

Fingerprint Recognition

Rohit Singh (Y6400), Utkarsh Shah (Y6510), Vinay Gupta (Y6534)

Department of Computer Science & engineering

Indian Institute of technology, Kanpur



Project Report

Computer Vision and Image Processing (CS676)

Guided By

Prof. Simant Dube

Date: 12/11/2009

Abstract

Our Term Project is to study and implement a fingerprint recognition system based on Minutiae based matching quite frequently used in various fingerprint algorithms and techniques. The approach mainly involves extraction of minutiae points from the sample fingerprint images and then performing fingerprint matching based on the number of minutiae pairings among two fingerprints in question.

Our implementation mainly incorporates image enhancement, image segmentation, feature (minutiae) extraction and minutiae matching. It finally generates a percent score which tells whether two fingerprints match or not. The project is coded in MATLAB.

Acknowledgement

We would like to express our sincere thanks and gratitude to Prof. Simant Dube for his suggestions, help and support. Also we take this opportunity to thank Prof. Amitabha Mukerjee for his valuable comments and feedback during our project presentations.

We would also like to appreciate our course TA's Amit Kumar Gupta and Rahul Gupta for the support.

Table of Contents

| | | |
|----------|-------------------------------------|-----------|
| 1 | Introduction | 5 |
| 1.1 | What is A Fingerprint | 5 |
| 1.2 | What is Fingerprint Recognition | 5 |
| 1.3 | Techniques for Fingerprint matching | 6 |
| 2 | Our Implementation | 6 |
| 2.1 | Design Description | 6 |
| 3 | Minutiae Extraction | 7 |
| 3.1 | Fingerprint Image Enhancement | 7 |
| 3.2 | Fingerprint Image Segmentation | 9 |
| 3.3 | Final Minutiae Extraction | 10 |
| 4 | Minutia matching | 12 |
| 4.1 | Minutiae Alignment | 12 |
| 4.2 | Minutiae Match | 13 |
| 5 | Experimentation Results | 13 |
| 5.1 | Performance Evaluation Indexes | 13 |
| 5.2 | Experiment Analysis | 13 |
| 6 | Conclusion | 14 |
| | References | 14 |

1 Introduction

Fingerprint recognition or fingerprint authentication refers to the automated method of verifying a match between two human fingerprints. Fingerprints are one of many forms of biometrics used to identify an individual and verify their identity. Because of their uniqueness and consistency over time, fingerprints have been used for over a century, more recently becoming automated (i.e. a biometric) due to advancement in computing capabilities. Fingerprint identification is popular because of the inherent ease in acquisition, the numerous sources (ten fingers) available for collection, and their established use and collections by law enforcement and immigration.

1.1 What is a Fingerprint?

A fingerprint is the feature pattern of one finger (Figure 1.1). It is an impression of the friction ridges and furrows on all parts of a finger. These ridges and furrows present good similarities in each small local window, like parallelism and average width.



Figure 1.1 Fingerprint image from a sensor

However, shown by intensive research on fingerprint recognition, fingerprints are not distinguished by their ridges and furrows, but by features called Minutiae, which are some abnormal points on the ridges (Figure 1.2). Among the variety of minutiae types reported in literatures, two are mostly significant and in heavy usage:

- Ridge ending - the abrupt end of a ridge
- Ridge bifurcation - a single ridge that divides into two ridges

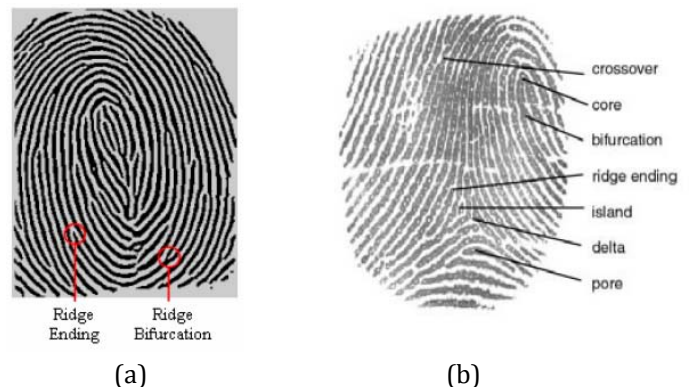


Figure 1.2(a) two important minutiae features
(b) Other minutiae features

1.2 What is Fingerprint Recognition?

Fingerprint recognition (sometimes referred to as dactyloscopy) is the process of comparing questioned and known fingerprint against another fingerprint to determine if the impressions are from the same finger or palm. It includes two sub-domains: one is fingerprint verification and the other is fingerprint identification (Figure 1.3). In addition, different from the manual approach for fingerprint recognition by experts, the fingerprint recognition here is referred as AFRS (Automatic Fingerprint Recognition System), which is program-based.

