

# A Review on Fingerprint Recognition Technique Using Real Minutia Identification

Komal Sharma<sup>1</sup>, Vinod Kumar Singla<sup>2</sup>

<sup>1</sup>Guru Kashi University, Department of CSE,  
 Talwandi Sabo, Punjab, India  
 Parisharma.sharma72@gmail.com

<sup>2</sup>AP, Guru Kashi University, Department of CSE,  
 Talwandi Sabo, Punjab, India  
 Vinodsingla29@gmail.com

**Abstract:** Fingerprint matching is one of the most efficient and successful technique for human identification with easy to implement and get accurate results. This paper presents a review on study and implementation of a fingerprint recognition system based on Minutiae based matching quite frequently used in various fingerprint algorithms and techniques. This approach mainly involves extraction of unique identification points called minutiae points from the captured fingerprint images and then performing fingerprint matching based on the number of minutiae pairings among two fingerprints in question. The spurious minutiae are removed by identifying the false percept minutiae at the extraction stage.

**Keywords:** Fingerprint Extraction, Authentication, Identification, Minutiae, Biometric, Fingerprint Recognition.

## 1. Introduction

Fingerprint recognition or fingerprint authentication refers to the automatic method of identifying and verifying a match between two human fingerprints [1]. Fingerprints are one of many available 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 centuries, more recently becoming automated (i.e. A biometric) due to advancement in computing capabilities like image capturing. 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)[3]. 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. [5]



Figure 1.1: Fingerprint image from a Sensors.

However, shown by intensive research on fingerprint recognition, fingerprints are not distinguished by their ridges and furrows, but by features called Minutia (Figure 1.2). Among the variety of minutia types reported in literatures, two are mostly significant and in heavy usage: Ridge ending - the abrupt end of a ridge and Ridge bifurcation - a single ridge that divides into two ridges [6].

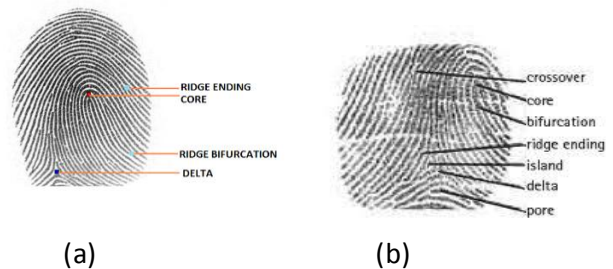


Figure 1.2: (a) Two important minutia features  
 (b) Other minutiae features

## 2. 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 [10]. 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. [7]

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.

### Method Implementation

The implementation is done on Minutiae based method. In particular this technique uses only in two of the most important minutia features i.e. Ridge Ending and Ridge bifurcation. (Figure 2.1)

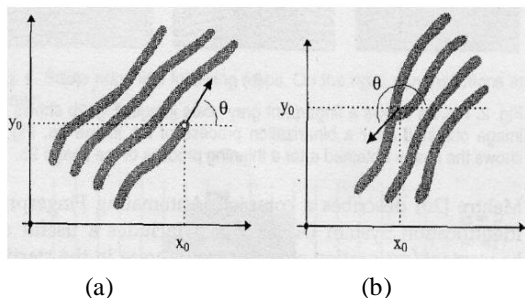


Figure 2.1: (a) Ridge Ending, (b) Ridge Bifurcation

### Design Description

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

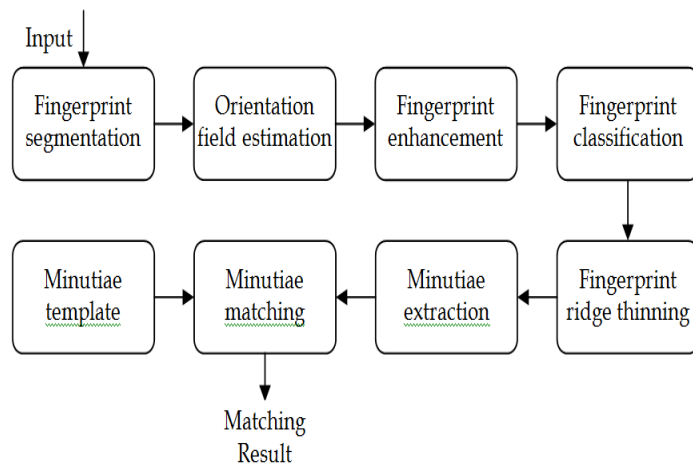


Figure 2.2: Detailed Design Description

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.

### 2.1 Implemented Techniques

#### 2.1.1 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. [9]

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 2.3(a)), the histogram after the histogram equalization occupies all the range from 0 to 255 and the visualization effect is enhanced (Figure 2.3(b)).

