Radio Frequency Identification based Electronic Passport

Shital P. Kalyankar[1], S. R.Chougule[2]

[1, 2] Department of Technology, Shivaji University, Kolhapur, Maharashtra, India

Abstract: ***We propose a system in which fingerprint verification is done by using extraction of minutiae technique and the system that automates the whole process of taking attendance [1,4,6],Manually which is a laborious and troublesome work and waste a lot of time, with its managing and maintaining the records for a period of time is also a burdensome task. For this purpose we use fingerprint verification system using extraction of minutiae techniques. The goal of develop a complete system for fingerprint verification through extraction and matching minutiae. To achieve good minutiae extraction in fingerprints with varying quality, preprocessing in form of image enhancement and binarization is first applied on fingerprints before they are evaluated. Many methods have been combined to build a minutia extractor and a minutia matcher. This project highlights the development of fingerprint verification system using MATLAB. Radio-frequency identification (RFID) devices for electronic passports.***

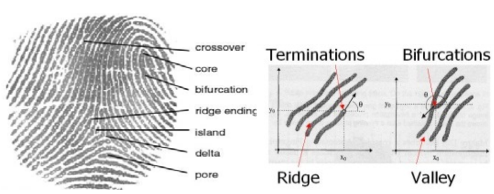
Keywords: RFID identification Biometrics, dynamic programming,

1. Introduction

Matching a finger print is a un resolved. Finger print verification is an important biometric Technique for personal identification. In this paper, we describe the design and implementation of a prototype

1.1 Fingerprint

A fingerprint is composed of many ridges and furrows. 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 1.1.1).Among the variety of minutia types reported in literatures, two are mostly significant and in heavy usage: one is called termination, which is the immediate ending of a ridge; the other is called bifurcation, which is the point on the ridge from which two branches derive.



a) b)

Figure 1.1: (a) The fingerprint image b) Minutiae points

1.2 Fingerprint Recognition

The fingerprint recognition problem can be grouped into two sub-domains: one is fingerprint verification and the other is fingerprint identification [2] (Figure1.2). finger print recognition here is referred as AFRS (Automatic Fingerprint Recognition System), which is program-based.

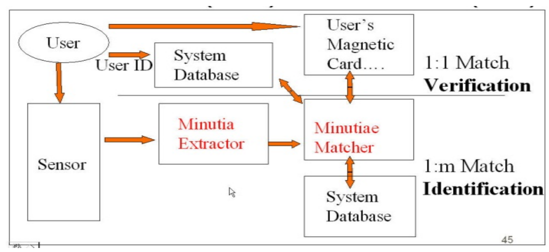


Figure 1.2.: Verification vs. Identification

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).

2. Proposed System

A fingerprint recognition system constitutes of fingerprint acquiring device, minutia extractor and minutia matcher [Figure 2.1.1].

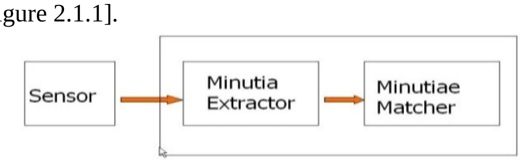


Figure 2: Simplified Fingerprint Recognition System

For fingerprint acquisition, optical or semi-conduct sensors are widely used. 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.

2.1 Algorithm Level Design

To implement a minutia extractor, a three-stage approach is widely used by researchers. They are preprocessing, minutia extraction and post processing stage [Figure 2.2.1].

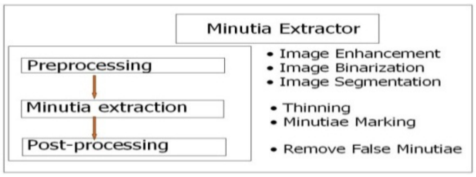


Figure 2.1: Minutia Extractor

a) **Preprocessing**
For the fingerprint image preprocessing stage, I use Histogram Equalization and Fourier Transform to do image enhancement [9]. And then the fingerprint image is binarized using the locally adaptive threshold method. 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. Also the morphological operations for extraction ROI are introduced to fingerprint image segmentation by me.

b) **Minutia Extraction**
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. The minutia marking is a simple task as most literatures reported but one special case is found during my implementation and an additional check mechanism is enforced to avoid such kind of oversight.

c) **Post processing**
For the post processing stage, a more rigorous algorithm is developed to remove false minutia based on. Also a novel representation for bifurcations is proposed to unify terminations and bifurcations.

Figure2.2: Minutia Matcher

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 [Figure 2.2.2].

3. Finger Print Image Preprocessing

3.1 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. Two Methods are adopted in my fingerprint recognition system: the first one is Histogram Equalization; the next one is Fourier Transform.

