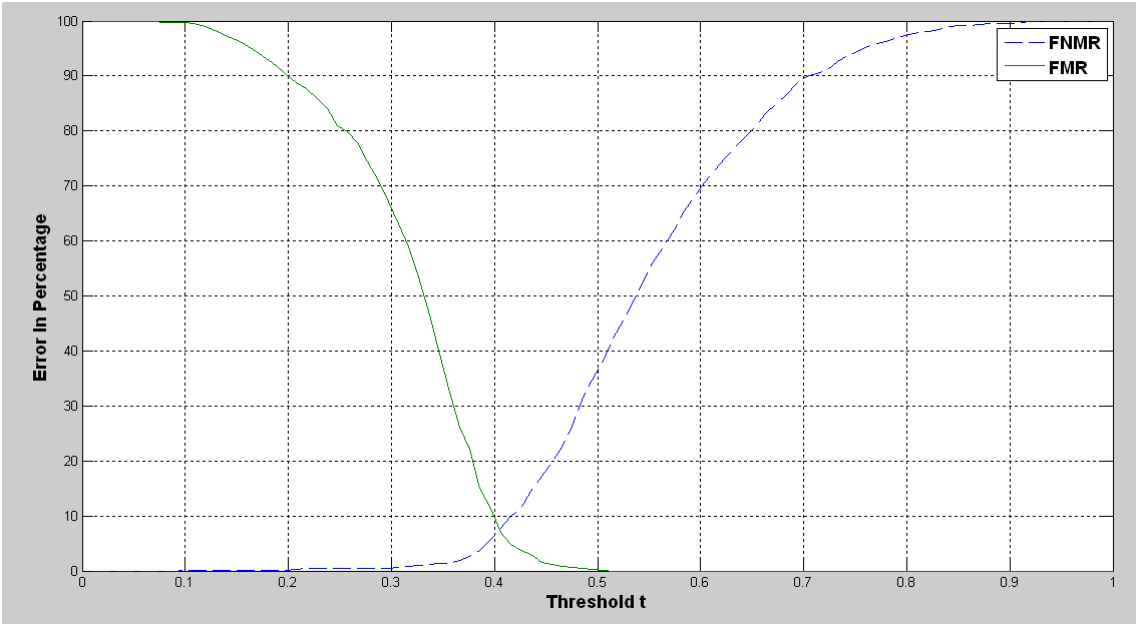


(a)

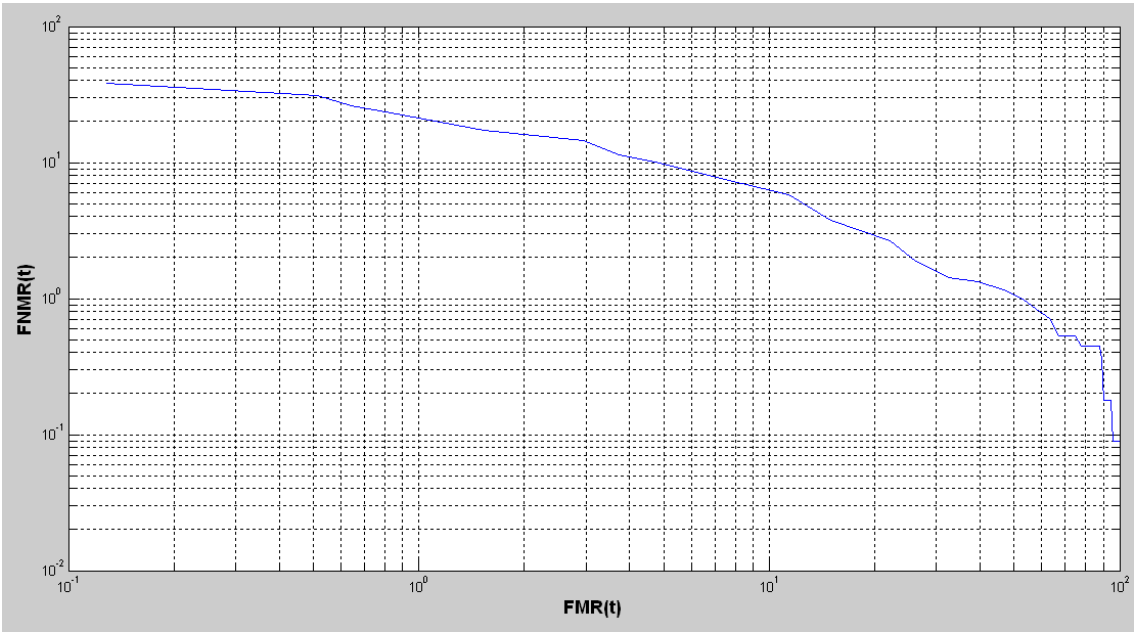


(b)

Figure 4.4 (a) The FMR and FNMR curves and (b) The DET curve of the hybrid of minutiae and filterbank matching system.