The result of the histogram equalization is shown in figure 2.4.

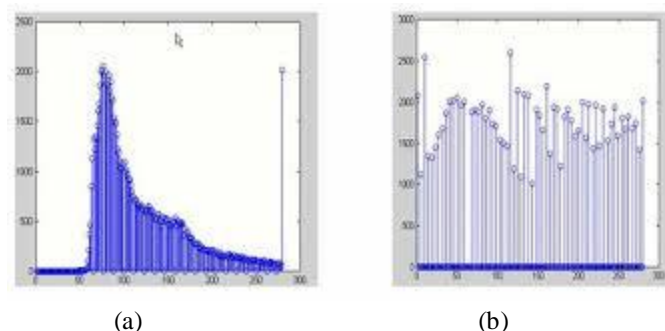
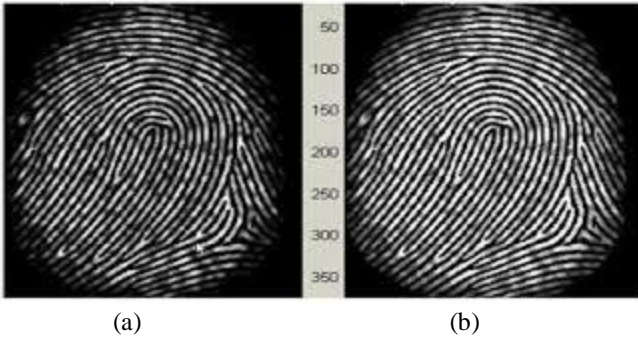


Figure 2.3: (a) Original histogram, (b) Histogram after equalization



**Figure 2.4:** (a) Original Image, (b) Enhanced Image after histogram equalization

### 2.1.2 Fast Fourier Transformation

In this method the image is divided 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:

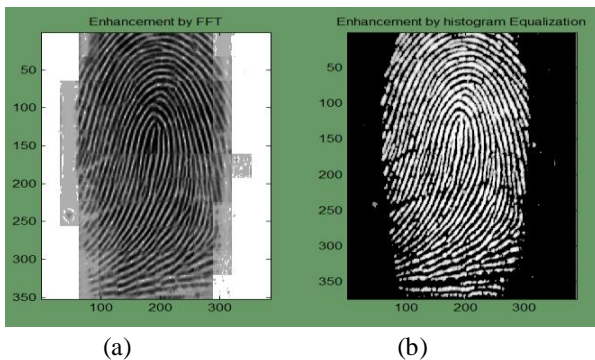
$$g(x, y) = F^{-1} \left\{ |F(u, v)|^k \times |F(u, v)| \right\} \quad (2)$$

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

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

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.

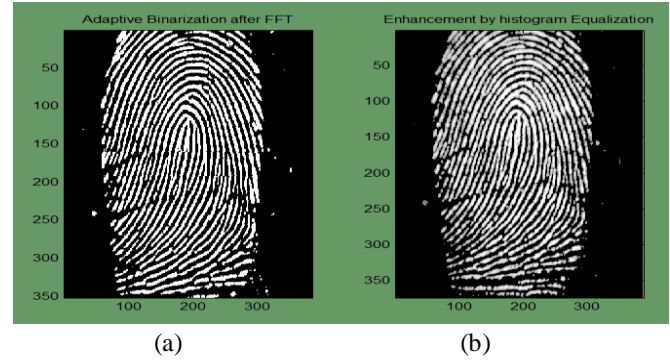


**Figure 2.5:** (a) Enhanced Image after FFT, (b) Image before FFT

### 2.1.3 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.[2] After the operation, ridges in the fingerprint are highlighted with black color while furrows are white. [12]

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 2.6).



**Figure 2.6:** (a) Binarized Image after FFT, (b) Image before binarization

## 2.2 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.

### 2.2.1 Block direction estimation

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

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.

For each block, use following formula to get the Least Square approximation of the block direction  $\text{tg}2\beta = 2 \frac{\sum \sum (g_x * g_y)}{\sum \sum (g_x^2 - g_y^2)}$  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.

$$\text{tg}2\theta = 2 \sin\theta \cos\theta / (\cos^2\theta - \sin^2\theta)$$

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:

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

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 2.7).

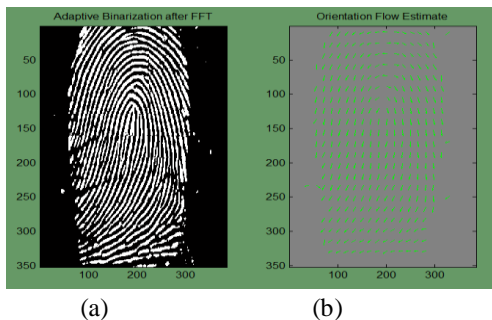


Figure 2.7: (a) Binarized Image, (b) Direction map of image

### 2.2.2 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 2.9). [2] The 'CLOSE' operation can shrink images and eliminate small cavities (Figure 2.10).