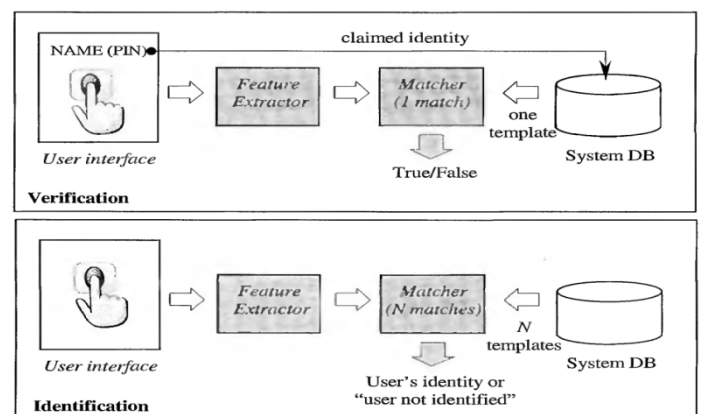


Figure 1.3 Verification vs. Identification

However, in all fingerprint recognition problems, either verification(one to one matching) or identification(one to many matching), the underlining principles of well defined representation of a fingerprint and matching remains the same.

1.3 Fingerprint matching techniques

The large number of approaches to fingerprint matching can be coarsely classified into three families.

- **Correlation-based matching:** Two fingerprint images are superimposed and the correlation between corresponding pixels is computed for different alignments (e.g. various displacements and rotations).
- **Minutiae-based matching:** This is the most popular and widely used technique, being the basis of the fingerprint comparison made by fingerprint examiners. Minutiae are extracted from the two fingerprints and stored as sets of points in the two-dimensional plane. Minutiae-based matching essentially consists of finding the alignment between the template and the input minutiae sets that results in the maximum number of minutiae pairings
- **Pattern-based (or image-based) matching:** Pattern based algorithms compare the basic fingerprint patterns (arch, whorl, and loop) between a previously stored template and a candidate fingerprint. This requires that the images be aligned in the same orientation. To do this, the algorithm finds a central point in the fingerprint image and centers on that. In a pattern-based algorithm, the template contains the type, size, and orientation of patterns within the aligned fingerprint image. The candidate fingerprint image is graphically compared with the template to determine the degree to which they match.

In Our project we have implemented a minutiae based matching technique. This approach has been intensively studied, also is the backbone of the current available fingerprint recognition products.

2 Our Implementation

We have concentrated our implementation on Minutiae based method. In particular we are interested only in two of the most important minutia features i.e. Ridge Ending and Ridge bifurcation. (Figure 2.1)

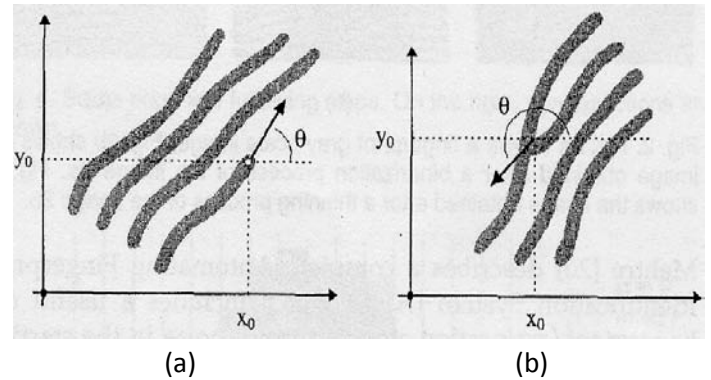


Figure 2.1(a) Ridge Ending, (b) Ridge Bifurcation

The outline of our approach can be broadly classified into 2 stages - Minutiae Extraction and Minutiae matching. Figure 2.2 illustrates the flow diagram of the same.

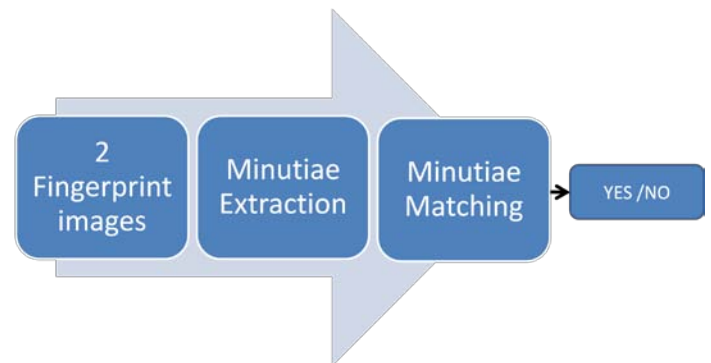


Figure 2.2 System Flow Diagram

The system takes in 2 input fingerprints to be matched and gives a percentage score of the extent of match between the two. Based on the score and threshold match value it can distinguish whether the two fingerprints match or not. The input fingerprints are taken from the database provided by FVC2004 (Fingerprint Verification Competition 2004).

