Fingerprint Matching Using Delaunay Triangle

A Dissertation submitted to the University of Hyderabad in partial fulfillment of the degree of

MASTER OF TECHNOLOGY

in

Information Technology

by

Elham Mashhadi

Department of Computer and Information Sciences

School of Mathematics, Computer and Information Sciences

University of Hyderabad

(P.O.) Central University, Gachibowli

Hyderabad – 500 046

Andhra Pradesh

India

CERTIFICATE

This is to certify that the dissertation entitled "**Fingerprint matching using Delaunay Triangle"** submitted by **Elham Mashhadi** bearing Reg. No 09MCMB34 in partial fulfillment of the requirements for the award of Master of Technology in Information Technology is a bonafide work carried out by him/her under my/our supervision and guidance.

The dissertation has not been submitted previously in part or in full to this or any other University or Institution for the award of any degree or diploma.

Associate Professor, IDRBT.
Signature of the Supervisor

Head of the Department/Centre

Dean of the School

DECLARATION

I **Elham Mashhadi** hereby declare that this Dissertation entitled "**Fingerprint matching using Delaunay Triangle"** submitted by me under the guidance and supervision of **Dr. Mahil Carr, Associate Professor, IDRBT**, is a bonafide work. I also declare that it has not been submitted previously in part or in full to this University or other University or Institution for the award of any degree or diploma.

Date: Name: **Elham Mashhadi**

Signature of the Student:

Regd. No.**09MCMB34**

Acknowledgement

I would like to express my great gratitude towards my supervisor, Dr. Mahil Carr Associate Professor, IDRBT, Hyderabad, whose encouragement; guidance and support from the initial to the final level enabled me to develop an understanding of the subject.

I would also thanks to all of our friends who have patiently extended all sorts of help in this project. I also extend my heartfelt thanks to my family and well wishers.

Elham Mashhadi

09MCMB34

University of Hyderabad and IDRBT

Hyderabad

ABSTRACT

Fingerprint matching is the most important step in fingerprint identification. The purpose of fingerprint matching is to compare two fingerprint images and return a similarity score that represents to the probability of match between the two fingerprints so most of the fingerprint matching algorithms have been developed using minutiae based matching. We extracted minutiae from NBIS software provided by NIST. In this thesis, we introduced a novel matching algorithm based on Delaunay Triangles. First, we extracted the minutiae and next made triangles and compare the similarity between triangles.

My demonstration program is coded by MATLAB. The effectiveness of the proposed algorithm is tested on a public database FVC2002 DB1, DB2 and FAR, FRR is computed.

Contents

Table of Figures

1. INTRODUCTION

1.1. Biometrics

The science and technology of measuring and analyzing biological data is called Biometrics. In information technology, biometrics refers to technologies that measure and analyze human body characteristics, such as eye retinas, fingerprints and irises, voice patterns, facial patterns and hand measurements.

Biometrics is now the new forefront method of security systems. Biometrics aroused to prevent unauthorized access to cellular phones, ATM, laptops, cars and many other security concerned things. Biometric have brought significant changes in security systems making them more secure than ever, efficient, effective and cheap. They have changed the security system from what you remember (such as password) or what you own (such as car keys) to something you physical characteristics (retinal patterns, fingerprints, voice recognition)

