

Image Analysis for the Enhancement of the Noisy Fingerprint Image

Avinash Pokhriyal, Praveen Sengar

Faculty of Management & Computer Applications, R.B.S. College, Khandari, Agra

Shachipati Pandey

Faculty of Engineering & Technology, R.B.S. College, Bichpuri, Agra

(Received September 19, 2007)

1. Introduction

Each person has his own fingerprints with the permanent uniqueness. So fingerprints have been used for identification and forensic investigation for a long time. A fingerprint is composed of many ridges and furrows (Fig1.1). These ridges and furrows present good similarities in each small local window, like parallelism and average width. However, shown by intensive research on fingerprint recognition, fingerprints are not distinguished by their ridges and furrows, but by Minutia, which are some abnormal points on the ridges (Figure 1.2). Among the variety of minutia types reported in literatures, two are mostly significant and in heavy usage: one is called termination (or ending), which is the immediate ending of a ridge; the other is called bifurcation (or branch), which is the point on the ridge from which two branches derive.



Fig 1.1: A fingerprint

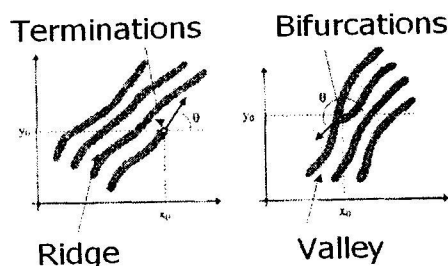


Fig 1.2 Minutia (terminations & bifurcations).

The fingerprint recognition problem can be grouped into 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 programme-based.

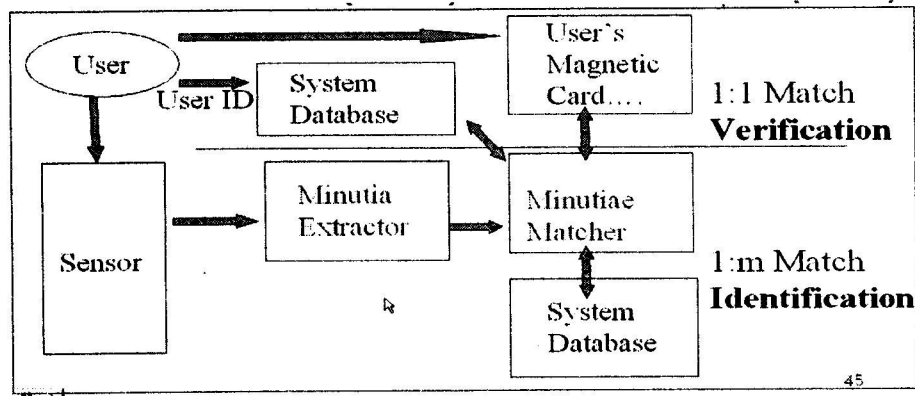


Fig 1.3 Verification vs. Identification

Fingerprint verification is to verify the authenticity of one person by his fingerprint. The user provides his fingerprint together with his identity information like his ID number. The fingerprint verification system retrieves the fingerprint template according to the ID number and matches the template with the real-time acquired fingerprint from the user. Usually it is the underlying design principle of AFAS (Automatic Fingerprint Authentication System).

Fingerprint identification is to specify one person's identity by his fingerprint(s). Without knowledge of the person's identity, the fingerprint identification system tries to match his fingerprint(s) with those in the whole fingerprint database. It is especially useful for criminal investigation cases. And it is the design principle of AFIS (Automatic Fingerprint Identification System).

However, all fingerprint recognition problems, either verification or identification, are ultimately based on a well-defined representation of a fingerprint. As long as the representation of fingerprints remains the uniqueness and keeps simple, the fingerprint matching, either for the 1-to-1 verification cases or 1-to-m identification case, is straightforward and easy.

2. Fingerprint Image Enhancement

Fingerprint Image enhancement 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.

Histogram Equalization

Histogram equalization is to expand the pixel value distribution of an image so as to increase the perceptual information. The original histogram of a fingerprint image has the bimodal type [Figure 2.1], the histogram after the histogram equalization occupies all the range from 0 to 255 and the visualization effect is enhanced [Figure 2.2].