3.2 Histogram Equalization

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

3.3 Fingerprint Enhancement by Fourier Transform

We divide the image into small processing blocks [3,5](32 by 32 pixels) and perform the Fourier transform according to:

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

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)|.

Gets the enhanced block according to2)

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

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

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

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

The k in formula 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. The enhanced image after FFT has the improvements to connect some falsely broken points on ridges and to remove some spurious connections between ridges.

3.4 Fingerprint Image Binarization

Fingerprint Image Binarization is to transform the 8-bit Gray fingerprint 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. Such a named method comes from the mechanism of transforming a pixel value to 1 if the value is larger than the mean intensity value of the current block (16x16) to which the pixel belongs

3.5 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 is confusing with that spurious minutia that is generated when the ridges are out of the sensor.

To extract the ROI, a two-step method is used. The first step is block direction estimation and direction variety check [7,10], while the second is intrigued from some Morphological methods.

3.6 Block direction estimation

I. Estimate the block direction for each block of the fingerprint image with WxW in size (W is 16 pixels by default).
II. The algorithm is:

Calculate the gradient values along x-direction (gx) and y-direction (gy) 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.

tg2ß = 2 ∑∑ (gx*gy)/∑∑ (gx2-gy2) for all the pixels in each block.

The tangent value of the block direction is estimated nearly the same as the way illustrated by the following formula.
tg2θ = 2sinθ cosθ /(cos2θ -sin2θ)

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 ∑∑ (gx*gy)+ ∑∑ (gx2-gy2)}/ W*W*∑∑ (gx2+gy2)

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

3.7 ROI extraction by Morphological operations

Two Morphological operations called 'OPEN' and 'CLOSE' are adopted. The 'OPEN' operation can expand images and remove peaks introduced by background noise The 'CLOSE' operation can shrink images and eliminate small cavities.

3.8 Fingerprint Image Binarization

Fingerprint Image Binarization is to transform the 8-bit Gray fingerprint 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.

4. Minutia Extraction

4.1 Fingerprint Ridge Thinning

Ridge Thinning is to eliminate the redundant pixels of ridges till the ridges are just one pixel wide. [12] 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 my testing, such an iterative, parallel thinning algorithm has bad efficiency although it can get an ideal thinned ridge map after enough scans. [2] uses a one-in-all method to extract thinned ridges from gray-level fingerprint images directly. Also in my 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.

4.2 Minutia Marking

After the fingerprint ridge thinning, marking minutia points is relatively easy. But it is still not a trivial task as most literatures declared because at least one special case evokes my caution during the minutia marking stage.

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[11] [Figure 4.2.1]. If the central pixel is 1 and has only 1 one-value neighbor, then the central pixel is a ridge ending [Figure b)].

Figure c) illustrates a special case that 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.

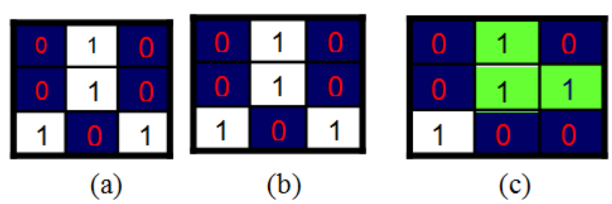


Figure 3: a)Bifurication b)Termination c)Trippal counting branch

Also the average inter-ridge width D is estimated at this stage. The average inter-ridge width refers to the average distance between two neighboring ridges. The way to approximate the D value is simple. Scan a row of the thinned ridge image and sum up all pixels in the row whose value is one. Then divide the row length with the above summation to get an inter-ridge width.

5. Minutia Post Processing

5.1 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 [8]. Actually all the earlier stages themselves occasionally introduce some artifacts which later lead to spurious minutia. This false minutia 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:

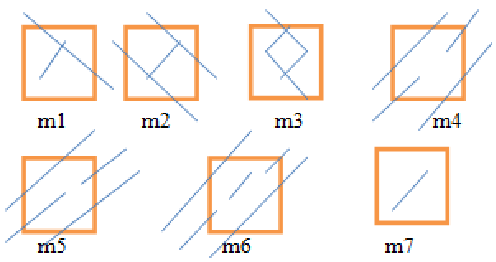


Figure 4: Seven types of false minutia

a. 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.[4] only handles the case m1, m4,m5 and m6. [9] and [2] have not false minutia removal by simply assuming the image quality is fairly good. [12] 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.

5.2 Unify terminations and bifurcations

Since various data acquisition conditions such as impression pressure can easily change one type of minutia into the other, most researchers adopt the unification representation for both termination and bifurcation. So each minutia is completely characterized by the following parameters at last: 1) x-coordinate, 2) y-coordinate, and 3) orientation.