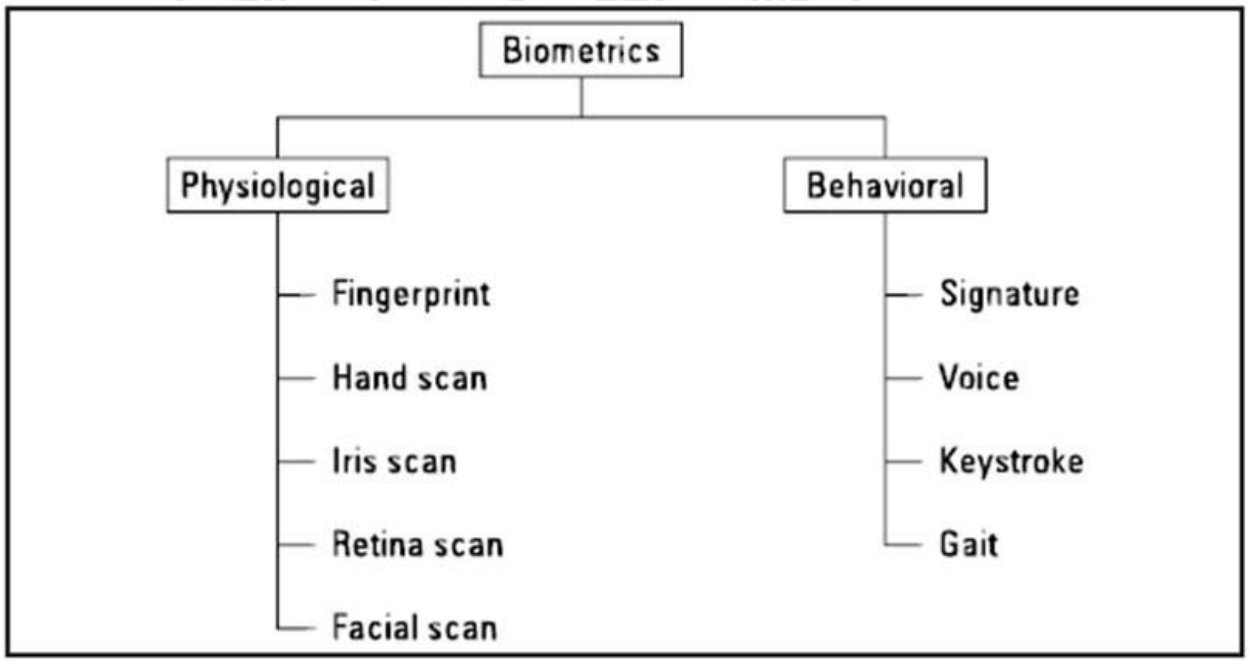


Figure 1.1.1 Two biometric families

Among all the biometric techniques, the oldest method is fingerprint-based identification which has been successfully used in various applications for over a century, and more recently it is automated due to improvements in computing

capabilities. The reason of popularity of Fingerprint identification is because of the inherent ease in acquisition, the various sources (ten fingers) available for collection, and their established use and collections by law enforcement and immigration.

There are two biometric systems: an 'identification' system and a 'verification' (authentication) system, which are defined below.

1.1.1. Authentication/Verification

Authentication (or Verification) is the process of checking a known biometric sample to an already existing reference identified in a database or a personal token to confirm the identity of the claimant. This is called one-*to-one* process (1:1), and it is usually used in identity documents such as corporate badges, National ID cards or e Passports, in order to prove that the carrier of the document is actually the authorized owner of the document.

Authentication comes with this question that biometric system tries to answer: “Is this X?”

1.1.2. Identification

Identification is the process of checking an unidentified biometric sample to multiple references in a database to find out the identity of the owner of this unknown sample. This is called *one-to-many* process (1: N), mostly it is used in criminal sciences (e.g. AFIS -Automated Fingerprint Identification System-). Civilian AFIS can be used in both positive and negative authentication. Positive identification refers to proving the membership of a group (white list, e.g. VIP access to a secure area), and vice versa.

Authentication means proving the non-membership of a group (blacklist, e.g. Pathological gamblers shortlist used in casinos). Identification is often used at enrollment, even for authentication-oriented systems, to find out repeated identities before registering the new user.

Identification comes with this question, the biometric system tries to answer: “Who is X?

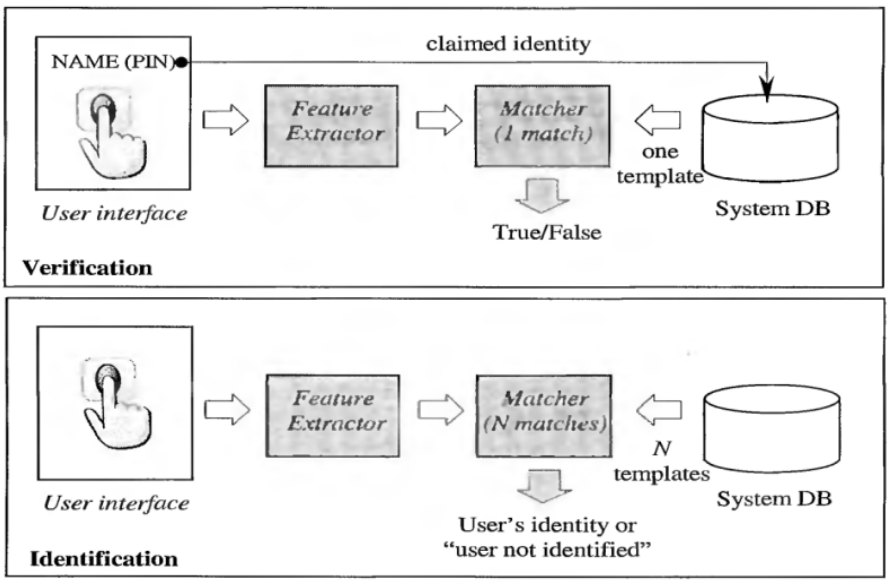


Figure 1.1.2 Verification vs. Identification

1.1.3. Biometric Systems Errors

The authentication process is a comparison between a pre-registered reference image (input) and a newly captured candidate image (template). Depending on the relationship between these two samples, the algorithm will determine the acceptance or rejection of applicants. This statistical process lead to a False Acceptance Rate (FAR, i.e. the probability to accept a non-authorized user) and a False Rejection Rate (FRR, i.e. the probability to reject an authorized user).