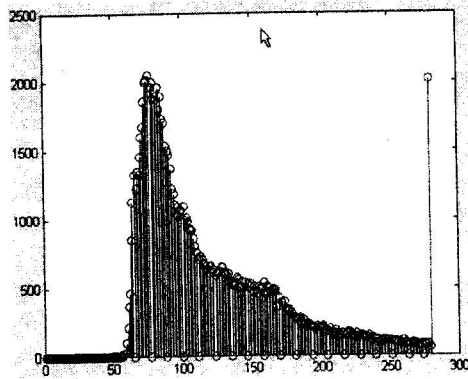


Fig 2.1 the Original Histogram

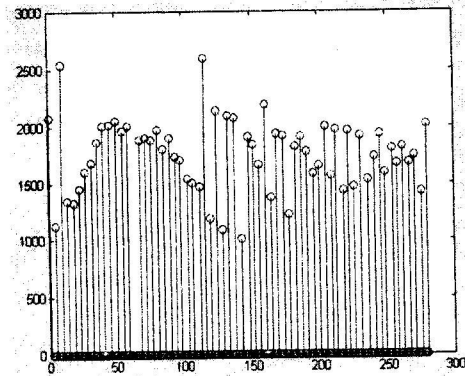


Fig 2.2 after the Histogram Equalization

The right side of the following figure is the output after the histogram equalization.

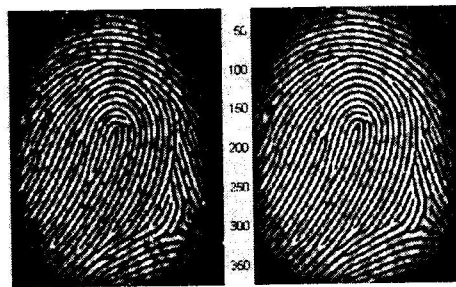


Figure 2.3 Histogram Enhancement.
Original Image (Left). Enhanced image (Right)

Fingerprint Image Binarization

Fingerprint Image Binarization is to transform a gray-scale image to a black-and-white image. In an 8-bit Gray-scale image, a pixel can take on 256 different intensity values while in a binary image, each pixel is assigned to be either black or white. So, binarization converts an 8-bit gray-scale image a 1-bit image with 0-value for ridges and 1-value for furrows. This conversion is performed by applying a threshold value to the image.

In MATLAB, a pixel value of 1 means the pixel is white, and a value of 0 indicates that the pixel is black. For an 8-bit gray-scale image, the pixels are decimal values between 0 and 1. so, when a threshold is applied to the image, all pixel values are compared to the input threshold. All the pixel values below the threshold are set to 0, while all the pixel values above threshold are set to 1. by the end of this process, all the pixel values within the image are either 0 or 1, and the image has been converted to the binary format with ridges as black and furrows as white. If $I(i, j)$ is the old gray-scale image, then the new binary image $I_1(i, j)$ is computed as:

$$I_1(i, j) = \begin{cases} 1, & \text{if } I(i, j) \geq \text{local mean} \\ 0, & \text{otherwise.} \end{cases}$$

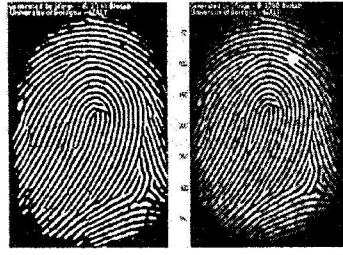


Fig3.2.1 Fingerprint image after binarization
Binarized image(left), Gray-scale image(right)

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 minutia in the bound region 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. The first step is orientation estimation¹, while the second is intrigued from some Morphological methods.

Step-I: We estimate the block direction for each block of the fingerprint image with $W \times W$ in size (W is 16 pixels by default). Then we calculate the gradient values along x-direction (g_x) and y-direction (g_y) for each pixel of the block. We choose two Sobel filters to fulfill this task because sobel filters are a good choice when it comes to noise suppression. The horizontal sobel operator is used to compute g_x

$$\begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix}$$

The vertical sobel operator is used to compute g_y

$$\begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}$$

The direction map is shown in the following diagram. We assume there is only one fingerprint in each image.

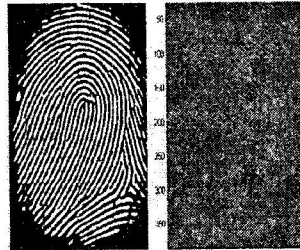


Fig 2.3.1 Direction map.
Binarized fingerprint (left), Direction map (right)

Step-II: Here we extract ROI using two Morphological operations called 'OPEN' and 'CLOSE'. The 'OPEN' operation can expand images and remove peaks introduced by background noise [Fig 2.3.2]. The 'CLOSE' operation can shrink images and eliminate small cavities [Figure 2.3.3].

Figure 2.3.4 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.