The orientation calculation for a bifurcation needs to be specially considered. All three ridges deriving from the bifurcation point have their own direction, represents the bifurcation orientation using a technique proposed in [13]. [1] Simply chooses the minimum angle among the three anticlockwise orientations starting from the x-axis. Both methods cast the other two directions away, so some information loses. Here I propose a novel representation to break a bifurcation into three terminations. The three new terminations are the three neighbor pixels of the bifurcation and each of the three ridges connected to the bifurcation before is now associated with a termination respectively [Figure 5.2.1].

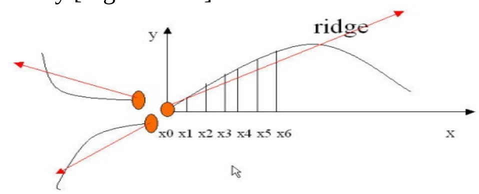


Figure 5.2.1: A bifurcation to three terminations

Three neighbors become terminations (Left)Each termination has their own orientation (Right)And the orientation of each termination (tx,ty) is estimated by following method [book ch2]: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: atan((sy-ty)/(sx-tx)).

6. Minutia Match

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 alignment-based match algorithm partially derived from the [1] is used in my project. It includes two consecutive stages: one is alignment stage and the second is match stage.

6.1 Alignment Stage

Given two fingerprint images to be matched, choose any one minutia from each image; calculate the similarity of the two ridges associated with the two referenced minutia points[7]. If the similarity is larger than a threshold, transform each set of minutia to a new coordination system whose origin is at the referenced point and whose x-axis is coincident with the direction of the referenced point

The ridge associated with each minutia is represented as a series of x-coordinates (x1, x2…xn) 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*L.

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

$S = \sum_{i=0}^{m} x_i X_i / [\sum_{i=0}^{m} x_i^2 X_i^2]^{0.5}$,

1. Where (xi~xn) and (Xi~XN) are the set of minutia for each fingerprint image respectively. 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. For each fingerprint, translate and rotate all other minutia with respect to the reference minutia according to the following formula:

$$\begin{pmatrix} xi\_new \\ yi\_new \\ \theta i\_new \end{pmatrix} = TM * \begin{bmatrix} (xi - x) \\ (yi - y) \\ (\theta i - \theta) \end{bmatrix}$$

Where (x,y,θ) is the parameters of the reference minutia, and TM is

$$TM = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

The following diagram illustrates the effect of translation and rotation:

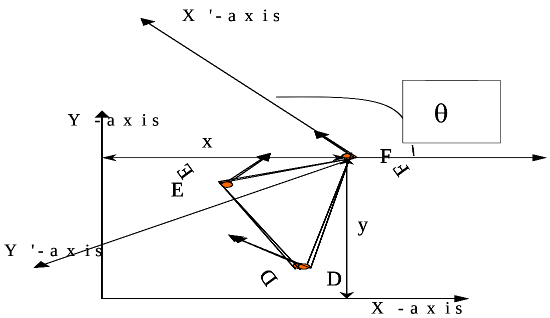


Figure 5: Effect of Translation and Rotation

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

My method to align two fingerprints is almost the same with the one used by [1] but is different at step 2. Lin's method uses the rotation angle calculated from all the sparsely sampled ridge points. My method use the rotation angle calculated earlier by densely tracing a short ridge start from the minutia with length D. Since I have already got the minutia direction at the minutia extraction stage, obviously my method reduces the redundant calculation but still holds the accuracy.

My approach is to transform each according to its own reference minutia and then do match in a unified x-y coordinate. Therefore, less computation workload is achieved through my method.

6.2 Match Stage

After we get two set of transformed minutia points, we use the elastic match algorithm to count the matched minutia pairs by assuming two minutia having nearly the same position and direction are identical. The matching algorithm for the aligned minutia patterns needs to be elastic since the strict match requiring that all parameters (x, y, θ) are the same for two identical minutia is impossible due to the slight deformations and inexact quantization's of minutia.

My approach to elastically match minutia is achieved by placing a bounding box around each template minutia. If the minutia to be matched is within the rectangle box and the direction discrepancy between them is very small, then the two minutia are regarded as a matched minutia pair. Each minutia in the template image either has no matched minutia or hasonly one corresponding minutia.

The final match ratio for two fingerprints is the number of total matched pair over the number of minutia of the template fingerprint. The score is 100*ratio and ranges from 0 to 100. If the score is larger than a pre-specified threshold, the two fingerprints are from the same finger.

7. Result & Discussion

A fingerprint database from the FVC2000 (Fingerprint Verification Competition 2000) is used to test the experiment performance. My program tests all the images without any fine-tuning for the database. The experiments show my 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 declared by [10]. Further studies on good designs of training and testing are expected to improve the result.