The low FAR shows security and low FRR shows user convenience: a system with a very low FAR hence has a high FRR, and it stays perfectly secure until the authorized user himself can't use it!

Researcher has to focus his efforts on FAR or FRR, depending on the application, here are two different examples:

• Where researcher does not want to take the risk that a wrong person gets in, even if an authorized user will need to apply twice or even more, researcher has very secure access to a limited area: this is low FAR.

• Forensic applications where researchers need to identify the wrong person, even if in a first pass, researcher will identify multiple suspects and continue our investigations later on: this is low FRR.

Another metric is EER (Equal Error Rate, point where FAR=FRR), this is interesting to examine different biometric systems, but it is not a good choice of FAR vs. FRR trade-off in the real world since any well-studied application will for sure need a focus on either FAR or FRR. See figure 1.3/4

FAR = FA/N

Where FA = number o f false acceptance, and N = total (*large*) *number o f samples*

FRR = FR/N

Where FR = number o f false rejection, and N = total (*large*) *number o f samples*

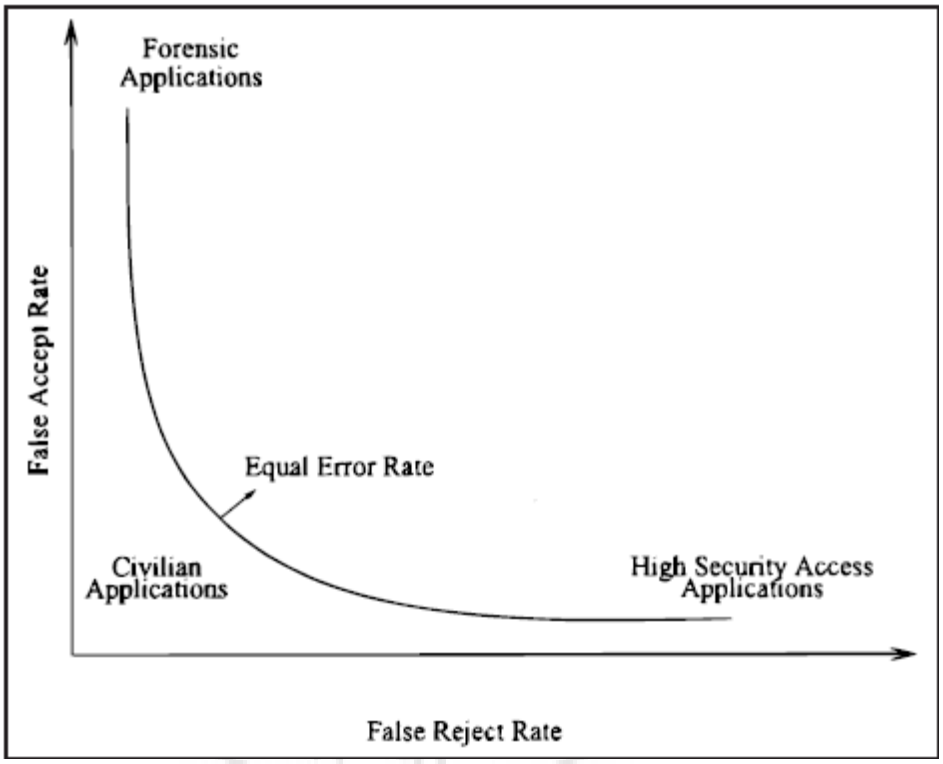


Figure 1.1.3 Error rates1

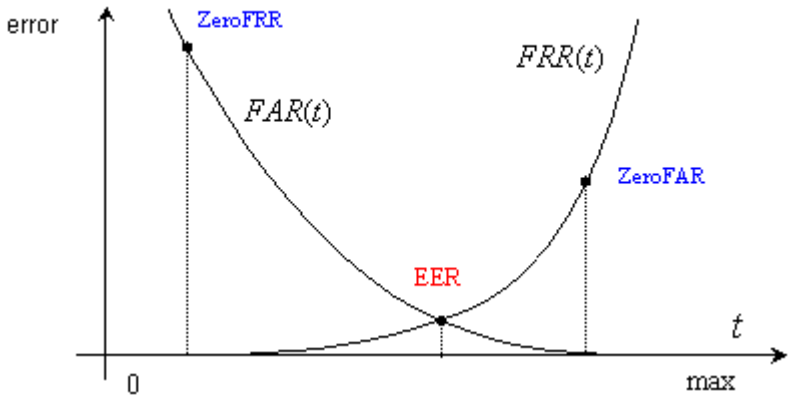


Figure 1.1.4 Error Rates 2

1.2. What is A Fingerprint?