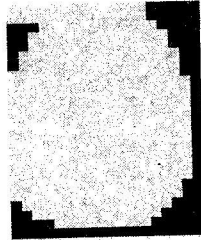


Fig 2.3.1 Original Image Area

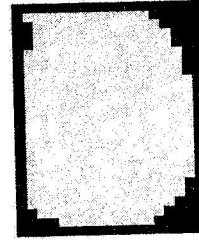


Fig 2.3.3 After CLOSE operation

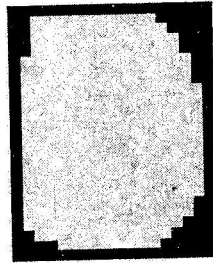


Fig 2.3.2 After OPEN operation

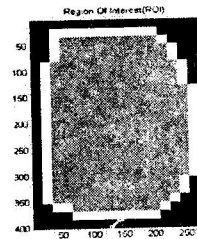


Fig 2.3.4 ROI + Bound

3. Fingerprint Ridge Thinning

Ridge Thinning is to eliminate the redundant pixels of ridges till the ridges are just one pixel wide. Website² uses an iterative, parallel thinning algorithm. In each scan of the full fingerprint image, the algorithm marks down redundant pixels in each small image window (3x3). And finally removes all those marked pixels after several scans. In our testing, such an iterative, parallel thinning algorithm has bad efficiency although it can get an ideal thinned ridge map after enough scans. One-in-all method³ to extract thinned ridges from gray-level fingerprint images directly. Their method traces along the ridges having maximum gray intensity value. However, binarization is implicitly enforced since only pixels with maximum gray intensity value are remained. Also in our testing, the advancement of each trace step still has large computation complexity although it does not require the movement of pixel by pixel as in other thinning algorithms. Thus the third method is bid out which uses the built-in Morphological thinning function in MATLAB, and so we use MATLAB's **bwmorph** thinning function alongwith 'thin' operator.

The thinned ridge map is then filtered by other three Morphological operations to remove some H breaks, isolated points and spikes.

4. Minutia Detection

After the fingerprint ridge thinning, detecting minutia points is an easy task. We analyze 3x3 neighbourhood of a pixel. Let (x,y) denotes a pixel in the thinned ridge and it has eight neighbours, then

if $\sum_{i=0}^7 N_i = 1$, then pixel (x,y) is minutiae type ridge-ending (termination).

if $\sum_{i=0}^7 N_i = 3$, then pixel (x,y) is minutiae type bifurcation (branch).

if $\sum_{i=0}^7 N_i = 0$, then pixel (x,y) is isolated and should be erased.

if $\sum_{i=0}^7 N_i = 2$, then pixel (x,y) is on the ridge.

In general, for each 3x3 window, if the central pixel is 1 and has exactly 3 one-value neighbors, then the central pixel is a ridge branch [Fig 4.1]. If the central pixel is 1 and has only 1 one-value neighbor, then the central pixel is a ridge ending [Fig 4.2].

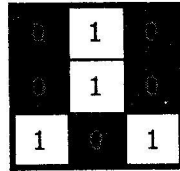


Fig 4.1 Bifurcation

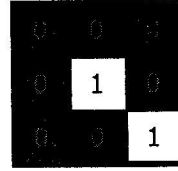
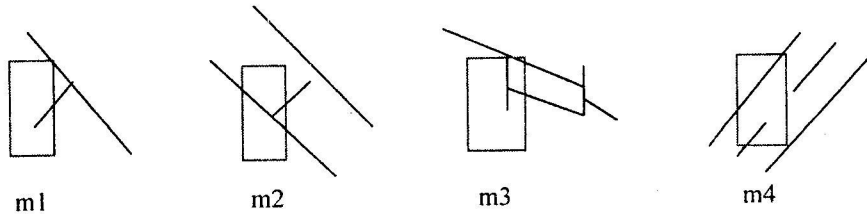


Fig 4.2 Termination

5. False Minutia Removal

The preprocessing stage does not totally heal the fingerprint image. For example, false ridge breaks due to insufficient amount of ink and ridge cross-connections due to over inking are not totally eliminated. Actually all the earlier stages themselves occasionally introduce some artifacts which later lead to spurious minutia. These false minutias will significantly affect the accuracy of matching if they are simply regarded as genuine minutia. So some mechanisms of removing false minutia are essential to keep the fingerprint verification system effective.