(a)



(b)

Figure 4.5 (a) The FMR and FNMR curves and (b) The DET curve of the hybrid of minutiae and the new proposed matching system.

The DET curves of all fingerprint matching systems are shown below. Where, the symbols M, F, P, MP and MF in the legend of the figure 4.6 stands for minutiae, filterbank, proposed

system, hybrid of minutiae-proposed system and hybrid of minutiae-filterbank based matching systems respectively.

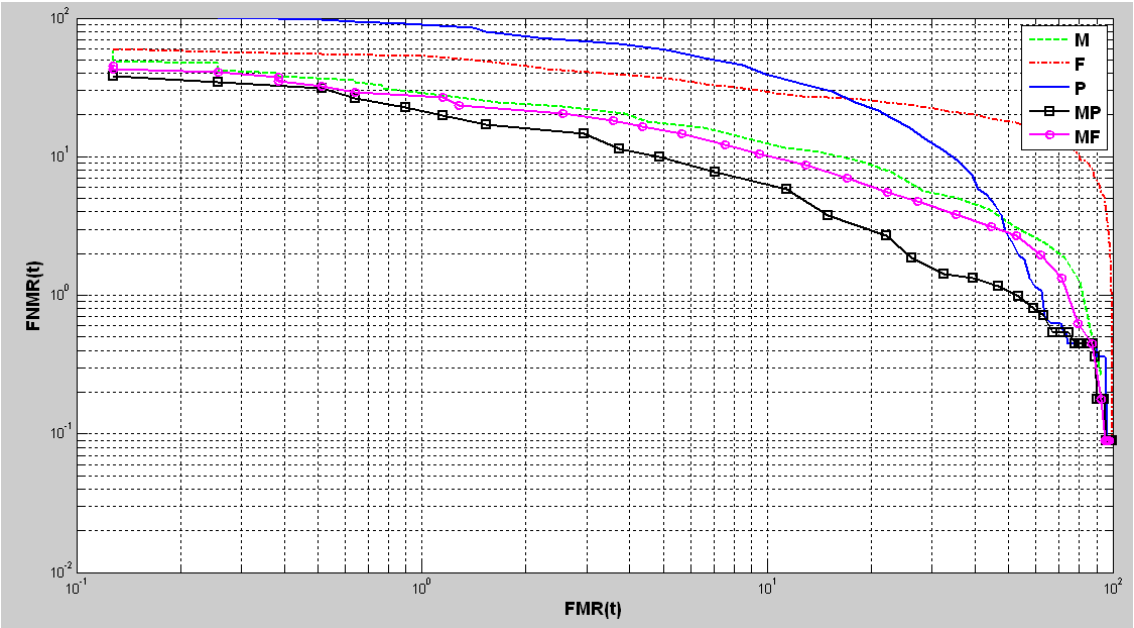


Figure 4.6 The DET curves of all fingerprint matching systems.

The performance evaluation results can be put in a table as follows:

Matching system	EER (in %)	Average enroll time (in Sec)	Average match time (in Sec)	Average Template size (K bytes)	Average Image size (K bytes)
Minutiae based	11.8	5.74	0.03	27.99	125
Filterbank based	29.2	15.7	0.08	4	125
The proposed system	21.2	23.5	14.6	126.124	125
Hybrid of minutiae and filterbank	10.12	21.44	0.11	31.99	125
Hybrid of minutiae and the proposed system	7.55	29.24	14.63	130.124	125

Table 4.2 The results of algorithm performance evaluation of all fingerprint matching systems.

4.2 Conclusion

Minutiae based matching method involves many processing steps. The errors in each processing step propagate to the next step. This problem decreases the performance of the matching method. Especially, beyond the error introduce in the matching method, the quality of the fingerprint image and the errors introduced in binarization and thinning methods highly affect the performance. The minutia matching method used here, tolerate, to some extent, the errors introduced in each processing steps by using a changeable size bounding box around each template minutiae.

In non-minutiae based methods the features used are the average absolute deviation from the mean. Even if, these features are not very discriminative like minutiae, they can be used to increase the performance of fingerprint matching system. As we can see from the table 4.2, the hybrid matching algorithms have low EER than the individual algorithms.

In this paper, new non-minutiae based matching method is proposed that solve the short comings of those methods proposed in the papers [8] and [17]. Also, the hybrid of the proposed method with minutiae based method improves the EER.

4.3 Recommendation

In the binarization and thinning methods of minutiae based matching, much information are lost from the gray scale fingerprint images. Therefore, other methods that extract minutiae directly from the gray scale can be used in the future work.

