B.Tech Project

Fingerprint Analysis for Biometric Authentication

By

Ashish Kumar

200101112

Dhirubhai Ambani Institute of Information &

Communication Technology

Gandhinagar, GUJARAT

April,2005

Dhirubhai Ambani Institute of Information &

Communication Technology

Gandhinagar, GUJARAT

CERTIFICATE

This is to certify that the Project Report titled "**Fingerprint Analysis for Biometric Authentication**" submitted by **Ashish Kumar**, ID **200101112** for the partial fulfillment of the requirements of B.Tech(ICT) degree of the institute embodies the work done by him on campus under my supervision.

Date: _____ Signature: _____

(Prof. Suman K Mitra)

Date: _____ Signature: _____

(Prof. Asim Banerjee)

Date: _____ Signature: _____

(Prof. Anil K Roy)

Abstract

My B.Tech. Project is to investigate the current techniques for fingerprint classification and recognition. Both targets can be mainly decomposed into image preprocessing, feature extraction and feature match. For each sub-task, some classical and up-to-date methods in literatures are analyzed. Based on the analysis, an integrated solution for fingerprint classification and recognition are developed for demonstration.

For fingerprint classification a new approach called Water Reservoir Technique has been designed and implemented. Water Reservoir Technique is primarily used for image segmentation for character recognition in handwritten text.

For fingerprint recognition a detail survey of existing techniques has been done and based on available information we have implemented a recognition algorithm that is supposed to give better result in terms of speed and accuracy.

Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

1.1 Biometric Authentication System

Question of identity is always there for every system to grant permission or access to only authorized persons. To make the system secure from any kind of fraud we need a system that can identify the authenticity of the person. Traditional means of identification systems such as knowledge-based (e.g., password) or token-based (e.g., key) do not know difference between a genuine user, and an imposer. All these deficiencies of traditional authentication system can be overcome by biometric authentication system that has high level of confidence, such as an error rate of 0.001% [16].

Biometrics is a measurable, physical characteristic or personal behavioral trait used to recognize the identity, or verify the claimed identity, of an enrollee. Biometric recognition is personal recognition based on "who you are or what you do" as opposed to "what you know" (password) or "what you have" (ID card). The advantages of biometrics are - it cannot be transferred, forgotten, lost or copied, it eliminates repudiation claims and it helps in automatic personalization of user interfaces. Biometrics used for identification should have following requirements:

a. Universality (every person should have this characteristics)

b. Uniqueness (no two persons should have the same characteristics)

c. Permanence (characteristic should be invariant)

d. Collectability (characteristic can be measured quantitatively)

Apart from these performances, acceptability and circumvention are some of the criteria that go into consideration.

Based on these criteria face, fingerprint, iris, retina, voice are used as biometrics for authentication.

1.2 What is Fingerprint?

Fingerprints are graphical flow-like ridges (which is a kind of pattern) present on human fingers. Their formation depends on the initial conditions of the embryonic mesoderm from which they develop. Each person has his own fingerprints with the permanent uniqueness. So fingerprints have being used for identification and forensic investigation for a long time.

A fingerprint is composed of many ridges and furrows (Fig: 1.1). Dark portion of the fingerprint image is called ridge and white is called furrow. These ridges and furrows present good similarities in each small local window, like parallelism and average width.

Fingerprints are recognized by special features on ridges which are called Minutia, which are abnormal points on the ridges. Among the variety of minutia types reported in literatures, two are mostly significant and in heavy usage: one is called ridge termination, which is the immediate ending of a ridge; the other is called ridge bifurcation, which is the point on the ridge from which two branches derive (Fig: 1.2).

One of the advantages of fingerprint is that it does not wear out with age. The probability that a fingerprint with 36 minutiae points will share 12 minutiae points with another arbitrarily chosen fingerprint with 36 minutiae points is $6 \cdot 10 \times 10^{-8}$ [23]. Fingerprint identification has 100 years of history and the court accepts it. Even identical twins have different fingerprints.

Figure 1.1: A fingerprint image acquired by an Optical Sensor

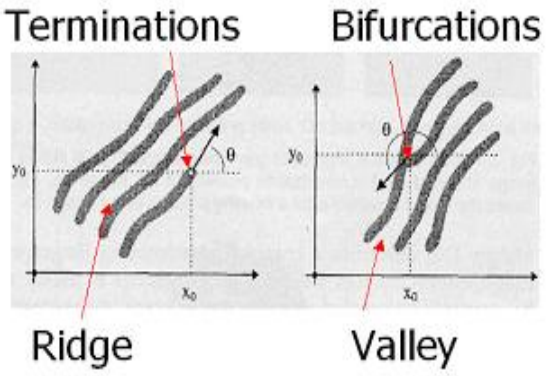

Figure 1.2: Minutia. (Valley is also referred as Furrow, Termination is also called Ending,and Bifurcation is also called Branch)

1.3 Fingerprint Recognition and Classification

The fingerprint recognition problem can be grouped into two sub-domains: one is fingerprint verification and the other is fingerprint identification (Fig: 1.3). Verification is 1:1 match whereas identification is 1:Many match. Watchlist is another kind of identification in which we have one to few match. For identification where there

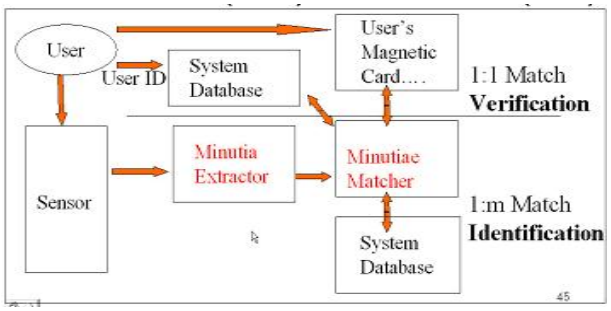

Figure 1.3: Verification vs. Identification

is 1:Many match we need to reduce the domain of search space based on features of fingerprint. To accomplish this task we need fingerprint classification system that divides fingerprint in five broad categories: Left Loop, Right Loop, Whorl, Tainted Arch and Plain Arch. We will discuss this in detail in chapter 7 Fingerprint Classification.

Chapter 2

BACKGROUND

2.1 Biometric Security

As our everyday life is getting more and more computerized automated security systems are getting more and more important. Today most personal banking tasks can be performed over the Internet and soon they can also be performed on mobile devices such as cell phones and PDAs[1]. The key task of an automated security system is to verify that the users are in fact who they claim to be. There are three main methodologies when performing this verification. The security system could ask the user to provide some information known only to the user, it could ask the user to provide something only the user has access to or it could identify some sort of trait that is unique for the user. Of course, some sort of combination of these methodologies is also possible.

The first approach, asking for some personal information such as a password, is the classical approach. It has been used for decades in computer systems, but unfortunately this methodology has a major drawback. The problem is related to how the human memory works and what is demanded of a password for it to be considered secure. For a password to be considered secure, an imposter should not be able to guess the password within a reasonably large number of attempts. This means that it should be randomly chosen and of a certain minimum length. Unfortunately studies have shown that this secure length is longer than seven digits,which makes passwords hard to remember since humans usually only can hold five to nine digits in their short-term memory at any one time [1].

The second approach, asking for some personal belonging such as a smart card, has also been used for a number of years, for example when accessing high security facilities. This methodology also has a major drawback, since what is identified by the security system is not the user but actually the belonging. For example, if an imposter steal an authorized users access card and try to enter a restricted area, there is no way for the security system to know that it is giving access to an imposter and not the user. Of course the method can be combined with a password to get around this problem but then the previously mentioned password problem will be introduced instead.

The third approach, identifying some trait that is unique for the user, is known as biometric security and it is an attempt to get around the previously mentioned problems. A biometrics system is a pattern recognition system that establishes the authenticity of a specific physiological or behavioral characteristic possessed by a user. This thesis will investigate the pros and cons of this approach.

[1] Personal Digital Assistant, mobile handheld device that provide computing capacity and memory.

2.2 System Design for Fingerprint Recognition System

2.2.1 System Level Design

A fingerprint recognition system constitutes of fingerprint acquiring device, minutia extractor and minutia matcher (Fig: 2.1).

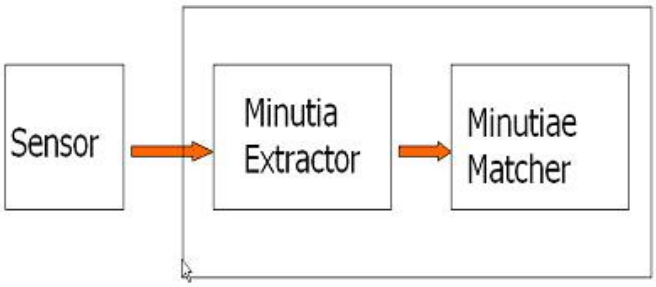

Figure 2.1: Simplified Fingerprint Recognition System

For fingerprint acquisition, optical or semi-conduct sensors are widely used. In our project we have used Optical swipe sensor (model KC-901) from Kinetic Science Inc. They have high efficiency and acceptable accuracy except for some cases that the user's finger is too dirty or dry.

The minutia extractor and minutia matcher modules are explained in detail in the next part for algorithm design and other subsequent sections

2.2.2 Algorithm Level Design

To implement a minutia extractor, a three-stage approach is widely used by researchers. They are preprocessing, minutia extraction and postprocessing stage (Fig: 2.2).