Seven types of false minutia are specified in following diagrams:



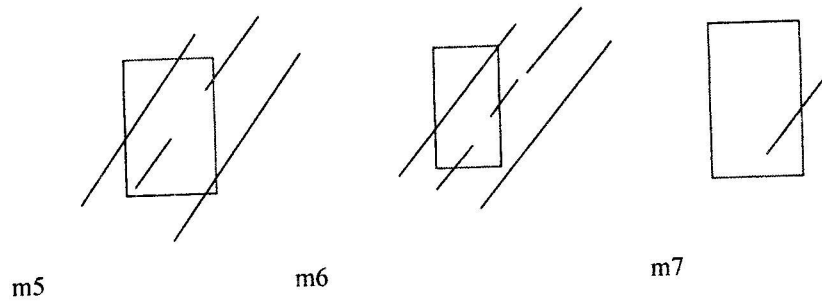


Figure 5.1. False Minutia Structures

m1 is a spike piercing into a valley. In the m2 case a spike falsely connects two ridges. m3 has two near bifurcations located in the same ridge. The two ridge broken points in the m4 case have nearly the same orientation and a short distance. m5 is alike the m4 case with the exception that one part of the broken ridge is so short that another termination is generated. m6 extends the m4 case but with the extra property that a third ridge is found in the middle of the two parts of the broken ridge. m7 has only one short ridge found in the threshold window.

N. Ratha⁴ only handles the case m1, m4, m5 and m6. Thomas Yeo⁵ and D. Maio³ have not false minutia removal by simply assuming the image quality is fairly good. Website² has not a systematic healing method to remove those spurious minutia although it lists all types of false minutia shown in Figure 5.1.1 except the m3 case.

Our procedures in removing false minutia are:

1. If the distance between one bifurcation and one termination is less than D and the two minutia are in the same ridge (m1 case). Remove both of them. Where D is the average inter-ridge width representing the average distance between two parallel neighboring ridges.
2. If the distance between two bifurcations is less than D and they are in the same ridge, remove the two bifurcations. (m2, m3 cases).
3. 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 m4, m5, m6).
4. If two terminations are located in a short ridge with length less than D , remove the two terminations (m7).

Our proposed procedures in removing false minutia have two advantages. One is that the ridge ID is used to distinguish minutia and the seven types of false minutia are strictly defined comparing with those loosely defined by other methods. The second advantage is that the order of removal procedures is well considered to reduce the computation complexity. It surpasses the way adopted by Website² that does not utilize the relations among the false minutia types. For example, the procedure3 solves the m4, m5 and m6 cases in a single check routine. And after procedure 3, the number of false minutia satisfying the m7 case is significantly reduced.

6. Experimentation Results

Evaluation indexes for fingerprint recognition

Two indexes are well accepted to determine the performance of a fingerprint recognition system: one is FRR (false rejection rate) and the other is FAR (false acceptance rate). For an image database, each sample is matched against the remaining samples of the same finger to compute the False Rejection Rate. If the matching g against h is performed, the symmetric one (i.e., h against g) is not executed to avoid correlation. All the scores for such matches are composed into a series of Correct Score. 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. If the matching g against h is performed, the symmetric one (i.e., h against g) is not executed to avoid correlation. All the scores from such matches are composed into a series of Incorrect Score.

Experimentation Results

A fingerprint database from the FVC2000 (Fingerprint Verification Competition 2000) is used to test the experiment performance. The MATLAB program made by us tests all the images without any fine-tuning for the database. The experiments show that our program can differentiate imposturous minutia pairs from genuine minutia pairs in a certain confidence level. Furthermore, good experiment designs can surely improve the accuracy as declared⁶. Here is the diagram for Correct Score and Incorrect Score distribution:

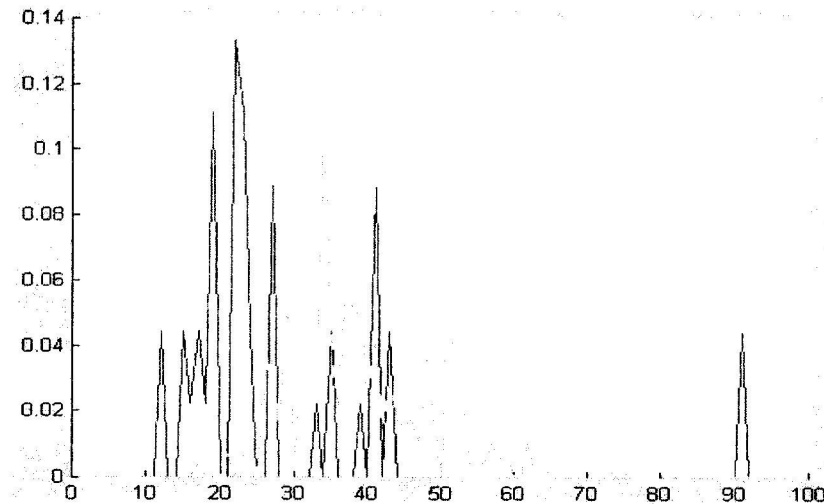


Fig 6.2.1 Distribution of Correct and Incorrect Scores
Dark line: Incorrect Score, Light line: Correct Scores