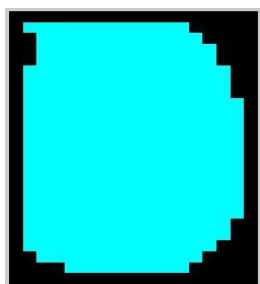
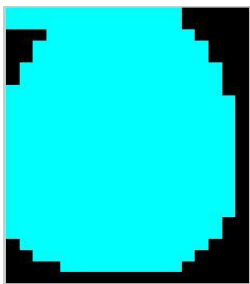


Figure 2.8: Original image area

Figure 2.9: After CLOSE

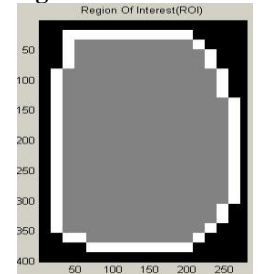
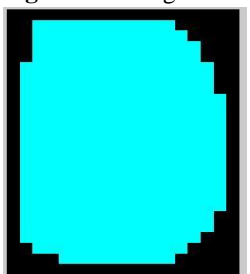


Figure 2.10: After OPEN

Figure 2.11: Final ROI

Figure 2.11 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.

### 2.3 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.[13]

#### 2.3.1 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. [8] `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 2.12). [12]

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

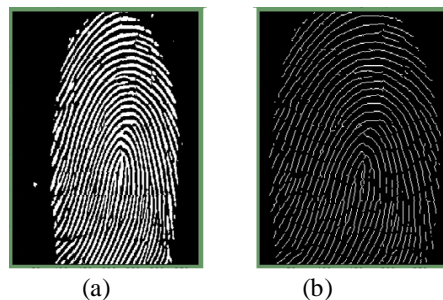


Figure 2.12: (a) Image before, (b) Image after thinning

#### 2.3.2 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 2.13). [7]

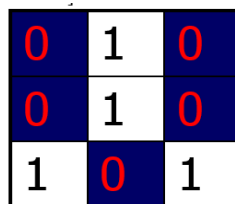


Figure 2.13

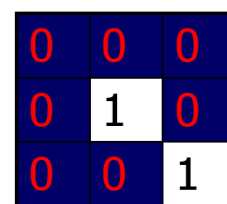


Figure 2.14

If the central pixel is 1 and has only 1 one-value neighbor, then the central pixel is a ridge ending (Figure 2.14).

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.

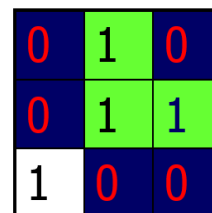


Figure 2.15

#### 2.3.3 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. [9]

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:

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

Finally an averaged value over all rows gives  $D$ . Now the following 7 types of false minutia points are removed using these steps (Figure 2.16). [3]

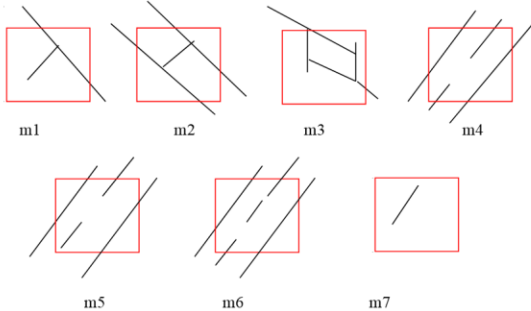


Figure 2.16

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.

Generally all the 7 cases are matched for presence and detection of the false minutia but it takes a lot of time at the real minutia extraction stage and the fingerprint matching stage. Thereby wasting the essential CPU time in particular regions where all the 7 cases does not exist. [13]

### 2.3.4 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 2.17)

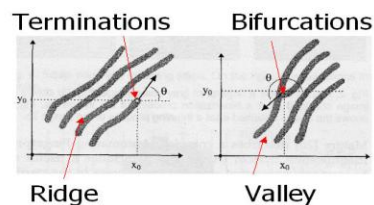


Figure 2.17

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 who's 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  $s_x$ . Then get  $s_y$  using the same way. [9]

$$\text{Get the direction from: } \tan^{-1} \frac{s_y - t_y}{s_x - t_x}$$

Results after the minutia extraction stage (Figure 2.18-2.19)



Figure 2.18: Minutiae after marking



Figure 2.19: Real Minutiae after false removal

## 3. 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.[8]

### 3.1 Minutiae Alignment

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

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

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

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

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

Where  $(x_i..x_n)$  and  $(X_i..X_n)$  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.