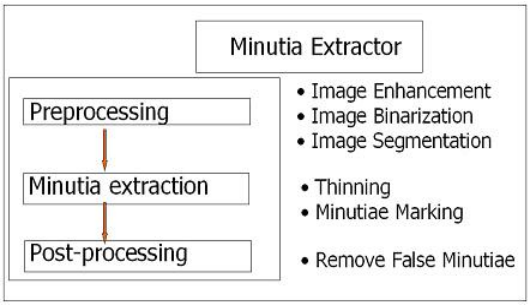

Figure 2.2: Minutia Extractor

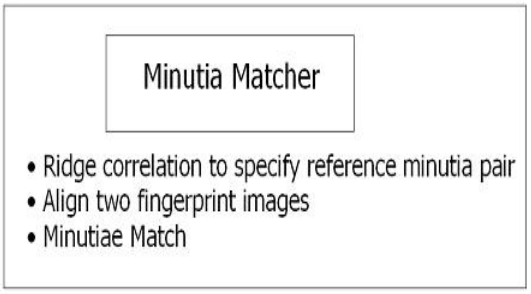

Figure 2.3: Minutia Matcher

For the fingerprint image preprocessing stage, Histogram Equalization and Fourier Transform have been used to do image enhancement. And then the fingerprint image is binarized using the locally adaptive threshold method as well as using Otsu algorithm [17]. The image segmentation task is fulfilled by a three-step approach: block direction estimation, segmentation by direction intensity and Region of Interest extraction by Morphological operations. Most methods used in the preprocessing stage are developed by other researchers but they form a brand new combination in the project through trial and error.

For minutia extraction stage, three thinning algorithms are tested and the Morphological thinning operation is finally bid out with high efficiency and pretty good thinning quality. Guo Hall algorithm is used for morphological thinning [29]. The minutia marking is a simple task as most literatures reported.

For the postprocessing stage, a more rigorous algorithm is developed to remove false minutia.

The minutia matcher chooses any two minutia as a reference minutia pair and then match their associated ridges first. If the ridges match well, two fingerprint images are aligned and matching is conducted for all remaining minutia (Fig: 2.3).

Chapter 3

AUTOMATIC FINGERPRINT CLASSIFICATION & IDENTIFICATION SYSTEM

3.1 Fingerprint Image Preprocessing

Fingerprint Image Preprocessing is to make the image clearer for easy further operations. Since the fingerprint images acquired from sensors or other medias are not assured with perfect quality, those enhancement methods, for increasing the contrast between ridges and furrows and for connecting the false broken points of ridges due to insufficient amount of ink, are very useful for keep a higher accuracy to fingerprint recognition.

The quality of the fingerprint depends on the scanner used for the fingerprint image acquisition. Before extraction of minutia features we need to preprocess the fingerprint image. The basic steps that are used for fingerprint image preprocessing is shown in figure (Fig: 3.1).

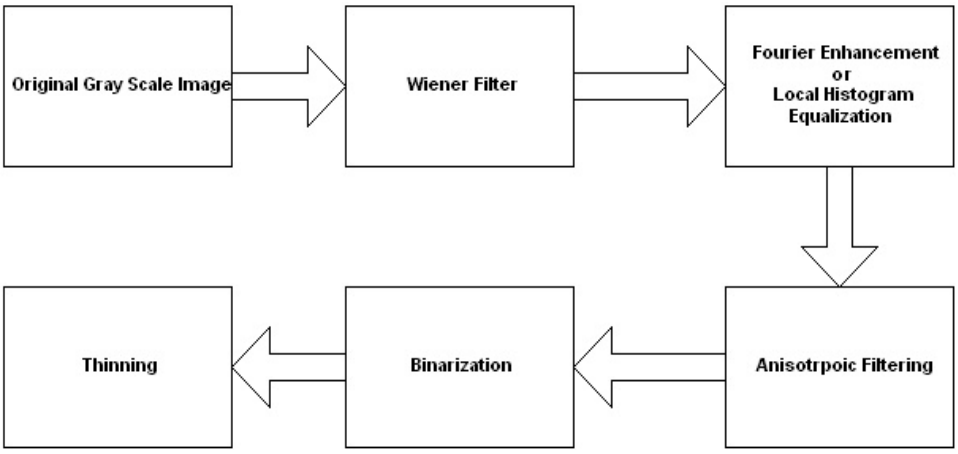

Figure 3.1: Steps of Fingerprint Enhancement

3.1.1 Fingerprint Image Noise Enhancement

Noise is inherent property of the signal. In fingerprint acquisition noise is imparted due to insufficiency of the fingerprint scanner. The nature of noise incorporated because of scanner is static in nature. To remove such noise wiener filter has been used in noise enhancement technique.

The Wiener filter is optimal stationary linear filter for images degraded by additive noise and blurring. We propose to use a pixel-wise adaptive Wiener method for noise reduction. The filter is based on local statistics estimated from a local neighborhood of size 3x3 of each pixel, and is given by:

$$b(n_1, n_2) = \mu + \frac{\sigma^2 - \nu^2}{\sigma^2}(a(n_1, n_2) - \mu) \tag{3.1}$$

where

$$\mu = \frac{1}{NM} \sum_{n_1, n_2 \in \eta} a(n_1, n_2) \tag{3.2}$$

$$\sigma^2 = \frac{1}{NM} \sum_{n_1, n_2 \in \eta} a^2(n_1, n_2) - \mu^2 \tag{3.3}$$

where η is the N-by-M local neighborhood of each pixel in the image A. If the noise variance is not given, then in the implementation the average of all the local estimated variances is used.

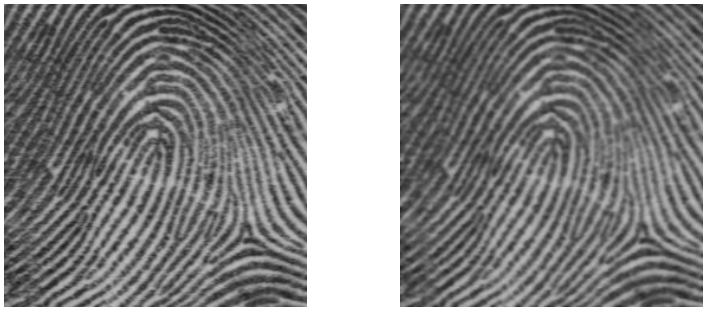

Figure 3.2: Effect of Winer Filter (Left: Raw Image Right: Wiener Filtered Image)

An image which is corrupted with pepper and salt noise can be treated with median filter. Instead of mean filter median filter is proposed because median filter preserves the edge information, it suppresses the impulse noise and smoothes the signal (Fig: 3.3) .

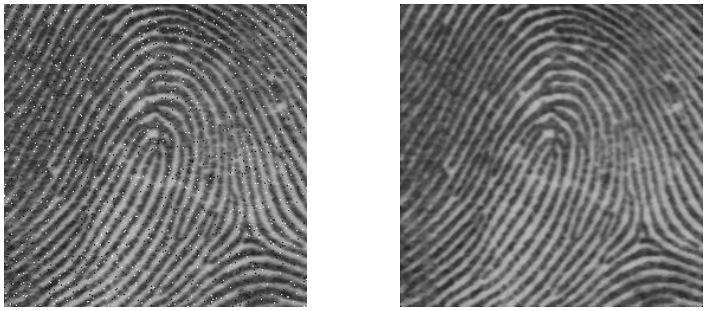

Figure 3.3: Effect of Median Filter (Left: Salt and Pepper Noise Right: Median Filtered Image)

Postliminary noise such as specks and holes are removed either through morphological operation or through Line Adjacency Graph [8].

3.1.2 Fingerprint Image Enhancement

Fingerprint Image enhancement is necessary for further easy operations. Fingerprint images acquired from sensors or other medias are not assured with perfect quality, hence there is need for image enhancement methods. Fingerprint image enhancement techniques help in increasing the contrast between ridges and furrows and for connecting the false broken points of ridges due to insufficient amount of ink. These techniques are inevitable for a higher accuracy of fingerprint recognition.

Three Methods are adopted in implemented fingerprint recognition system:

a. Histogram Equalization

b. Fourier Transform

c. Anisotropic Filtering

Histogram Equalization

Histogram equalization defines a mapping of gray levels p into gray levels q such that the distribution of gray level q is uniform. This mapping stretches contrast (expands the range of gray levels) for gray levels near the histogram maxima. Since contrast is expanded for most of the image pixels, the transformation improves the delectability of many image features [4].

Consider an image pixel value $u = 0$ to be a random variable with a continuous probability density function $p_u(u)$ and cumulative probability distribution $F_u(u) = P[u \leq u]$. Then the random variable

$$V = F_u(u) = \int_0^u p_u(u)du \tag{3.4}$$

will be uniformly distributed over (0,1)[10].

Let us assume the input image u has L gray levels $x_i, i = 0, 1, ..., L-1$ with probabilities $p_u(x_i)$ where,

$$p_u(x_i) = \frac{h(x_i)}{\sum_0^{L-1} h(x_i)}, i = 0, 1, ..., L-1 \tag{3.5}$$

(where $h(x_i)$ is the number of pixel with gray level value x_i.)