The feature pattern of one finger is called fingerprints (Figure 1.3). Strong evidences had proven that each fingerprint is unique. Each person has his own fingerprints with the uniqueness factors. So fingerprints have being used for identification and forensic investigation for a long time.

Figure 1.1.5 a fingerprint image acquired by an Optical Sensor

1.2.1.Fingerprint features

A fingerprint is made of different parts of ridges and valleys. In a fingerprint image, the ridges appear as dark lines while the valleys are the light areas between the ridges. Underlying ridge structure cannot be effected by a cut or burn of finger, and the original pattern will be recreated by new skin grow. Ridges and valleys generally run next to each other, and their patterns can be analyzed on a global and local level.

1.2.1.1 Global level

Here researcher has different patterns of fingerprints image will have one or more regions where the ridge lines have a distinctive shape.

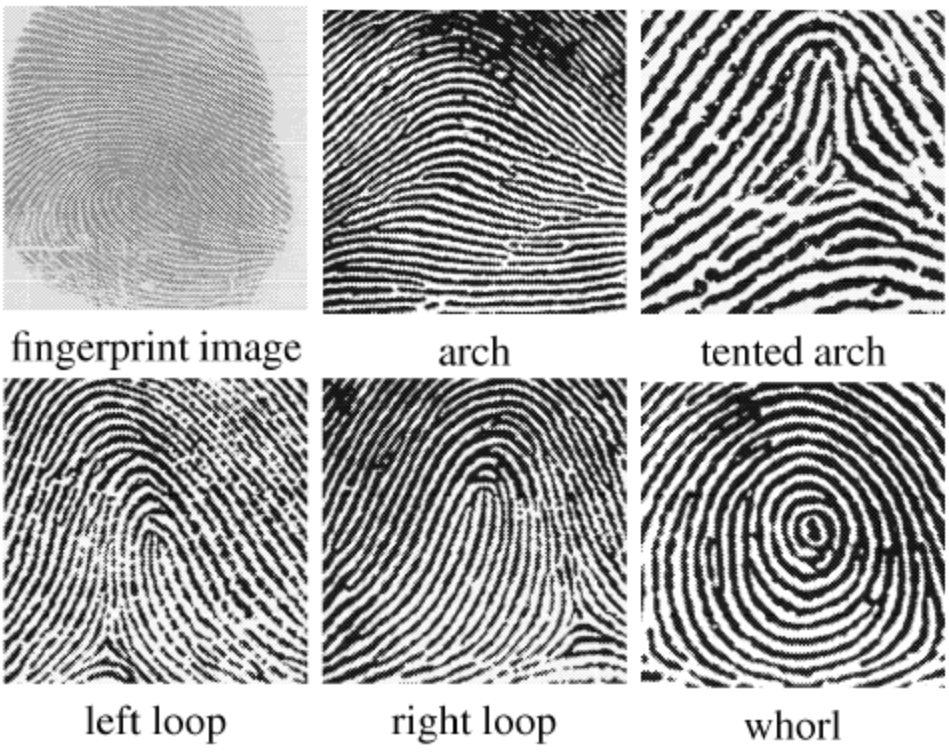


Figure 1.1.6 Global Fingerprint Structures

1.2.1.2 Local level

As the global level allows for a general classification of fingerprints, analyzing the image at the local level provides a significant amount of detail.

Minutiae points are acquired in details and observing the locations where a ridge becomes discontinuous .The project is also based on the minutiae based recognition.