It can be seen from the graph that there exist two partially overlapped distributions. The Red curve whose peaks are mainly located at the left part means the average incorrect match score is 25. The green curve whose peaks are mainly located on the right side of red curve means the average correct match score is 35. This indicates the algorithm is capable of differentiate fingerprints at a good correct rate by setting an appropriate threshold value.

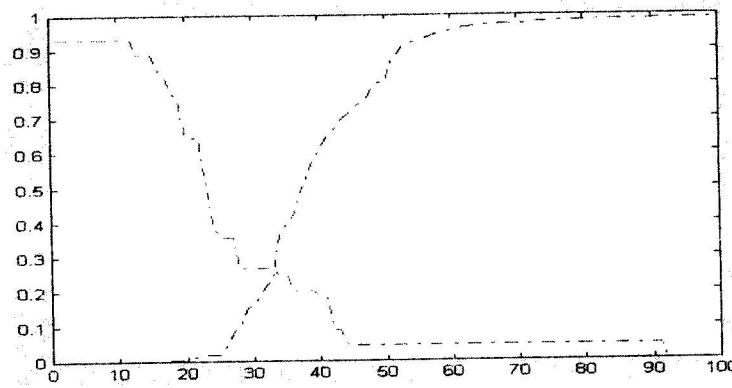


Fig 6.2.2 FAR and FRR curve

Left dot line: FRR curve

Right dot line: FAR curve

The above diagram shows the FRR and FAR curves. At the equal error rate 25%, the separating score 33 will falsely reject 25% genuine minutia pairs and falsely accept 25% imposturous minutia pairs and has 75% verification rate.

7. Conclusion

Our paper has combined many methods to extract good minutia points from even a noisy fingerprint image. The combination of multiple methods comes from a wide investigation into research paper. Also some novel changes like segmentation using Morphological operations, are used in our paper, which are not reported in other literatures I referred to.

Also a program coding with MATLAB going through all the stages of the fingerprint recognition is built. It is helpful to understand the procedures of fingerprint recognition. And demonstrate the key issues of fingerprint recognition.

References

1. Lin Hong, *Automatic Personal Identification Using Fingerprints*, Ph.D. Thesis, 1998.
2. FVC2002. <http://bias.csr.unibo.it/fvc2002/>
3. D. Maio and D. Maltoni. Direct gray-scale minutiae detection in fingerprints. *IEEE Trans. Pattern Anal. and Machine Intell.*, **19(1)** (1997) 27-40.
4. N. Ratha, S. Chen and A.K. Jain. Adaptive Flow Orientation Based Feature Extraction in Fingerprint Images, *Pattern Recognition*, **28** (1995) 1657-1672.
5. *Image Systems Engineering Program, Stanford University*. Student project by Thomas Yeo, Wee Peng Tay, Ying Yu Tai.
6. http://ise0.stanford.edu/class/ee368a_proj01/dropbox/project22/finger/
7. A. K. Jain, L. Hong and R. Bolle. On-Line Fingerprint Verification, *IEEE Trans. On Pattern Anal. and Machine Intell.*, **19(4)** (1997) 302-314.

8. Alessandro Farina, Zsolt M. Kovacs-Vajna and Alberto Leone. Fingerprint minutiae extraction from skeletonized binary images, *Pattern Recognition*, **32-4** (1999) 877-889.
9. C. J. Lee and S. D. Wang: Fingerprint feature extraction using Gabor filters, *Electron. Lett.*, **35(4)** (1999) 288-290.
10. M. Tico, P. Kuosmanen and J. Saarinen. Wavelet domain features for fingerprint recognition, *Electron. Lett.*, **37(1)** (2001) 21-22.
11. L. Hong, Y. Wan and A. K. Jain, Fingerprint Image Enhancement: Algorithms and Performance Evaluation, *IEEE Transactions on PAMI*, **20(8)** (1998) 777-789.
12. FVC2000. <http://bias.csr.unibo.it/fvc2000/>
13. L. C. Jain, U. Halici, I. Hayashi, S. B. Lee and S. Tsutsui, *Intelligent biometric techniques in fingerprint and face recognition*, 1999, the CRC Press.