The output v, also assumes to have L levels

$$v = \sum_0^u p_u(x_i) \tag{3.6}$$

where

$$v = Int[\frac{(v - v_{min}) \times (L-1)}{(1 - v_{min})} + 0.5] \tag{3.7}$$

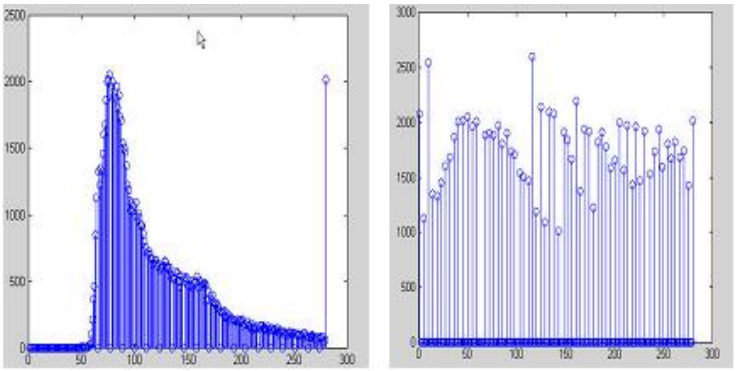

Figure 3.4: Histogram Equalization (Left: Original Histogram Right:Histogram after Histogram Equalization

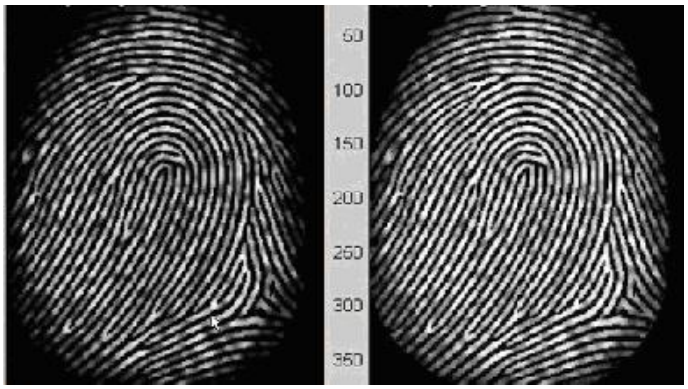

Figure 3.5: Histogram Enhancement Original Image (Left). Enhanced image (Right)

Fourier Transform

The image is divided into small processing blocks (32 by 32 pixels) and then Fourier transform is performed according to:

$$F(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \times \exp(-j2\pi \times (\frac{ux}{M} + \frac{vy}{N})) \tag{3.8}$$

for $u = 0, 1, 2, ..., 31$ and $v = 0, 1, 2, ..., 31$.

In order to enhance a specific block by its dominant frequencies, we multiply the FFT of the block by its magnitude a set of times. Where the magnitude of the original $FFT = abs(F(u,v)) = |F(u,v)|$.

Then get the enhanced block according to

$$g(x,y) = F^{-1}(F(u,v) \times |F(u,v)|^K) \tag{3.9}$$

where $F^{-1}(F(u,v))$ is done by

$$f(x,y) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} F(u,v) \times \exp(-j2\pi \times (\frac{ux}{M} + \frac{vy}{N})) \tag{3.10}$$

for $x = 0, 1, 2, ..., 31$ and $y = 0, 1, 2, ..., 31$.

The k in formula (3.9) is an experimentally determined constant, which we choose k=0.45 to calculate. While having a higher "k" improves the appearance of the ridges, filling up small holes in ridges, having too high a "k" can result in false joining of ridges. Thus a termination might become a bifurcation. Figure (3.6) presents the image after FFT enhancement.

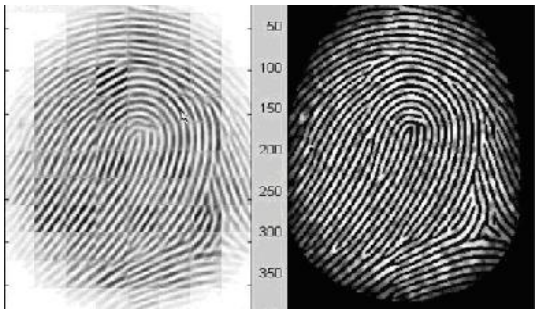

Figure 3.6: Fingerprint enhancement by FFT ,Enhanced image (left), Original image (right)

Anisotropic Filtering

Ridges are sometimes degraded due to limitations of sensors and subsequent preprocessing stages. In order to overcome this we use anisotropic filtering.

Anisotropic filtering is a feature of some video cards that sharpens the details of the fading-away part of an object that recedes into the distance. Anisotropic means "non-uniform shape" and it is because this filtering technique works on non-uniform, or uneven, shaped areas that it is more powerful, and takes much more processing power, than point sampling, bi-linear or tri-linear filtering.

Anisotropic diffusion [20] has been developed to smooth an image, thus removing high frequency noise, while preserving the boundaries of structures of interest.

3.1.3 Fingerprint Image Segmentation

The fingerprint image acquired from sensor is not all together of interest. There is Region of Interest (ROI) which is to be recognized and the image area without effective ridges and furrows needs to be discarded because it just holds the background information. Then the bound of the ROI is sketched out since the minutia in the ROI are confusing with those spurious minutia that are generated when the ridges are out of the sensor.

To extract the ROI, a two-step method is used.

a. Block Direction Estimation

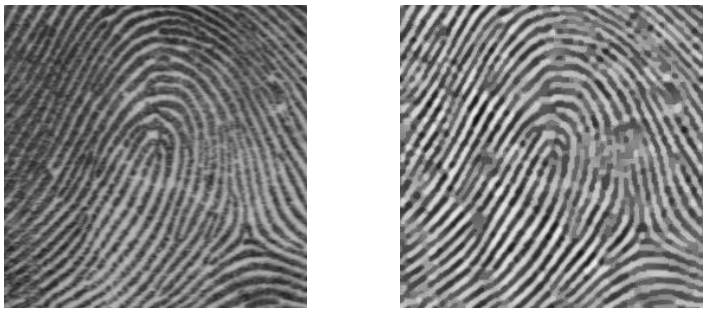

Figure 3.7: Fingerprint enhancement by Anisotropic Filtering ,Enhanced image (right), Original image (left)

b. Morphological operation to extract ROI.

Block Direction Estimation
Estimate the block direction for each block of the fingerprint image with $W \times W$ in size (W is 16 pixels by default). The algorithm is:

a. Calculate the gradient values along x-direction (g_x) and y-direction (g_y) for each pixel of the block. Two Gaussian filters are used to fulfill the task.

b. For each block, use Following formula to get the Least Square approximation of the block direction.

$$tg2\beta = 2\frac{\sum\sum g_x \times g_y}{\sum\sum g_x^2 - g_y^2} \tag{3.11}$$

for all the pixels in each block.

The formula is easy to understand by regarding gradient values along x-direction and y-direction as cosine value and sine value. So the tangent value of the block direction is estimated nearly the same as the way illustrated by the following formula.

$$tg2\theta = 2\frac{\sin\theta\cos\theta}{\sin^2\theta - \cos^2\theta} \tag{3.12}$$

After finished with the estimation of each block direction, those blocks without significant information on ridges and furrows are discarded based on the following formulas:

$$E = \frac{2\sum\sum(g_x \times g_y) + \sum\sum(g_x^2 - g_y^2)}{W \times W \times \sum\sum(g_x^2 + g_y^2)} \tag{3.13}$$

For each block, if its certainty level E is below a threshold, then the block is regarded as a background block.

The direction map is shown in the following figure 3.8

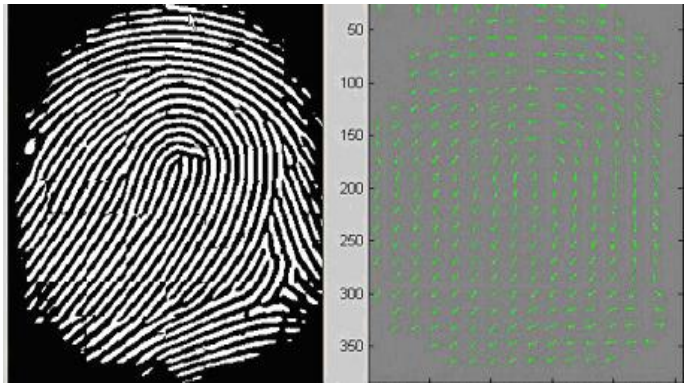

Figure 3.8: Directional map. Binarized fingerprint (left), Direction map (right)

Morphological operations to extract ROI
Two Morphological operations called 'OPEN' and 'CLOSE' are adopted. The 'OPEN' operation can expand images and remove peaks introduced by background noise [Fig: 3.9]. The 'CLOSE' operation can shrink images and eliminate small cavities [Fig: 3.9].