2.1 Design Description

The above system is further classified into various modules and sub-modules as given in Figure 2.3.

Minutia extraction includes Image Enhancement, Image Segmentation and Final Extraction processes while Minutiae matching include Minutiae Alignment and Match processes.

Fingerprint Recognition

Rohit Singh (Y6400), Utkarsh Shah (Y6510), Vinay Gupta (Y6534)

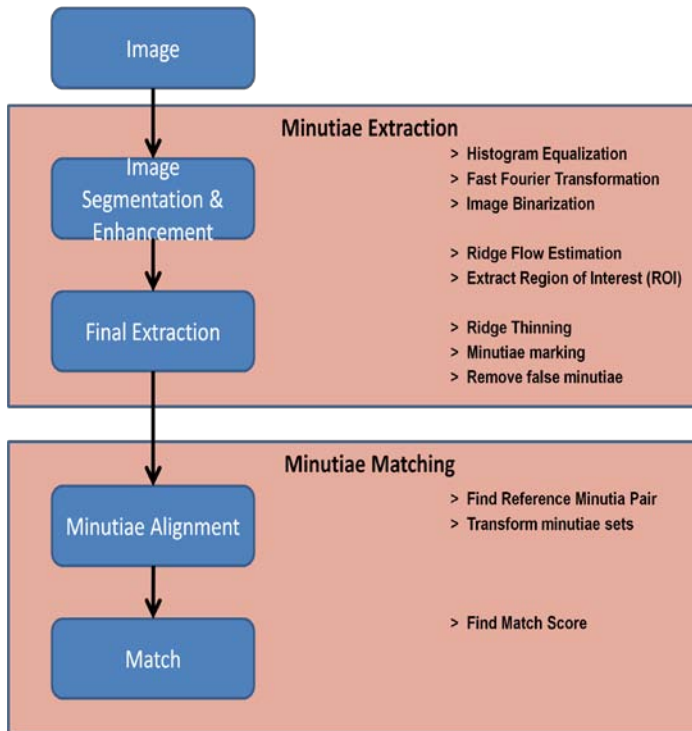


Figure 2.3 Detailed Design Description

Under image enhancement step Histogram Equalization, Fast Fourier Transformation increases the quality of the input image and Image Binarization converts the grey scale image to a binary image.

Then image segmentation is performed which extracts a Region of Interest using Ridge Flow Estimation and MATLAB's morphological functions.

Thereafter the minutia points are extracted in the Final Extraction step by Ridge Thinning, Minutia Marking and Removal of False Minutiae processes.

Using the above Minutia Extraction process we get the Minutiae sets for the two fingerprints to be matched. Minutiae Matching process iteratively chooses any two minutiae as a reference minutia pair and then matches their associated ridges first. If the ridges match well, two fingerprint images are aligned and matching is conducted for all remaining minutia to generate a Match Score.

3 Minutiae Extraction

As described earlier the Minutiae extraction process includes image enhancement, image segmentation and final Minutiae extraction.

3.1 Fingerprint Image Enhancement

The first step in the minutiae extraction stage is Fingerprint Image enhancement. This is mainly done to improve the image quality and to make it clearer for further operations. Often fingerprint images from various sources lack sufficient contrast and clarity. Hence image enhancement is necessary and a major challenge in all fingerprint techniques to improve the accuracy of matching. It increases the contrast between ridges and furrows and connects the some of the false broken points of ridges due to insufficient amount of ink or poor quality of sensor input.

In our project we have implemented three techniques: Histogram Equalization, Fast Fourier Transformation and Image Binarization.

3.1.1 Histogram Equalization

Histogram equalization is a technique of improving the global contrast of an image by adjusting the intensity distribution on a histogram. This allows areas of lower local contrast to gain a higher contrast without affecting the global contrast. Histogram equalization accomplishes this by effectively spreading out the most frequent intensity values. The original histogram of a fingerprint image has the bimodal type (Figure 3.1(a)), the histogram after the histogram equalization occupies all the range from 0 to 255 and the visualization effect is enhanced (Figure 3.1(b)).