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

For each fingerprint, translate and rotate all other minutiae  $(x_i, y_i, \theta_i)$  with respect to the M according to the following formula:

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

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  $I_1'$  &  $I_2'$

### 3.2 Minutiae Match

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  $\theta_0$ . [11]

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

$$dd = \min(|\theta_i - \theta_j|, 360 - |\theta_i - \theta_j|) \leq \theta_0$$

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.[1]

$$mm(m_i, m_j) = \begin{cases} 1, & sd(m_i, m_j) \leq r_0 \text{ and } dd(m_i, m_j) \leq \theta_0 \\ 0, & \text{otherwise} \end{cases}$$

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,

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

## RESULTS

### 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.[2]

**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.[2]

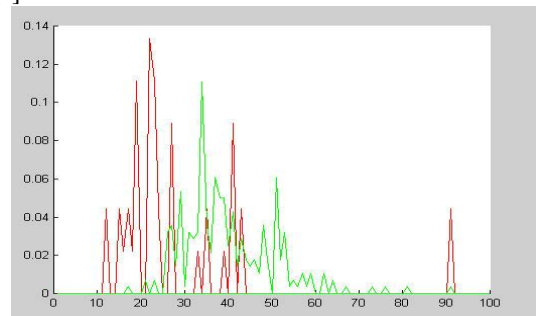


Figure - Distributions of Correct Scores and Incorrect Scores (Red: Incorrect Scores, Green: Correct Scores)

In conducted 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 3.1)

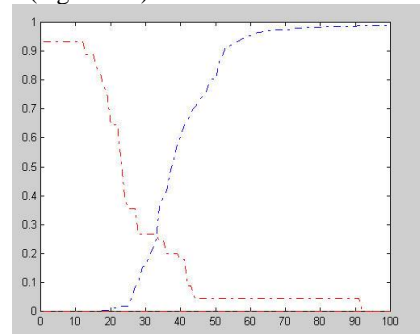


Figure 3.2 FRR and FAR curve (Red: FAR, Blue: FRR)

In conducted 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.

## 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

[1] Handbook of Fingerprint Recognition by Davide Maltoni, Dario Maio, Anil K. Jain & Salil Prabhakar

[2] Fingerprint Recognition, Paper by WUZHILI (Department of Computer Science & Engineering, Hong Kong Baptist University) 2002.

[3] 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.

[4] Wikipedia link - [http://en.wikipedia.org/wiki/Fingerprint\\_recognition](http://en.wikipedia.org/wiki/Fingerprint_recognition)

[5] Amir Najmi Jia Li and Robert M. Gray. Image classification by a two dimensional hidden markov model. Information Systems Laboratory, EE Dept. Stanford University, pages 3313 - 3316, 1999.

[6] Jain, R. Bolle, and S. Pankanti, "Biometrics Personal Identification in Networked Society", Kluwer Academic Publishers New York, Boston, Dordrecht, London, Moscow, pp. 1-64, 2002.

[7] A. K. Jain, F. Patrick, A. Arun, "Handbook of Biometrics", Springer science + Business Media, LLC, 1st edition, pp. 1-42, 2008.

[8] D. Maio, and D. Maltoni, "Direct gray-scale minutiae detection in fingerprints", IEEE Transactions Pattern Analysis and Machine Intelligence, vol. 19(1), pp. 27-40, 1997.

[9] D. Maltoni, D. Maio, and A. Jain, S. Prabhakar, "4.3: Minutiae-based Methods' (extract) from Handbook of Fingerprint Recognition", Springer, New York, pp. 141-144, 2003.

[10] E. Hastings, "A Survey of Thinning Methodologies", Pattern analysis and Machine Intelligence, IEEE Transactions, vol. 4, Issue 9, pp. 869-885, 1992.

[11] L. Hong, "Automatic Personal Identification Using Fingerprints", Ph.D. Thesis, 1998.

[12] L. Lam, S. W. Lee, and C. Y. Suen, "Thinning Methodologies-A Comprehensive Survey", IEEE Transactions on Pattern analysis and machine intelligence, vol. 14, no. 9, 1992.

[13] Manvjeet Kaur, Mukhwinder Singh, Akshay Girdhar, and Parvinder S. Sandhu, "Fingerprint Verification System using Minutiae Extraction Technique, World Academy of Science, Engineering and Technology 46 2008".

## Author Profile



**Komal Sharma** received the B.Tech degree in Computer Engineering from Punjab technical University Jalandhar in 2012. Currently, She is pursuing M.Tech degree in Computer Science from Guru Kashi University, Talwandi Sabo, Bathinda (Punjab). Her research interests include Image processing and Data mining.