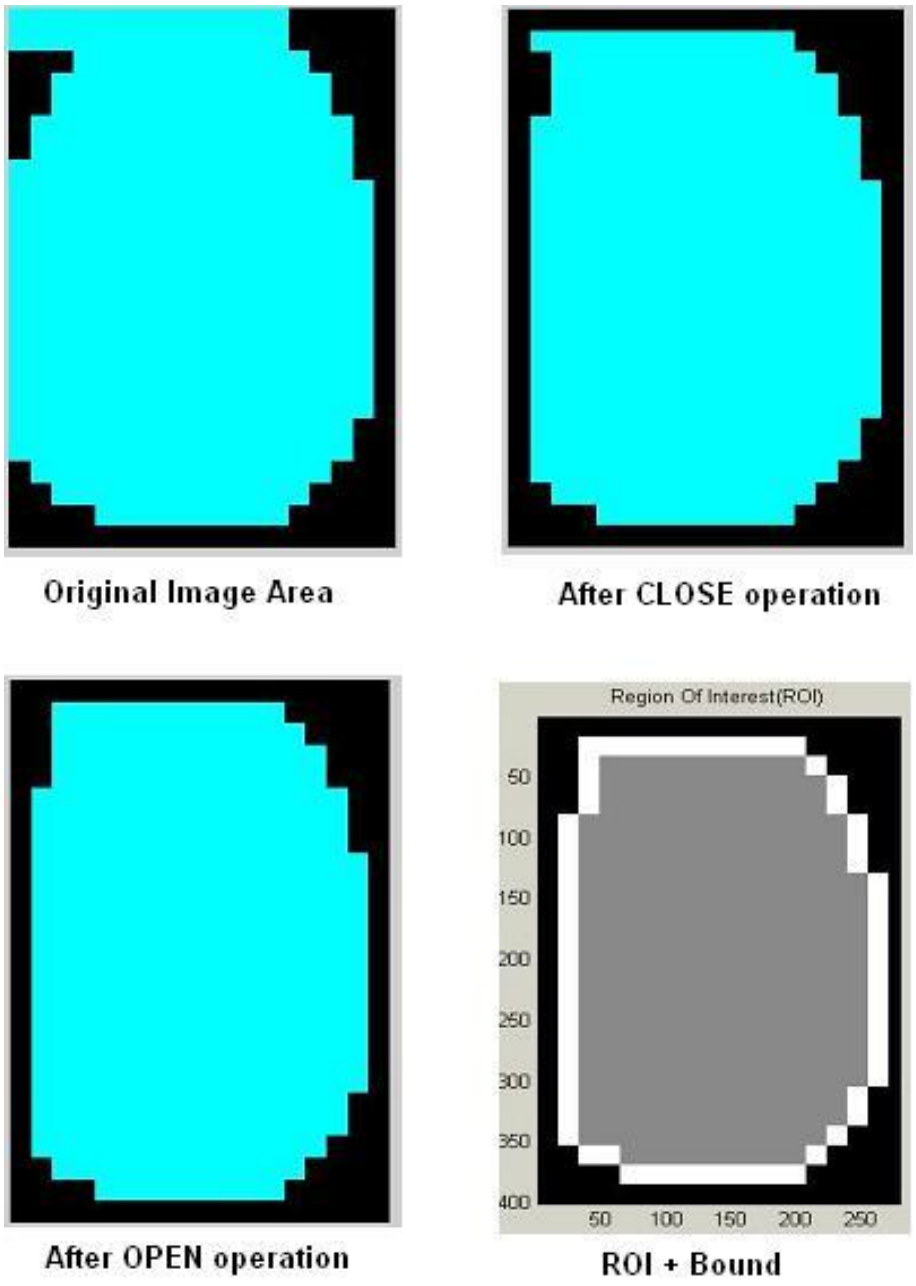

Figure 3.9: show the interest fingerprint image area and its bound. The bound is the subtraction of the closed area from the opened area. Then the algorithm throws away those leftmost, rightmost, uppermost and bottommost blocks out of the bound so as to get the tightly bounded region just containing the bound and inner area.

3.1.4 Fingerprint Image Binarization

Binarization is the process of converting the enhanced gray scale image into two-tone image having 0s and 1s only. In case of binarized image 1 represents foreground part of image and 0 represents background of image. Foreground constitutes the ridge of the image which is of interest for minutia extraction.

Two methods have been implemented to binarize the image:

a. Morphological Operation

b. Otsu Algorithm

Morphological Operation
Morphological operation to binarize the enhanced image constitutes following processes:

Ridge Segment

Ridge Segment identifies ridge regions of a fingerprint image and returns a mask identifying this region. It also normalizes the intensity values of the image so that the ridge regions have zero mean, unit standard deviation.

This function breaks the image up into blocks of size blksze x blksze and evaluates the standard deviation in each region. If the standard deviation is above the threshold it is deemed part of the fingerprint. Note that the image is normalized to have zero mean, unit standard deviation prior to performing this process so that the threshold you specify is relative to a unit standard deviation.

In our implementation Block size is 16 and threshold value is 0.1.

Ridge Orient

Ridge Orient - Estimates the local orientation of ridges in a fingerprint. In this we find out gradient of the ridges and use gaussian filter to smooth the ridges and find out the moments to get the orientation of each points on ridge [25].

Ridge Frequency and Ridge Filter

Ridge frequency estimates the fingerprint ridge frequency across a fingerprint image. Ridge Filter - enhances fingerprint image via oriented filters [7].

Binarization

In binarization the filtered output of the ridge filter and ridge frequency which has got value greater than 1 for ridge and lower than one for valley is simply converted to two-toned image by simple comparison.

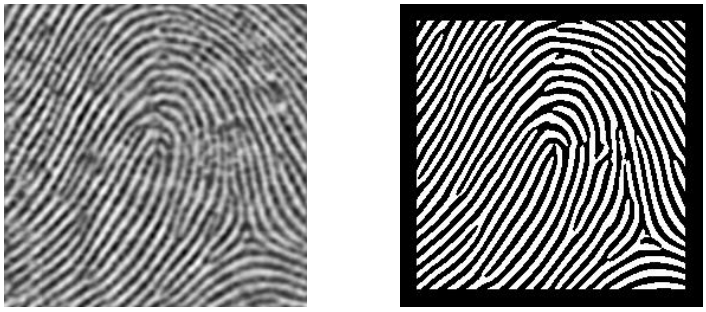

Figure 3.10: Morphologically Binarized Image. Enhanced fingerprint (left), Binarized fingerprint (right)

Otsu Algorithm
Otsu is the nonparametric and unsupervised method of automatic threshold selection for image segmentation. An optimal threshold is selected by the discriminant criterion, namely, so as to maximize the separability of the resultant classes in gray levels [18].

Between class variance

$$\sigma_B^2 = \omega_0\omega_1(\mu_1 - \mu_2)^2 \tag{3.14}$$

Total Variance

$$\sigma_T^2 = \sum_{1}^{L}(i - \mu_T)^2 p_i \tag{3.15}$$

Discriminant Criterion

$$\eta = \frac{\sigma_B^2}{\sigma_T^2} \tag{3.16}$$

where

$$p_i = n_i/N \tag{3.17}$$

is a probability distribution.

$$\omega_0 = \sum_{i=1}^{k} p_i \tag{3.18}$$

is probabilities of class occurrence

$$\omega_1 = 1 - \omega_0 \tag{3.19}$$

is probabilities of the class mean levels.

$$\mu_0 = \frac{\sum_{i=1}^{k} ip_i}{\omega_i} \tag{3.20}$$

is the zeroth-order cumulative moments of the the histogram up to kth level.

$$\mu_1 = \frac{\sum_{i=k+1}^{L} ip_i}{\omega_i} \tag{3.21}$$

is the first-order cumulative moments of the the histogram up to Lth level.

$$\mu_T = \sum_{i=1}^{L} ip_i \tag{3.22}$$

Our job now boils down to maximize the discriminant criterion η for the suitable value of grey level k.

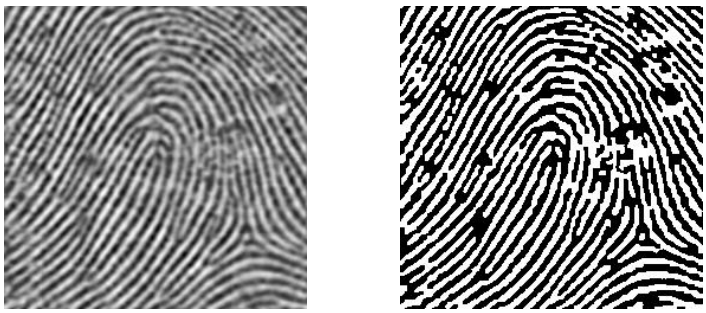

Figure 3.11: Binarized Image by Otsu Algorithm. Enhanced fingerprint (left), Binarized fingerprint (right)

3.1.5 Fingerprint Image Ridge Thinning

Thinning is the process of changing the foreground image i.e. ridge in case of fingerprint into one pixel wide so that further processing of the fingerprint for minutia extraction becomes easy.

There are various morphological operations for thinning like Guo Hall, Zhang and Wang [27], Hilditch [6], Zhang and Suen [30] and Rosenfeld [24]. Since Guo-Hall algorithm gives best this algorithm is described here.

Guo Hall Thinning Algorithm

We refer p_2, p_4, p_6, and p_8 as p's side neighbors and p_1, p_3, p_5 and p_7 as p's diagonal neighbors.

C(p) is defined as the numbers of distinct eight-connected components of ones in p's eight-neighborhood.

B(p) is the number of ones in p's eight-neighborhood.

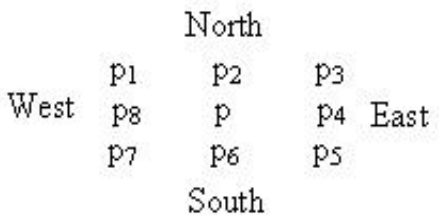

Figure 3.12: Neighborhood Definition for Pixel p

N(p) is useful for endpoint detection but it also helps in achieving thinner results:

$$N(p) = min[N_1(p), N_2(p)] \tag{3.23}$$

where

$$N_1(p) = (p_1 \vee p_2) + (p_3 \vee p_4) + (p_5 \vee p_6) + (p_7 \vee p_8) \tag{3.24}$$

And

$$N_2(p) = (p_2 \vee p_3) + (p_4 \vee p_5) + (p_6 \vee p_7) + (p_8 \vee p_1) \tag{3.25}$$

$N_1(p)$ and $N_2(p)$ each break the ordered set of p's neighborhood pixels into four pairs of adjoining pixels and count the number of pairs which contain 1 or 2 ones.