The result of the histogram equalization is shown in figure 3.2.

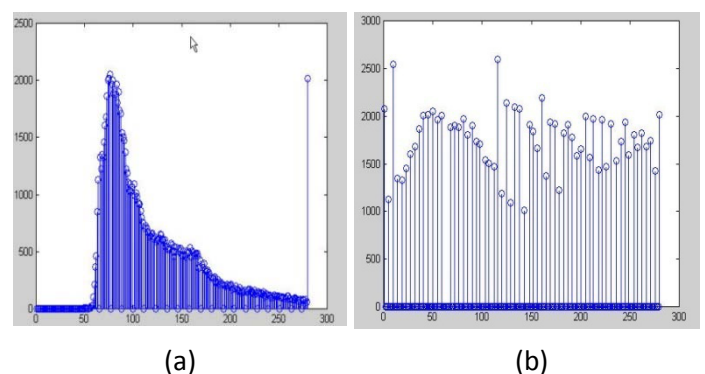


Figure 3.1(a) Original histogram, (b) Histogram after equalization



(a) (b)

Figure 3.2(a) Original Image, (b) Enhanced Image after histogram equalization



(a) (b)

Figure 3.3(a) Enhanced Image after FFT, (b) Image before FFT

The enhanced image after FFT has the improvements as some falsely broken points on ridges get connected and some spurious connections between ridges get removed.

3.1.2 Fast Fourier Transformation

In this method we divide the image into small processing blocks (32 x 32 pixels) and perform the Fourier transform according to equation:

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

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

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 = $\text{abs}(F(u, v)) = |F(u, v)|$.

So we get the enhanced block according to the equation:

$$g(x, y) = F^{-1} \left\{ F(u, v) \times |F(u, v)|^k \right\} \quad (2)$$

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

$$f(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) \times \exp \left\{ j2\pi \times \left(\frac{ux}{M} + \frac{vy}{N} \right) \right\} \quad (3)$$

For $x = 0, 1, 2, \dots, 31$ and $y = 0, 1, 2, \dots, 31$.

The k in formula (2) is an experimentally determined constant, which we choose $k=0.45$ to calculate. A high value of k improves the appearance of the ridges by filling up small holes in ridges, but too high value of k can result in false joining of ridges which might lead to a termination become a bifurcation.

Figure 3.3 presents the image after FFT enhancement.

3.1.3 Image Binarization

Image Binarization is a process which transforms the 8-bit Gray image to a 1-bit image with 0-value for ridges and 1-value for furrows. After the operation, ridges in the fingerprint are highlighted with black color while furrows are white.

A locally adaptive binarization method is performed to binarize the fingerprint image. In this method image is divided into blocks of 16 x 16 pixels. A pixel value is then set to 1 if its value is larger than the mean intensity value of the current block to which the pixel belongs (Figure 3.4).



(a) (b)

Figure 3.4(a) Binarized Image after FFT, (b) Image before binarization

3.2 Fingerprint Image Segmentation

After image enhancement the next step is fingerprint image segmentation. In general, only a Region of Interest (ROI) is useful to be recognized for each fingerprint image. The image area without effective ridges and furrows is first discarded since it only holds background information. Then the bound of the remaining effective area is sketched out since the minutiae in the bound region are confusing with those spurious minutiae that are generated when the ridges are out of the sensor.

To extract the region of interest, two steps are followed: Block direction estimation and ROI extraction by Morphological methods.

3.2.1 Block direction estimation

Here the fingerprint image is divided into blocks of size 16 x 16 pixels (W x W) after which the block direction of each block is calculated according to the algorithm:

- I. Calculate the gradient values along x-direction (g_x) and y-direction (g_y) for each pixel of the block. Two Sobel filters are used to fulfill the task.
- II. For each block, use following formula to get the Least Square approximation of the block direction.

$$\tan 2\beta = \frac{2 \sum \sum (g_x * g_y)}{\sum \sum (g_x^2 - g_y^2)}$$

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.

$$\tan 2\theta = \frac{2 \sin \theta \cos \theta}{\cos^2 \theta - \sin^2 \theta}$$

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:

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

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 diagram (Figure 3.5).