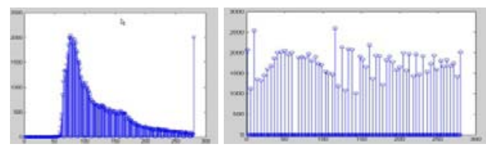


Figure 6: a) The Original Histogram b) After Equalization of a Fingerprint Image

Figure 7: c) Histogram Enhanced By FFT d) Fingerprint Enhancement

Figure 8:e)After adaptive binarization f) Direction map.

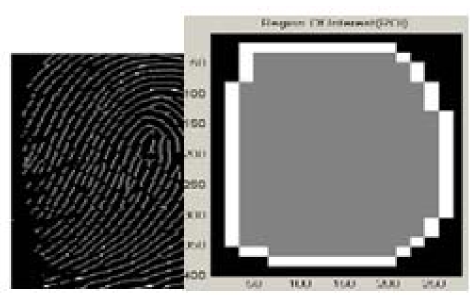


Figure 9: g) ROI+ Boundh h) Image after removal of H breaks

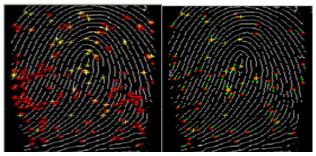


Figure 10: i) Extracted Minutiae j) after removal of spurious Minutiae

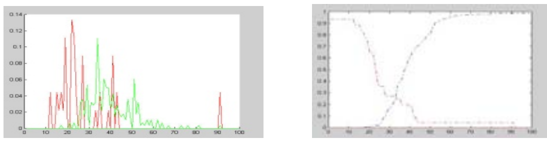


Figure 11: k) Distribution scores l) FAR & FRR Scores

It can be seen from the figure K) 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. The fig l) 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.

8. Conclusion

My project has combined many methods to build a minutia extractor and a minutia matcher. The combination of multiple methods comes from a wide investigation intoresearch paper. Also some novel changes like segmentation using Morphological operations, minutia marking with special considering the triple branch counting, minutia unification by decomposing a branch into three terminations, and matching in the unified x-y coordinate system after a two-step transformation are used in my project, 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] J.Hollingum,"Automated Finferprint Anaysis offers Fast Verification".Sensor Review, Vol 12, No.3,p.p 12-15,1992

[2] “Guide to Fingerprint Recognition” DigitalPersona, Inc. 720 Bay RoadRedwood City, CA 94063 USA,http://www.digitalpersona.com

[3] L. O’Gorman, “Overview of fingerprint verification technologies,”Elsevier Information Security Technical Report, Vol. 3, No. 1, 1998.

[4] Anil K. Jain, Arun Ross and Salil Prabhakar, “An introduction tobiometric recognition,” Circuits and Systems for Video Technology,

[5] IEEE Transactions on Volume 14, Issue 1, Jan. 2004 Page(s):4 – 20.

[6] Henry Selvaraj, S. Arivazhagan and L. Ganesan, “FingerprintVerification Using Wavelet Transform,” International Conference on

[7] Computational Intelligence and Multimedia Applications, 2003

[8] Henry Selvaraj, S. Arivazhagan and L. Ganesan, “FingerprintVerification Using Wavelet Transform,” International Conference on

[9] Computational Intelligence and Multimedia Applications, 2003.

[10] Lin Hong., "Automatic Personal Identification Using Fingerprints,"Ph.D. Thesis, ISBN: 0-599-07596-1, 1998

[11] U. Halici, L. C. Jain, A. Erol, "Introduction to Fingerprint Recognition,"Intelligent Biometric Techniques in Fingerprint and Face Recognition,L.C. Jain, U. Halici, I.Hayashi, S.B. Lee, S. Tsutsui (editors), CRCPress, 1999

[12] L. O’Gorman, “Overview of fingerprint verification technologies,”Elsevier Information Security Technical Report, Vol. 3, No. 1, 1998.

[13] Chaur-Chin Chen and Yaw-Yi Wang, “An AFIS Using FingerprintClassification,” Image and Vision Computing, 2003.

[14] Sandhu, “Fingerprint Verification System using Minutiae ExtractionTechnique,” World Academy of Science, Engineering and Technology46 2008

[15] E. Hastings, “A Survey of Thinning Methodologies,” Pattern analysisand Machine Intelligence, IEEE Transactions, vol. 4, Issue 9, pp. 869885, 1992.

[16] Ying Jie,YuanYi fang, Zhang Renjie, Song Qifa. ”Fingerprint minutiaematching algorithm for real time system.” Pattern Recognition 39 (2006)143 – 146.