Algorithm:

A one of S, p, is deleted (one changed to zero) iff all of the following conditions are satisfied:

a. $C(p) = 1$;

b. $2 \leq N(p \leq 3$; and

c. Apply one of the following

(1) $(p_2 \vee p_3 \vee \bar{p_5}) \vee p_4 = 0$ in odd iteration, or

(2) $(p_6 \vee p_7 \vee \bar{p_1}) \vee p_8 = 0$ in even iteration

Thinning stops when no further deletions occur [5].

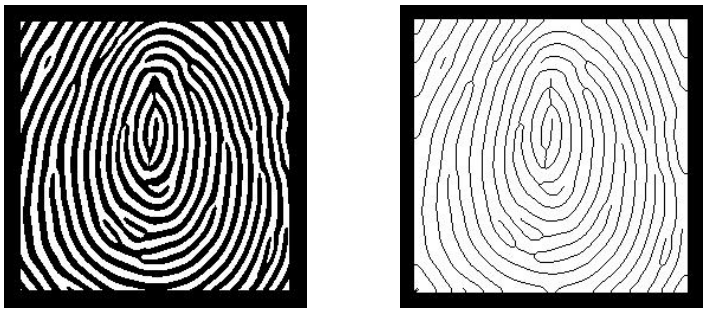

Figure 3.13: Guo Hall Thinning Algorithm. Binary Image (left) Thinned Image(right)

3.2 Fingerprint Classification

Fingerprint can be classified in different classes which help in reducing the search domain space for identification purpose. Fingerprints were first classified by Henry into mutually exclusive classes. The primary application of fingerprint classification is indexing [15].

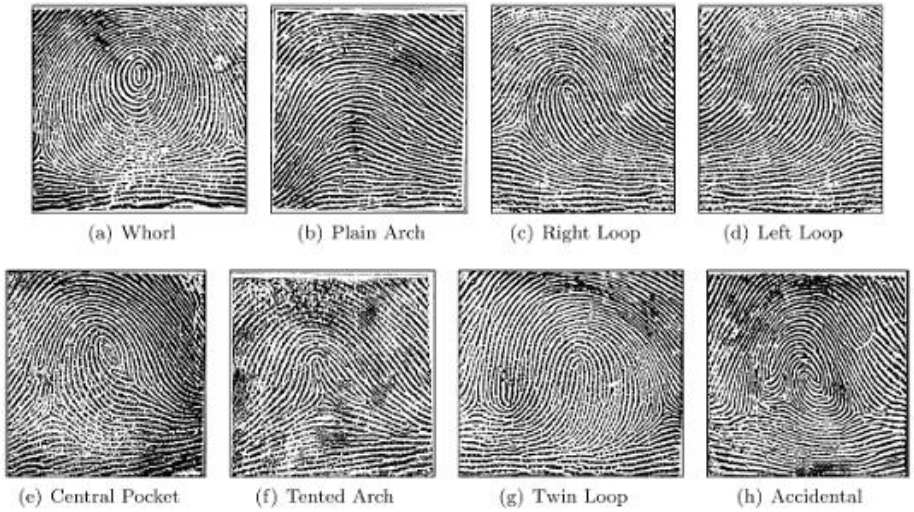

Figure 3.14: Henry' Fingerprint Classes

The distribution of fingerprint classes in nature is not uniform. Central pockets, twin loop and accidental are very rare so they are often ignored for classification purposes. The probabilities of other classes are approximately 0.037 (arch), 0.338 (left loop), 0.317 (right loop), 0.029 (tented arch) and 0.279 (whorls) [28].

3.2.1 Identification Vs Verification

The fingerprint recognition problem can be grouped into two sub-domains: one is fingerprint verification and the other is fingerprint identification. Verification is easier in the sense the search domain is very small whereas identification is time consuming because the search domain is very big.

Type of Process	Type of Match
Verification	1:1
Identification	1:Many
Watchlist	1:Few

Table I: Verification Vs Indentification

Thus when we are required to identify a user we are supposed to match an individual's fingerprint with that of the fingerprints stored in our database. Since the fingerprint images stored in database is of huge size we need to minimize the search domain in order to accelerate the matching process. To achieve this purpose we classify the fingerprint image into five main classes.

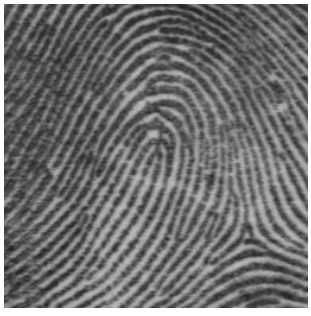

Figure 3.15: Left Loop

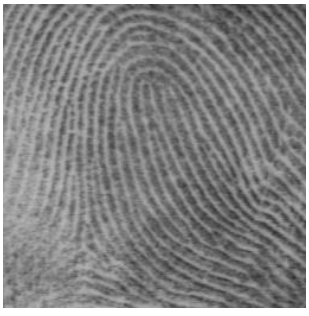

Figure 3.16: Right Loop

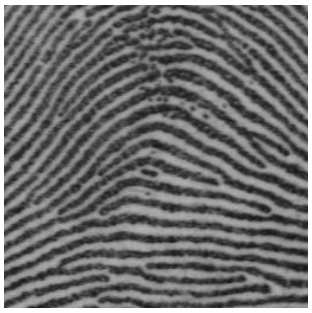

Figure 3.17: Plain Arch

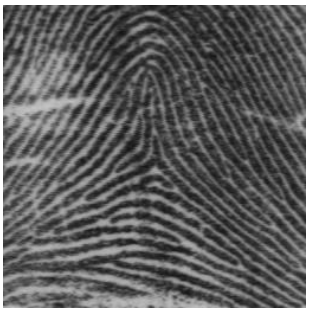

Figure 3.18: Tainted Arch

Figure 3.19: Whorl

3.2.2 Traditional Approach (Based on Global Features)

In traditional approach of fingerprint classification we use global features of fingerprint to classify the fingerprint. Global features are also known as singularity points of fingerprint. A singularity is local region of a fingerprint where the ridge pattern has special properties making it visually prominent [14]. There are two types of fingerprint singularities: cores and deltas [Fig:3.20].

Figure 3.20: Singularity Points

A core is the turning point of an inner-most ridge and a delta is a place where two ridges running side-by-side diverge. Henry originally introduced the concept of fingerprint singularities as an aid for fingerprint classification [3].

The most common singularity extraction is Poincare index. In the context of fingerprint image, the Poincare index is defined as the rotation of vectors along a curve in the orientation field. All the points in fingerprint can be classified as either core points, delta points or normal points depending on their Poincare index.

Once the Core and Delta points are determined we classify the fingerprint based on Kawagoe and Tojo's coarse classification by singularity count [12][see table II]. However one of the problem with this algorithm is that if the fingerprint is translated to such an extent that its delta point is out of ROI then we can not determine the class of the fingerprint.

Based on just Core and Delta points and location of the delta point fingerprint can be classified different classes [21] which is shown in table III.

Whorl	Core	Delta	Type
1	0	*	Whorl
0	1	*	Loop,Pocketed Loop,or Tented Arch
0	2	*	Twin Loop or Whorl
0	0	0	Arch

Table II: Kawagoe and Tojo's coarse classification by singularity count. A whorl point contains two close core points and *represents any number

Pattern Class	Core	Delta
Arch	0	0
Tented Arch	1	1(middle)
Left Loop	1	1(Right)
Right Loop	1	1(Left)
Whorl	2	2

Table III: Fingerprint pattern classes and the corresponding number of singular points

3.2.3 New Approach (Water Reservoir Technique)

Water reservoir technique is a method for segmentation of image. Segmented image then filled with water from different directions and then based on pattern of the water flow we select some features that can classify the image in particular category.

The water reservoir principle is as follows. If water is poured from one side of a component, the cavity regions of the component where water will be stored are considered as reservoirs [26]. By top (bottom) reservoirs we mean the reservoirs obtained when water is poured from top (bottom) of the component. (A bottom reservoir of a component is visualized as top reservoir when water will be poured from top after rotating the component by 180). Similarly if water is poured from left (right) side of the component, the cavity regions of the components where water will be stored are considered as left (right) reservoirs. For an illustration see Fig 3.21. Here top, bottom, left and right reservoirs of some Urdu characters are shown. Water flow direction from a full reservoir is also shown in this figure [19].

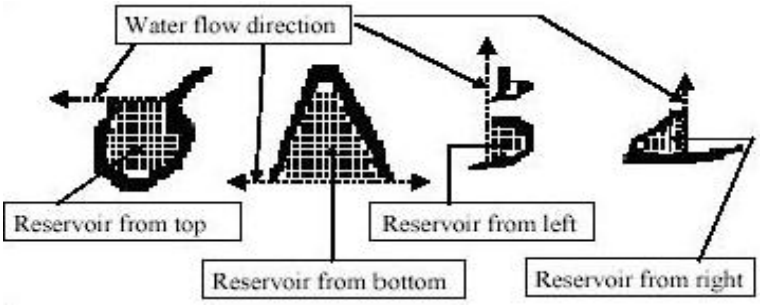

Figure 3.21: Different reservoirs and their water flow directions are shown in four characters. Water flow directions are shown by dotted arrow

Water reservoir technique for fingerprint classification constitutes following steps:

a. First of all core point of the raw image is detected. Around the core point 3*3 window is drawn (of given window size W) if possible depending on the window size. If drawing of 3*3 window of each window size W is not possible then stop at this stage and discard the image.

b. Each window of size W is then treated to connect the unconnected region using *Least Distance Joining Algorithm* in order to control the overflow of the water.

c. Now the water is filled from four directions one by one: from left, from right, from upper, from lower.

d. For each direction of water fill we decide certain features which is based on water fill pattern for different classes of fingerprint images. Based on these features we give score to each type of fingerprint and select the one which has got maximum score.