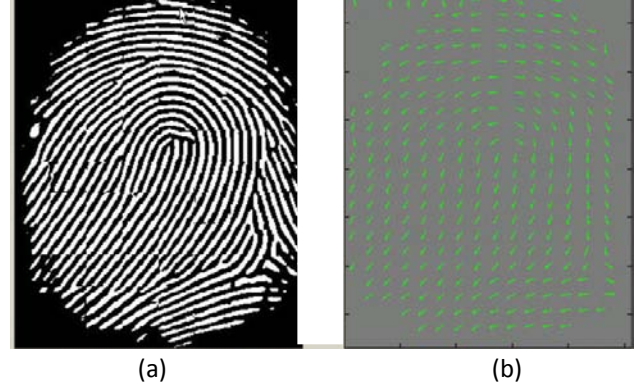


Figure 3.4(a) Binarized Image, (b) Direction map of image

3.2.2 ROI Extraction by Morphological operations

ROI extraction is done using two Morphological operations called OPEN and CLOSE. The OPEN operation can expand images and remove peaks introduced by background noise (Figure 3.6). The 'CLOSE' operation can shrink images and eliminate small cavities (Figure 3.7).

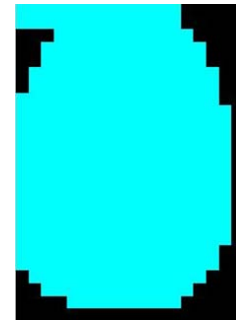


Figure 3.5 Original image area

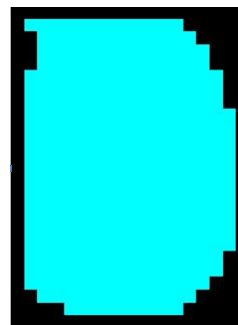


Figure 3.6 After CLOSE

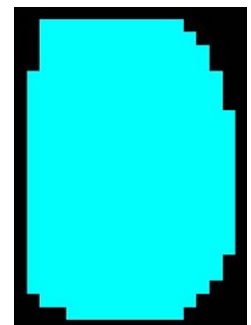


Figure 3.7 After OPEN



Figure 3.8 Final ROI

Figure 3.8 show the final ROI of the fingerprint which is the bound area after subtraction of the closed area from the opened area. Then the leftmost, rightmost, uppermost and bottommost blocks out of the bound area are discarded.

3.3 Final Minutiae Extraction

Now that we have enhanced the image and segmented the required area, the job of minutiae extraction closes down to four operations: Ridge Thinning, Minutiae Marking, False Minutiae Removal and Minutiae Representation.

3.3.1 Ridge Thinning

In this process we eliminate the redundant pixels of ridges till the ridges are just one pixel wide. This is done using the MATLAB's built in morphological thinning function.

```
bwmorph(binaryImage,'thin',Inf)
```

The thinned image is then filtered, again using MATLAB's three morphological functions to remove some H breaks, isolated points and spikes (Figure 3.9).

```
bwmorph(binaryImage,'hbreak',k)
```

```
bwmorph(binaryImage,'clean',k)
```

```
bwmorph(binaryImage,'spur',k)
```

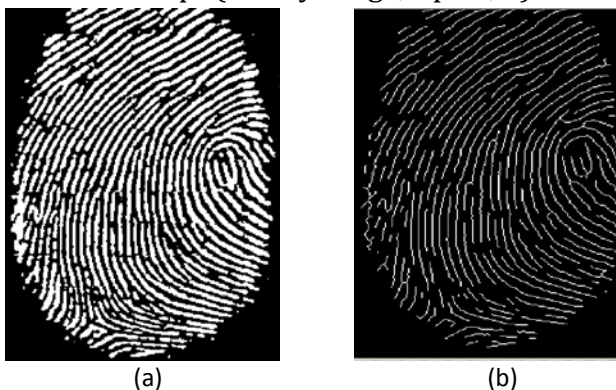


Figure 3.9(a) Image before, (b) Image after thinning

3.3.2 Minutiae Marking

Minutiae marking is now done using templates for each 3 x 3 pixel window as follows.

If the central pixel is 1 and has exactly 3 one-value neighbors, then the central pixel is a ridge branch (Figure 3.10).

| | | |
|---|---|---|
| 0 | 1 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 1 |

Figure 3.10

If the central pixel is 1 and has only 1 one-value neighbor, then the central pixel is a ridge ending (Figure 3.11).

| | | |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 1 |

Figure 3.11

There is one case where a general branch may be triple counted (Figure 3.12). Suppose both the uppermost pixel with value 1 and the rightmost pixel with value 1 have another neighbor outside the 3x3 window due to some left over spikes, so the two pixels will be marked as branches too, but actually only one branch is located in the small region. Thus this is taken care of.

| | | |
|---|---|---|
| 0 | 1 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |

Figure 3.12

3.3.3 False Minutiae Removal

At this stage false ridge breaks due to insufficient amount of ink & ridge cross connections due to over inking are not totally eliminated. Also some of the earlier methods introduce some spurious minutia points in the image. So to keep the recognition system consistent these false minutiae need to be removed.