Also, in non-minutiae based matching, the features used are the average absolute deviation from the mean. These features are not very discriminative like minutiae. Therefore, finding a feature that is more discriminative, that don’t need many preprocess to extract it and that don’t need high resolution scanner like sweet pores can be the extension of this work.

BIBLIOGRAPHY

[1] Alonso-Fernandez F., Fierrez-Aguilar J. and Ortega-Garcia J., "An Enhanced Gabor Filter-Based Segmentation Algorithm for Fingerprint Recognition Systems," in *Proc. Int. Symp. On Image and Signal Processing and Analysis*, 2005.

[2] Bazen A.M. and Gerez S.H, "Directional Field Computation for Fingerprints Based on the Principal Component Analysis of Local Gradients," *ProRISC 2000 Workshop on Circuits, Systems and Signal Processing, Veldhoven, Netherlands, 2000.*

[3] Bazen A.M. and Gerez S.H., "Systematic methods for the computation of the directional fields and singular points of fingerprints," *IEEE Transactions on Pattern Analysis Machine Intelligence*, vol. 24, no. 7, pp. 905–919, 2002.

[4] Bazen A.M., Verwaaijen G.T.B., Gerez S.H., Veelenturf L.P.J. and van der Zwaag B.J., "A Correlation-Based Fingerprint Verification System," in *Proc. Workshop on Circuits Systems and Signal Processing*, 2000.

[5] Gonzales R.C. and Woods R.E., Digital Image Processing, 3rd edition,Prentice-Hall, Englewood Cliffs, NJ, 2007.

[6] Hong L., Wan Y. and Jain A.K., "Fingerprint image enhancement: Algorithms and performance evaluation," *IEEE Transactions on Pattern Analysis Machine Intelligence*, vol. 20, no. 8, pp. 777–789, 1998.

[7] Jain A.K., Hong L., Pankanti S. and Bolle R., "An identity authentication system using fingerprints," *Proceedings of the IEEE*, vol. 85, no. 9, pp. 1365–1388, 1997.

[8] Jain A.K., Prabhakar S., Hong L. and Pankanti S., "Filterbank-based fingerprint matching," *IEEE Transactions on Image Processing*, vol. 9, pp. 846–859, 2000.

[9] Lu H., Jiang X. and Yau W.Y., "Effective and Efficient Fingerprint Image Post processing," *Laboratories for Information Technology*.

[10] Luo X., Tian J. and Wu Y., "A Minutia Matching Algorithm in Fingerprint Verification," in *Proc. Int. Conf. on Pattern Recognition (*15*th)*, vol. 4, pp. 833–836, 2000.

[11] Maio D. and Maltoni D., "Direct gray-scale minutiae detection in fingerprints," *IEEE Transactions on Pattern Analysis Machine Intelligence*, vol. 19, no. 1, 1997.

[12] Maio D., Maltoni D., Cappelli R., Wayman J.L. and Jain A.K., "FVC2002: Second Fingerprint Verification Competition," in *Proc. Int. Conf. on Pattern Recognition (*16*th)*, 2002b.

[13] Maltoni D., Maio D., Jain A.K. and prabhakar S., "Handbook of Fingerprint Recognition," *Second Edition*, *Springer*, 2009.

[14] McAndrew A., "An Introduction to Digital Image Processing with Matlab," Notes for SCM2511 Image Processing 1, Semester 1, 2004, School of Computer Science and Mathematics Victoria University of Technology.

[15] Meng X., Wu Z. and Zhao Y., "New Algorithm of Automation Fingerprint Recognition," *Proceedings of the IEEE International Conference on Automation and Logistics Qingdao, China September,* 2008.

[16] Ming X., Xiaoping W. and Quanping H., "Algorithm Based on Point Feature for Fingerprint Image Segmentation," *IDOSI Publications,* 2009.

[17] Ross A., Jain A.K. and Reisman J., "A hybrid fingerprint matcher," *Pattern Recognition*, vol. 36, no. 7, pp. 1661–1673, 2003.

[18] Tico M., Kuosmanen P., "An Algorithm for Fingerprint Image Post Processing." *Signals, systems and computers*, vol.2, no.2, pp.1735-1739, Aug.2002

[19] Zhau F., Tang X., "Preprocessing and Post Processing for Skeleton-Based Fingerprint Minutiae Extraction," *Pattern Recognition*, vol.40, no.4, pp.1270-1281, Apr.2007

DECLARATION

I, the undersigned, hereby declare that this thesis is my original work performed under the supervision of Dr. Assefa Dagne, has not been presented as a thesis for a degree program in any other university and all sources of materials used for the thesis are duly acknowledged.

Name: Biniyam Melese **Signature: _____**

Place: Addis Ababa

Date of submission: August, 2003 E.C

This thesis has been submitted for examination with my approval as a university advisor.

Advisor's Name: Dr. Assefa Dagne **Signature: _____**