Figure 3.22: Fingerprint and its core point detection

Figure 3.23: 3*3 window size selection and Water Flow from Left Side after Joining

e. The maximum consensus is chosen for the image. If there is consensus is less than two then the classification is undecided.

Figure 3.24 show the flow diagram for the given algorithm.

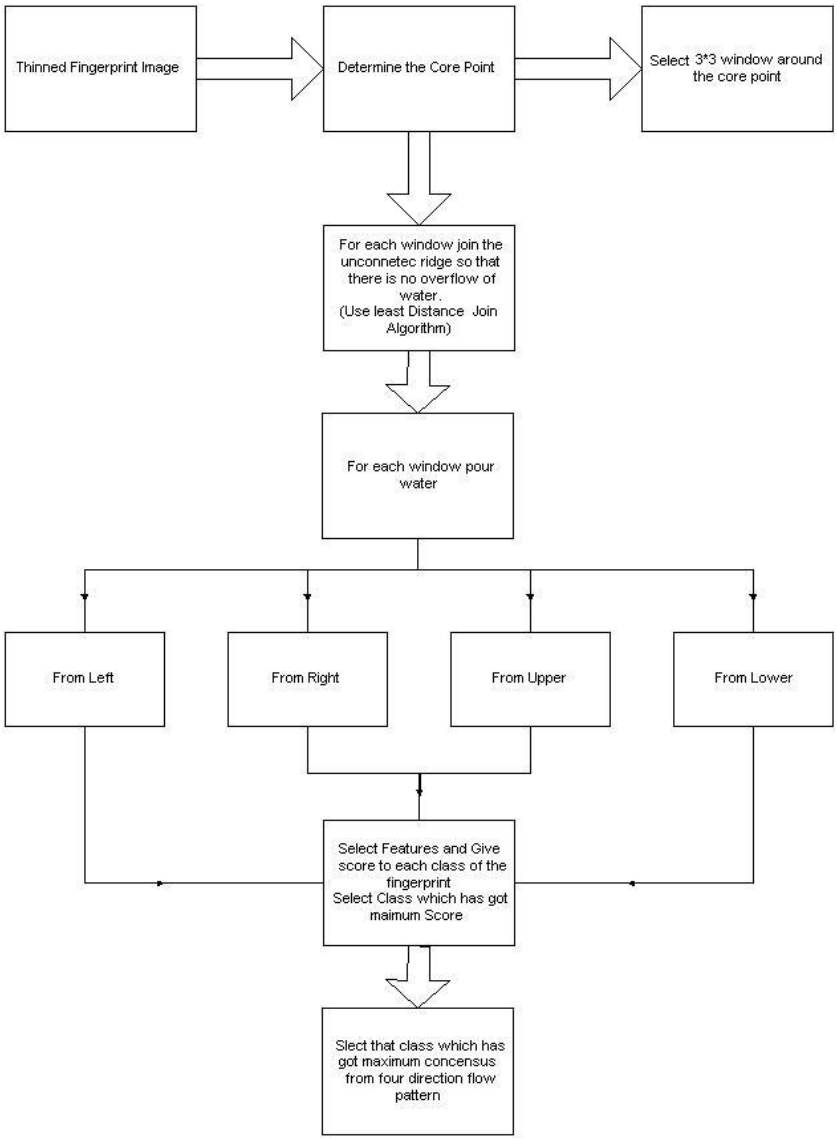

Figure 3.24: Flow diagram for fingerprint classification based on Water Reservoir Technique

Some of the features that are used as pattern for the classification are as follow:

a. Number of cells filled more than 50% [$N_{50\%}$].

b. Position of least filled region. [P_{less}]

c. Depth per column. [D_{col}]

d. Water-flow attributes.

Note: Attribute 2 is considered when we have segmented image into 5*5 window size in which case we don't bother about the position of the core point.

3.3 Fingerprint Recognition

3.3.1 Minutia Extraction

Fingerprint image preprocessing that we discussed in chapter 3 constitutes processing of gray scale raw fingerprint image in order to enhance its quality by various techniques that helps in clear generation of ridges. After enhancement of fingerprint image, the two-tone image is thinned so that extraction of minutia features becomes easy. Though there are some techniques described in some literature that claims minutia features can be derived directly from gray scale fingerprint image [11] [13]. In the implemented algorithm thinned fingerprint image is used to extract the minutia features.

Minutia Marking

Minutia marking constitutes extracting local and global features of the fingerprint. Local ridge features constitute ridge ending and ridge bifurcation. Global ridge features constitute singular points that primarily constitute core and delta point. We will talk about global features in fingerprint classification in chapter [4]. Local features are used as features for matching.

In general, for each 3×3 window, if the central pixel is 1 and has exactly 3 one-value neighbors, then the central pixel is a ridge branch [Figure 4.1.1]. If the central pixel is 1 and has only 1 one-value neighbor, then the central pixel is a ridge ending [3.25].

The algorithm applied to detect the minutia is mask based [9]. Minutia detection is a trivial task when an ideal thinned ridge map is obtained. Without loss of generality, we assume that if a pixel is on thinned ridge (eight-connected), then it has a value 1, and 0 otherwise. Let (x, y) denote a pixel on a thinned ridge (it will have value 1), and $N_0, N_1, , N_7$ denote its eight neighbours. Then,

a. A pixel (x, y) is a ridge ending if,

$$\sum_{i=0}^{8} = 1 \tag{3.26}$$

b. A pixel (x, y) is ridge bifurcation if,

$$\sum_{i=0}^{8} > 2 \tag{3.27}$$

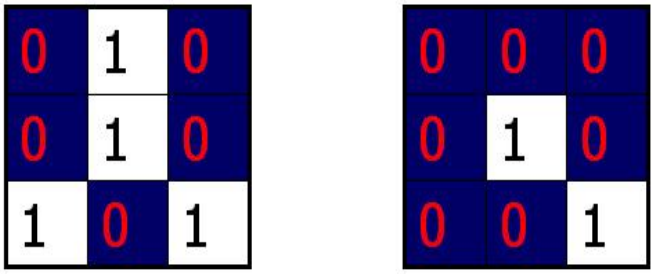

Figure 3.25: Bifurcation(left) Termination(right)

3.3.2 Minutia Post-processing

The presence of undesired spikes and breaks present in a thinned ridge map may lead to many spurious minutia being detected. Therefore before the minutia detection, smoothing procedure is applied to remove spikes and to join broken ridges. After false minutia have been removed the minutia features are stored in database.

False Minutia Removal

Because of limitations of enhanced techniques we may end up in following types of false minutia detection which needs to be removed.

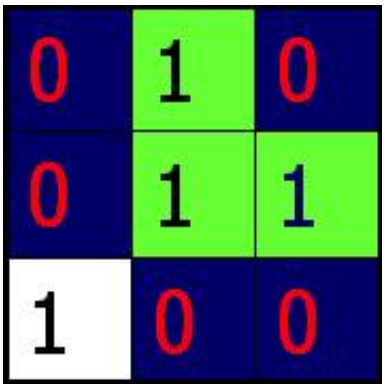

Figure 3.26: A special case that depicts a genuine branch is triple counted. Suppose both the uppermost pixel with value 1 and the rightmost pixel with value 1 have another neighbor outside the 3x3 window, so the two pixels will be marked as branches too. But actually only one branch is located in the small region. So a check routine requiring that none of the neighbors of a branch are branches is added

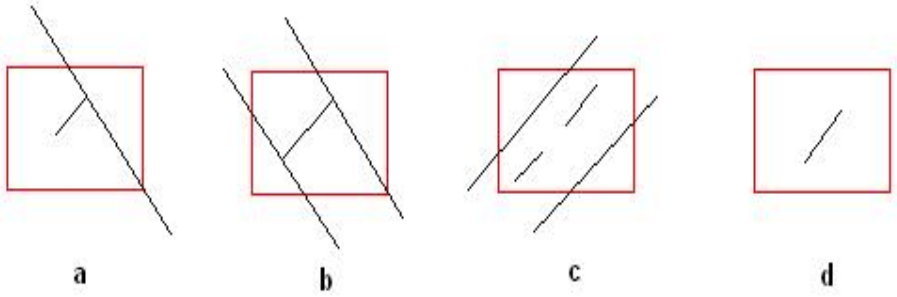

Figure 3.27: Four types of false minutia

In 3.28 first figure 'a' refers to spike piercing into a valley, figure 'b' is 'H' connection which is a spike that connects two ridges falsely, figure 'c' is ridge broken points and figure 'd' is isolated short ridge. These all give rise to false minutia points.

Our ridge smoothing algorithm uses following heuristics:

a. If the distance between one bifurcation and one termination is less than D and the two minutia are in the same ridge (case a). Remove both of them. Where D is the average inter-ridge width representing the average distance between two parallel neighboring ridges.

b. If the distance between two bifurcations is less than D and they are in the same ridge, remove the two bifurcations. (cases b).

c. If two terminations are within a distance D and their directions are coincident with a small angle variation. And they suffice the condition that no any other termination is located between the two terminations. Then the two terminations are regarded as false minutia derived from a broken ridge and are removed. (case c).

d. If two terminations are located in a short ridge with length less than D, remove the two terminations (d).

e. If several minutia form a cluster in a small region, then remove all of them except for the one nearest to the cluster center.