Here we first calculate the inter ridge distance D which is the average distance between two neighboring ridges. For this scan each row to calculate the inter ridge distance using the formula:

$$\text{Inter ridge distance} = \frac{\text{sum all pixels with h value 1}}{\text{row length}}$$

Finally an averaged value over all rows gives D .

All we label all thinned ridges in the fingerprint image with a unique ID for further operation using a MATLAB morphological operation BWLABEL.

Now the following 7 types of false minutia points are removed using these steps (Figure 3.13).

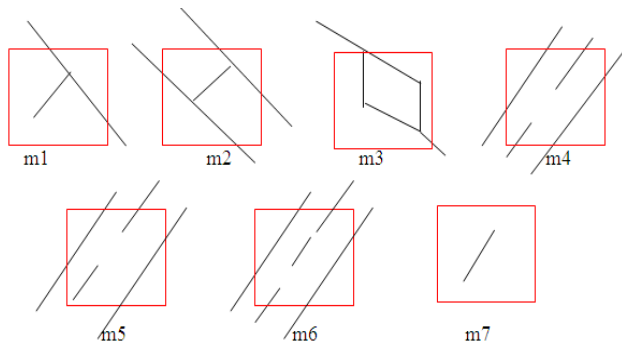


Figure 3.13

- If $d(\text{bifurcation}, \text{termination}) < D$ & the 2 minutia are in the same ridge then remove both of them (case m1)
 - If $d(\text{bifurcation}, \text{bifurcation}) < D$ & the 2 minutia are in the same ridge then remove both of them (case m2, m3)
 - If $d(\text{termination}, \text{termination}) \approx D$ & the their directions are coincident with a small angle variation & no any other termination is located between the two terminations then remove both of them (case m4, m5, m6)
 - If $d(\text{termination}, \text{termination}) < D$ & the 2 minutia are in the same ridge then remove both of them (case m7)
- where $d(X, Y)$ is the distance between 2 minutia points.

3.3.4 Minutiae Representation

Finally after extracting valid minutia points from the fingerprint they need to be stored in some form of representation common for both ridge ending and bifurcation.

So each minutia is completely characterized by the following parameters 1) x-coordinate, 2) y-coordinate, 3) orientation and 4) ridge associated with it (Figure 3.14)

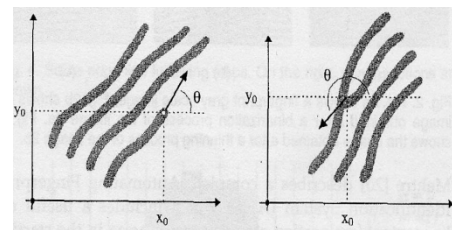


Figure 3.14

Actually a bifurcation can be broken down to three terminations each having their own x-y coordinates (pixel adjacent to the bifurcating pixel), orientation and an associated ridge.

The orientation of each termination (tx, ty) is estimated by following method. Track a ridge segment whose starting point is the termination and length is D . Sum up all x-coordinates of points in the ridge segment. Divide above summation with D to get sx . Then get sy using the same way.

Get the direction from: $\tan^{-1} \frac{sy - ty}{sx - tx}$

Results after the minutia extraction stage (Figure 3.15-3.17)