Minutia Features Store

Once the false minutia has been removed we are ready to store the minutia features. Following are the features that are stored as minutia features.

a. Total minutia points.

b. Minutia positions (x-coordinate and y-coordinate)

c. Minutia angle/direction (orientation which is defined as the local ridge orientation of the associated ridge).

d. The associated ridge (Each ridge is labeled and each minutia point is associated with particular labeled ridge).

e. Average inner ridge width. (The average inter-ridge width refers to the average distance between two neighboring ridges.)

In our implementation these information are stored in a file. Since access to the file is faster than the database, storing minutia in file system would help in reducing the time taken for matching process.

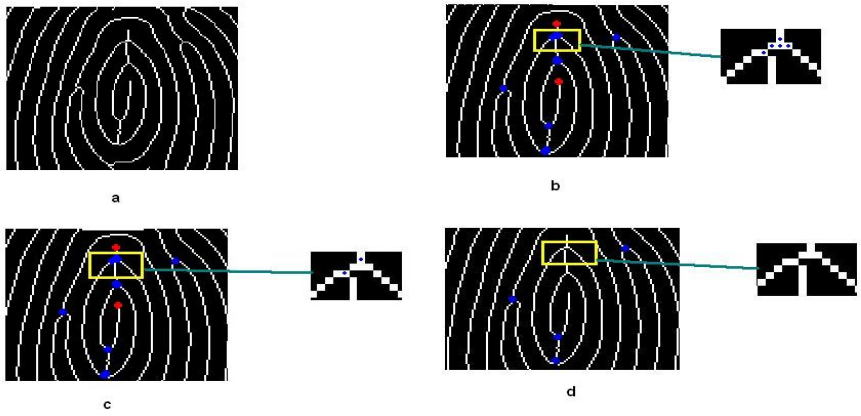

Figure 3.28: (a):Thinned Fingerprint, (b)Minutia Marking, (c)False Minutia Removal-iteration 1, (d)False Minutia Removal-iteration 2

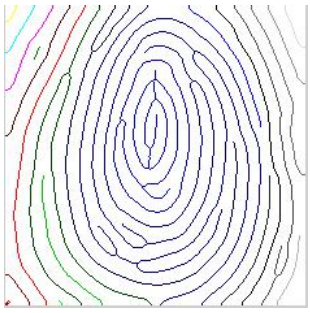

Figure 3.29: Minutia Labelling

FileName Whorl Thin.bmp

AverageRidgeWidth 1.222080e+001

Termination::

Number	X	Y	Orientation
1	108	37	1.730086e+000
2	197	94	4.634102e-001

Bifurcation::

Number	X	Y	Orientation
1	68	36	2.074638e+000
2	146	36	1.473258e+000

Figure 3.30: Minutia Information

3.3.3 Minutiae Matching

Given two set of minutia of two fingerprint images, the minutia match algorithm determines whether the two minutia sets are from the same finger or not. An automatic fingerprint verification/identification is achieved with point pattern matching (minutiae matching) instead of a pixel-wise matching or a ridge pattern matching of fingerprint images. A general point matching problem is essentially intractable, features associated with each point and their spatial properties such as the relative distances between points are often used in these algorithms to reduce the exponential number of search paths.

Our alignment based minutiae matching algorithm is decomposed into two steps:

a. Alignment - where transformations such as translation and rotation between an input and a template in the database are estimated and the input minutiae are aligned with the template minutiae according to the estimated parameters; and

b. Matching - where both input and template minutiae are matched. Matching score is calculated as following:

$$MatchingScore = \frac{Q \cap D}{Q + D - (Q \cap D)} \tag{3.28}$$

where

Q = Number of Minutiae in input fingerprint

D = Number of Minutiae in fingerprint template

$Q \cap D = \sharp$ of minutiae that occur in both images simultaneously.

Fingerprint Alignment

In this step transformations such as translation and rotation between input and template image as estimated and input minutiae are aligned with the template minutiae according to the estimated parameters. We do not expect any distortion due to scaling considering the nature of images we will get from the scanner [2].

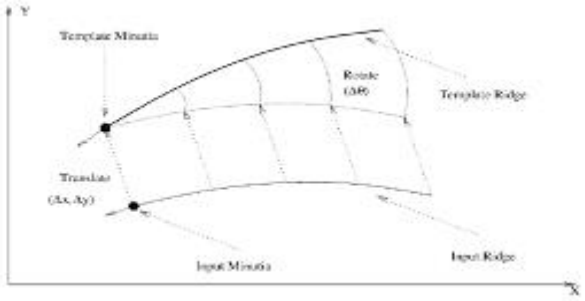

Figure 3.31: Alignment of the input and the template Image for matching

Translation
In order to determine translation we need to determine the reference point, which is defined as the most curvature convex point on a fingerprint image. This point is also known as core point. This point is approximately the centre point of the fingerprint images and can be found from ridge directional field, which is determined in the pre-processing step. We use Poincare method for finding out core point.

After determination of the core point in the input fingerprint, it is compared with corresponding core point in the template, and input fingerprint is shifted by the difference between the two core points.

Rotation
There exists a possibility of input image from the fingerprint scanner being a rotated version of the image in the template. This will lead to wrong coordinates of minutiae in the input image and matching will fail. So we need to take care of the rotation. We take three minutiae nearest to the core point of the input image and template image. We calculate the angles between the minutiae with respect to the core point (See figure 2) for both input and template minutiae. We use two angles thus calculated to estimate what is the degree of rotation in the image.

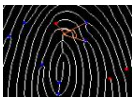

Figure 3.32: Alignment of the input and the template Image for matching

Fingerprint Matching

Once the input fingerprint has been aligned with the template we try to match the minutiae in input fingerprint with that in the image. For every minutia in the input fingerprint, correlation matching is performed. Let Q be the number of minutiae in the query image and D be the number of minutiae in the template, the matching score (decision level) is defined as

$$MatchingScore = \frac{Q \cap D}{Q + D - (Q \cap D)} \tag{3.29}$$

where

$Q \cap D = \sharp$ of minutiae that occur in both images simultaneously [22].

For every minutia (X_1, Y_1) in the input fingerprint we try to find a corresponding minutia (X_2, Y_2) in the template. We say there exists a match between (X_1, Y_1) and (X_2, Y_2) if $\mid (X_1 - X_2) \mid < 2$ and $\mid (Y1 - Y2) \mid < 2$ and (ridgeOrientation1 - ridgeOrientation2) less that a threshold Th_1. If final minutiae match score is greater than a threshold Th_2 then we call it a match.

Chapter 4

EXPERIMENTAL RESULTS

4.1 Experimental Results

4.1.1 Fingerprint Classification

Initially we had started without considering the core point of the fingerprint. we used to segment the fingerprint into 5*5 block and used to treat middle 3*3 block irrespective of the position of the core point. The result of the classification algorithm implemented with that algorithm is given in table I.

	Left Loop	Right Loop	Whorl	Arch	Undecided[1](match)	Undecided(mismatch)
Left Loop	19	02	19	30	10	3
Right Loop	01	39	05	15	7	3
Whorl	17	28	21	9	7	3
Arch	2	7	1	43	5	0

Table I: Confusion Table (without considering core point)

Since the result that we got from earlier experiment were not that encouraging. It was clear that the pattern for classification can be derived only when we consider the 3*3 window around the core point,because the curvature of ridges is more prominent near the ridge only which helps in derivation of some distinct feature for each class of the fingerprint.

In later implementation we gave consideration to core point and we started selecting 3*3 window around the core point (given the core point lies in the 3*3 window of the segmented 5*5 image, if it lies outside we reject the fingerprint for classification). Since the fingerprint patterns is mostly visible around the core region the results were encouraging. The result with this experiment is shown in the table III.

	Left Loop	Right Loop	Whorl	Arch	Undecided[1](match)	Undecided(mismatch)	Rejected[2]
Left Loop	39	02	05	05	21	10	5
Right Loop	01	39	05	05	9	6	5
Whorl	07	08	41	09	12	3	5
Arch	2	4	1	41	5	0	5

Table II: Confusion Table (considering core point)

Total Numbers of Experimental Data:

a. Left Loop:87

b. Right Loop:70

c. Whorl:85

d. Arch:58

[1]Undecided means there is conflict of class

[2]The core point lies outside the central 3*3 window

[Note: These test results were carried out on synthetically generated fingerprint. Because of the limitations of the software the quality of the fingerprint was not that good. I hope the classification algorithm will perform better than what we are getting experimentally].

4.1.2 Fingerprint Matching

We tested the algorithms in our fingerprint verification on various parameters. We tested for different combinations of the algorithms (namely, histogram equalization, Fourier Transform and anisotropic filtering). In Fourier transform we are using a K parameter. On experimentation we found $K = 0.48$ yields good results. In case of histogram equalization we tested for various window sizes and found that window size of 32x32 yields best result. We also found that median filtering which we did had almost no effect on matching score as salt and pepper noise was mostly absent in the input images. We used 300 images for testing our algorithms. Also, we found that using Fourier Transform or Anisotropic Filtering in isolation leads to poor results [table III]. We found error margin in ridge orientation, $Th1 = 5^\circ$ yields good results. Fingerprint Matching Score threshold, $Th2 > 0.6$ gives good results.