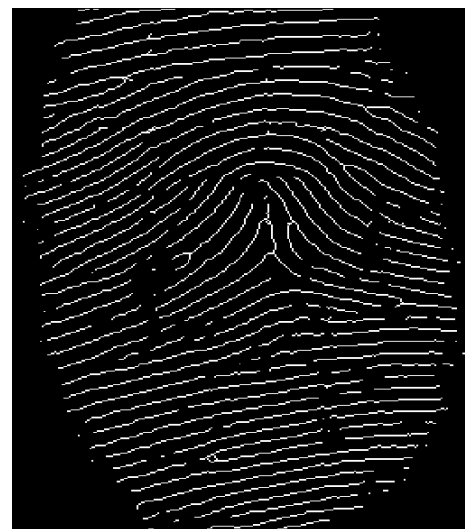


Figure 3.15 Thinned image

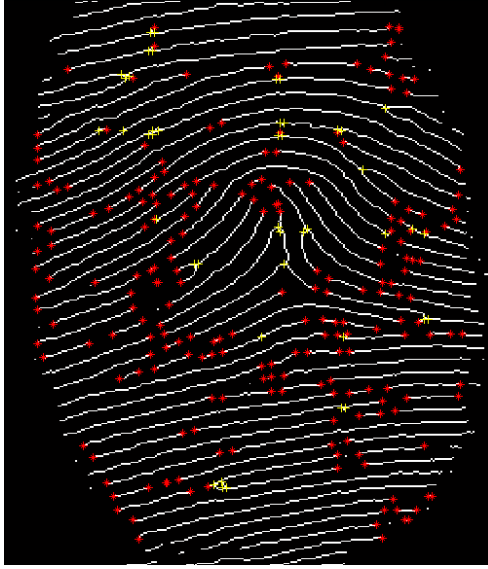


Figure 3.16 Minutiae after marking



Figure 3.17 Real Minutiae after false removal

4 Minutiae Matching

After successfully extracting the set of minutia points of 2 fingerprint images to be tested, we perform Minutiae Matching to check whether they belong to the same person or not.

We use an iterative ridge alignment algorithm to first align one set of minutiae w.r.t other set and then carry-out an elastic match algorithm to count the number of matched minutia pairs.

4.1 Minutiae Alignment

Let I_1 & I_2 be the two minutiae sets given by,

$$I_1 = \{m_1, m_2, \dots, m_M\} \text{ where } m_i = (x_i, y_i, \theta_i)$$

$$I_2 = \{m'_1, m'_2, \dots, m'_N\} \text{ where } m'_i = (x'_i, y'_i, \theta'_i)$$

Now we choose one minutia from each set to find the ridge correlation factor between them. The ridge associated with each minutia is represented as a series of x-coordinates (x_1, x_2, \dots, x_n) of the points on the ridge. A point is sampled per ridge length L starting from the minutia point, where the L is the average inter-ridge length. And n is set to 10 unless the total ridge length is less than $10 \cdot L$.

So the similarity of correlating the two ridges is derived from:

$$S = \sqrt{\frac{\sum_{i=0}^m x_i X_i}{\sum_{i=0}^m x_i^2 X_i^2}}$$

where (x_i, \dots, x_n) and (X_i, \dots, X_n) are the set of x-coordinates for each of the 2 minutia chosen. And m is minimal one of the n and N value. If the similarity score is larger than 0.8, then go to step 2, otherwise continue to match the next pair of ridges.

2. The approach is to transform each set according to its own reference minutia and then do match in a unified x-y coordinate.

Let $M(x, y, \theta)$ be reference minutia found from step 1 (say from I_1). For each fingerprint, translate and rotate all other minutiae (x_i, y_i, θ_i) with respect to the M according to the following formula:

$$\begin{pmatrix} x_{i_new} \\ y_{i_new} \\ \theta_{i_new} \end{pmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_i - x \\ y_i - y \\ \theta_i - \theta \end{bmatrix}$$

The new coordinate system is originated at reference minutia M and the new x-axis is coincident with the direction of minutia M . No scaling effect is taken into account by assuming two fingerprints from the same finger have nearly the same size.

So we get transformed sets of minutiae I'_1 & I'_2

4.2 Minutiae Match

An elastic string (x, y, θ) match algorithm is used to find number of matched minutia pairs among I_1' & I_2' .

According to the elastic string match algorithm minutia m_i in I_1' and a minutia m_j in I_2' are considered "matching," if the spatial distance (sd) between them is smaller than a given tolerance r_0 and the direction difference (dd) between them is smaller than an angular tolerance Θ_0 .

$$sd = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \leq r_0$$

$$dd = \min(|\theta_i - \theta_j|, 360 - |\theta_i - \theta_j|) \leq \Theta_0$$

Let $mm(.)$ be an indicator function that returns 1 in the case where the minutiae m_i and m_j match according to above equations.

$$mm(m_i, m_j) = \begin{cases} 1, & sd(m_i, m_j) \leq r_0 \text{ and } dd(m_i, m_j) \leq \Theta_0 \\ 0, & \text{otherwise} \end{cases}$$

Now the total number of matched minutiae pair given by,

$$\text{num (matched minutiae)} = \sum mm(m_i, m_j)$$

and final match score is given by,

$$\text{Match Score} = \frac{\text{num (matched minutiae)}}{\max(\text{num of minutiae in } I_1, I_2)}$$

5 Experimental Results

5.1 Performance Evaluation Index

Two indexes are well accepted to determine the performance of a fingerprint recognition system:

- **False Rejection Rate (FRR):** For an image database, each sample is matched against the remaining samples of the same finger to compute the False Rejection Rate
- **False Acceptance Rate (FAR):** Also the first sample of each finger in the database is matched against the first sample of the remaining fingers to compute the False Acceptance Rate.