Algorithm Combination	Acceptance (in%)	Rejection (in%)
Histogram (16) +Fourier +noAnisotropic	23.17	76.83
Histogram (32) +Fourier +noAnisotropic	67.68	32.22
Histogram (16) +noFourier +Anisotropic	13.45	86.55
Histogram (32) +noFourier +Anisotropic	41.32	58.68

Table III: Success Rate when using either Fourier Transform or Anisotropic Filtering

Fingerprint Type	$Th_2 = 0.5$(in%)			$Th_2 = 0.6$(in%)		
	Correct	FRR[1]	FAR[2]	Correct	FRR	FAR
Left Loop	86	12	04	85	15	0
Right Loop	78	12	10	86	14	0
Whorl	89	09	02	92	08	0
Plain Arch	85	09	06	89	11	0

Table IV: Fingerprint Verification Scores

Fingerprint Type	$Th_2 = 0.5$(in%)			$Th_2 = 0.6$(in%)		
	Correct	FRR	FIR[3]	Correct	FRR	FIR
Left Loop	86	08	06	83	16	01
Right Loop	76	12	12	82	14	04
Whorl	88	09	03	91	09	00
Plain Arch	83	08	09	87	11	02

Table V: Fingerprint Identification Scores

[1]False Rejection Ratio
[2]False Acceptance Ratio
[3]False Identification Ratio

Chapter 5

CONCLUSION

Our project has combined many methods to build a minutia extractor and a minutia matcher. The combination of multiple methods comes from a wide investigation into research paper. Also some novel changes like segmentation using Morphological operations, minutia marking with special consideration of false minutia removal, and matching in the unified x-y coordinate system after a two-step transformation are used in my project.Also some new minutia features has been suggested that can be implemented for minutia match that can enhance the reliability of the matching.

The water reservoir technique used for classification is totally new approach for the classification of the fingerprint. Primarily water reservoir technique is used for the segmentation of image and recognition of characters in hand written text. Since each class of fingerprint shows a definite pattern when water is poured in processed image, this property of fingerprint is exploited for the classification. In the current approach we are filling water from all the four directions, in future if we derive more features then we can classify the fingerprint using the flow from only one direction.

Bibliography

[1] Miller G. A. The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psychological Review,*, 63:81–97, 1956.

[2] Rudd Bolle Anil Jain, Lin Hong. On-line fingerprint verification. *IEEE transaction of Pattern Analysis and Machine Intelligence*, 19:302–314, 1997.

[3] Henry E. Classification and uses of fingerprints. *Routledge, London*, 1900.

[4] M. Kogan D. Dimitrov I Greenberg, S. Aladjem. Fingerprint image enhancement using filtering techniques. *International Conference on , Pattern Recognition*, 3:322–325, 2000.

[5] Zicheng Guo and Richard W. Hall. Parallel thinning with two-subiteration algorithms. *Communications of the ACM*, 32:359–373, 1989.

[6] C. J. Hilditch. Linear skeletons from square cupboards. *Machine Intell. (B.Meltzer and D. Michie, Eds.). New York; Amer. Elsevier*, 4:403–420, 1969.

[7] Wan Y. Hong, L. and A. K. Jain. Fingerprint image enhancement: Algorithm and performance evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:777–789, 1998.

[8] R Iliescu, S. Shinghal. Noise removal from binary patterns by using line adjacency graphs. *IEEE International Conference on Systems, Man, and Cybernetics,'Humans, Information and Technology'*, 1:79 – 84, 1994.

[9] A. K. Jain and L. Hong. On-line fingerprint verification. *Technical Report MSU-CPS-99- 19, Department of Computer Science, Michigan State University, East Lansing, Michigan*, 1999.

[10] Anil K Jain. Fundamentals of digital image processing. *Pearson Education Information and System Science Series*, 1:241–242, 2003.

[11] Kap Luk Chan Jinxiang Liu, Zhongyang Huang. Direct minutiae extraction from gray-level fingerprint image by relationship examination. *International Conference on Image Processing*, 2:427–430, 2000.

[12] Tojo A Kawagoe M. Fingerprint pattern classification. *Patt Recog*, 17:295–303, 1984.

[13] Dario Maio and Davide maltoni. Direct gray-scale minutiae detection in fingerprints. *IEEE Trans on PAMI*, 19:27–39, 1997.

[14] Adnan Amin Neil Yager. Fingerprint classification: a review. *Springer-Verlag London Ltd*, 7:77–93, 2004.

[15] Adnan Amin Neil Yager. Fingerprint verification based on minutia features: a review. *Springer-Verlag London Ltd*, 7:94–113, 2004.

[16] E. Newham. The biometric report. *New York: SJB Services*, 1995.

[17] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Trans. Systems, Man, and Cybernetics*, 9:62–66, 1979.

[18] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Trans. Systems, Man, and Cybernetics*, 9:62–66, 1979.

[19] A Pal, U. Sarkar. Document analysis and recognition. *Seventh International Conference on Pattern Recognition Letters*, 24:1183–1187, 2003.

[20] P. Perona and J. Malik. Scale-space and edge detection using ansotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:629–639, 2000.

[21] Kai Huang Qinzhi Zhang and Hong Yan. Fingerprint classification based on extraction and analysis of singularities and pseudoridges. *Conferences in Research and Practice in Information Technology Series*, 11:83–87, 2001.

[22] Changick Kim Sanpachai Huvanandana and Jenq-Neng Hwang. Reliable and fast fingerprint identification for security applications. *Information Processing Laboratory, Dept. Of Electrical Engineering, University of Washington, Seattle*, pages 504–506.

[23] Anil K. Jain Sharath Pankanti, Salil Prabhakar. On the individuality of fingerprints. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 24:1010–1025, 2002.

[24] R. Stefanelli and A. Rosenfeld. Some parallel thinning algorithms for digital pictures. *Journal of the ACM*, 18:255–264, 1971.

[25] Raymond Thai. Fingerprint image enhancement and minutiae extraction. *Thesis Paper*.

[26] A. Belad U. Pal and Ch. Choisy. Touching numeral segmentation using water reservoir concept. *Pattern Recognition Letters*, 24:261–272, 2003.

[27] P. S. P. Wang and Y. Y. Zhang. A fast and flexible thinning algorithm. *IEEE Transactions on Computers*, 38:741–745, 1989.

[28] G Watson C Wilson C. Candela. Neural networks fingerprint classification. *J Artif Neur Ntwks*, 1:203–228, 1993.

[29] Z.Guo and R.W.Hall. Fast fully parallel thinning algorithms. *CVGIP: Image Understanding*, 55:317–328, 1992.

[30] T.Y. Zhang and C.Y. Suen. A fast parallel algorithm for thinning digital patterns, comm. *ACM Commn.*, 27:236–239, 1984.

Appendix A

Least Distance Joining Algorithm (LDJA)

Least Joining Distance Algorithm is the novel approach to join the unconnected[1] ridges of the fingerprint. This algorithm is required for controlling the over-flow of water in water reservoir technique so that distinct patterns appear for the features selection. This algorithm joins the unconnected point to that point which is nearest in its n*n neighbourhood recursively till it is connected[2].

Two version of the LDJA are implemented. First the fingerprint is divided into four regions - Left Division, Right Division, Upper Division, Lower Division. In version one every unconnected point on fingerprint ridge is joined using LDJA based on the feature of the segmented part, whereas in version two only the points originating from four edges which are not connected (here connected means its other extreme points should be connected to some edge) are joined using LDJA.

The point on ridge which is not connected is identified and around it starting from 3*3 window upto n*n window (n<N and N*N is the size of fingerprint- we assume our fingerprint to be square) is drawn. Once the ridge point is found inside n*n window then we travel diagonally inside this widow towards the original ridge point which is to be connected. If we reach the column of the original image then we advance vertically and if we reach the row of the original image then we advance horizontally. Then same algorithm is repeated for the newly found ridge point till it merges with some ridge which is connected i.e. its other extreme meets with some edge.

The order of joining that we have implemented is as follows: North, South, East, West.

It is found that the version two gives better result than version one as there is over connection of ridges in version one which prevents it from getting any distinct pattern. If the quality of the fingerprint is very good then we may not need to join the unconnected as ridges around the core point are generally smooth and connected.

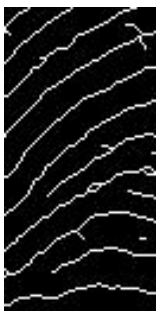

Figure A.1: Left Segmented Image(Unconnected)

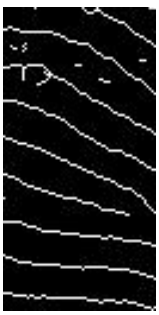

Figure A.2: Right Segmented Image(Unconnected)

Note that there is unnecessary connected part in version one which is shown in Fig[A.5]. The outcome of version two [Fig:A.6]is encouraging as there is no over connection.

Note the water flow algorithm gives good pattern to the fingerprint which is joined using LDJA ver2 as compared to LDJA ver1.

[1]The point is not connected to the ridge which ends in edge

[2]ridge ends in edge

Figure A.3: Upper Segmented Image(Unconnected)

Figure A.4: Lower Segmented Image(Unconnected)

Figure A.5: Upper Segmented Image(connected ver(1))

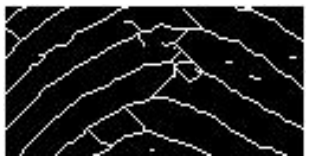

Figure A.6: Upper Segmented Image(connected ver(2))

Figure A.7: Upper Segmented Image(Filled ver(1))

Figure A.8: Upper Segmented Image(Filled ver(2))