5.2 Experiment Analysis

A fingerprint database from the FVC2002 (Fingerprint Verification Competition 2002) is used to test the program's performance. A series of correct and incorrect match score is recorded.

Following is the distribution curve obtained after experiments (Figure 4.1).

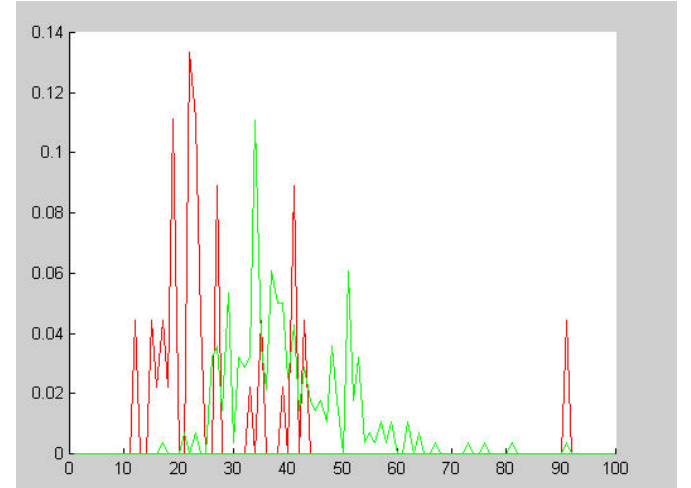


Figure 5.1 Distribution of Correct Scores and Incorrect Scores (Red: Incorrect Scores, Green: Correct Scores)

In our experiments distribution curve gives an average correct match score of about 30 and average incorrect match score of 25 on the database chosen.

The FAR and FRR curve as claimed by the algorithm is shown under (Figure 5.2)

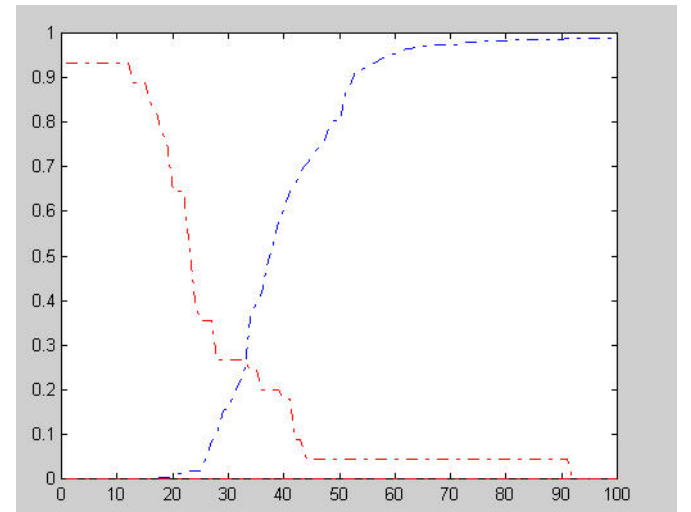


Figure 5.1 FRR and FAR curve (Red: FAR, Blue: FRR)

In our experiments FAR and FRR values were 30-35% approximately. Thus at a threshold match score of

about 28 the verification rate of the algorithm is about 65-70%.

The relatively low percentage of verification rate is due to poor quality of images in the database and the inefficient matching algorithm which lead to incorrect matches.

6 Conclusion

The above implementation was an effort to understand how Fingerprint Recognition is used as a form of biometric to recognize identities of human beings. It includes all the stages from minutiae extraction from fingerprints to minutiae matching which generates a match score. Various standard techniques are used in the intermediate stages of processing.

The relatively low percentage of verification rate as compared to other forms of biometrics indicates that the algorithm used is not very robust and is vulnerable to effects like scaling and elastic deformations. Various new techniques and algorithm have been found out which give better results.

Also a major challenge in Fingerprint recognition lies in the pre processing of the bad quality of fingerprint images which also add to the low verification rate.

References

- Handbook of Fingerprint Recognition by Davide Maltoni, Dario Maio, Anil K. Jain & Salil Prabhakar
- Fingerprint Recognition, Paper by WUZHILI (Department of Computer Science & Engineering, Hong Kong Baptist University) 2002
- Fingerprint Classification and Matching by Anil Jain (Department of Computer Science & Engineering, Michigan State University) & Sharath Pankanti (Exploratory Computer Vision Group IBM T. J. Watson Research Centre) 2000
- Fingerprint database - FVC2002 (Fingerprint Verification Competition 2002)
- Wikipedia link - http://en.wikipedia.org/wiki/Fingerprint_recognition