The most common types of minutiae are shown in Figure below

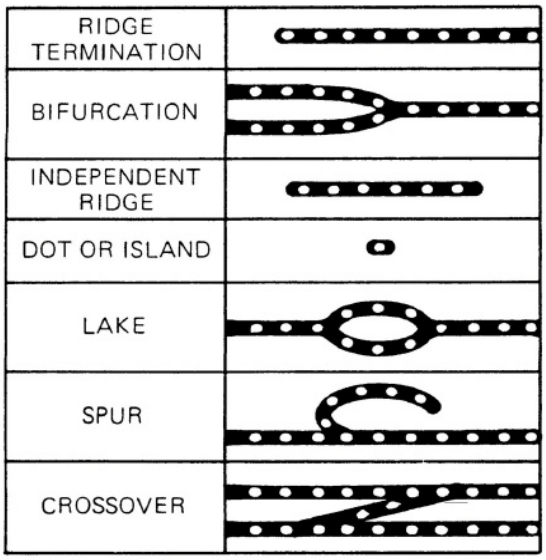


Figure 1.1.7 Local Fingerprint Structures

Among the different types of minutiae reported in literatures, there are two significant and savagely used types of minutiae:

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

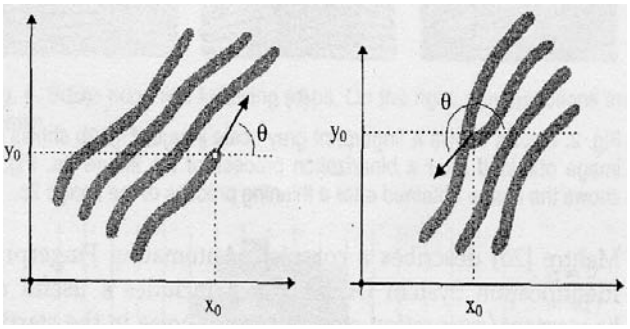


Figure 1.1.8 (a) Ridge Ending, (b) Ridge Bifurcation

(b)

1.3. Fingerprint matching techniques

There are three kind of fingerprint matching as follows:

- **Correlation-based matching**: In this kind two fingerprint images are overlapped and the correlation between corresponding pixels is computed for different adjustments (e.g. various displacements and rotations).
- **Minutiae-based matching**: This technique is the most popular and it is widely used, it is 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 including finding the adjustment between the template and the input minutiae sets that result 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 technique requires that the images be aligned in the same orientation. To do this, the algorithm finds a central point in the fingerprint image and focus 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 performed a minutiae based matching technique. This approach has been intensively studied; also it is the backbone of the current available fingerprint recognition products.

1.4. Algorithm Level Design

A three-stage approach is widely used by researchers, to perform a minutiae extractor. These stages are called preprocessing, minutiae extraction and post processing stage.

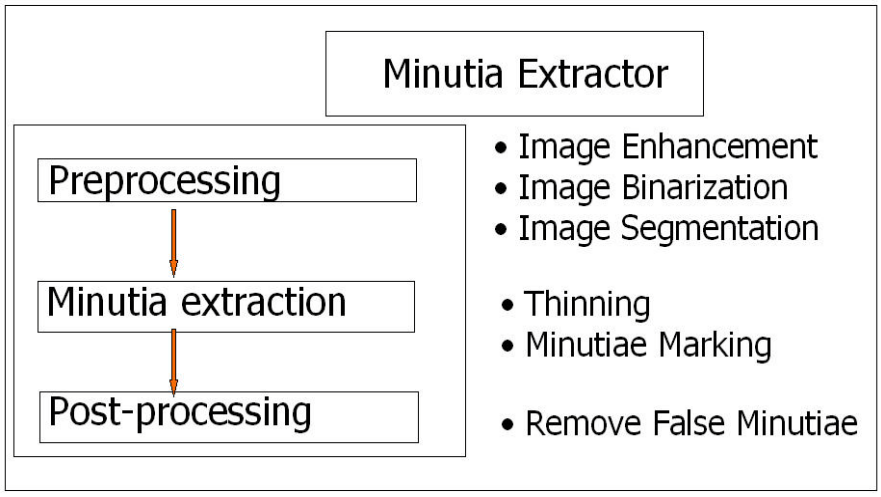


Figure 1.1.9 Minutiae Extractor

1.4.1. Fingerprint Image Preprocessing

1.4.1.1 Fingerprint Image Enhancement

An important characteristic of the ridge structures in a fingerprint image is the information of characteristic features needed for Minutiae extraction. Normally, in a well-defined fingerprint image, the ridges and Valleys should change and flow in locally constant ways. Allowing minutiae to be precisely extracted from the thinned ridges cause facilitating of detection of ridges. However, in practice, because of elements such as noise that corrupt the clarity of the ridge structures, a fingerprint

image may not always be well defined. This corruption may occur because of differences in skin and impressive conditions such as scars, humidity, dirt, and non-uniform contact with the fingerprint capture device. So, image enhancement techniques are often used to reduce the noise and improve the definition of ridges against valleys.

There are two methods in fingerprint recognition system: the first one is Histogram Equalization; the second one is Fourier Transform.

1.4.2. Image Binarization

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

A locally adaptive binarization method is being done to binarize the fingerprint image. In this method image is divided into various 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 1.13).

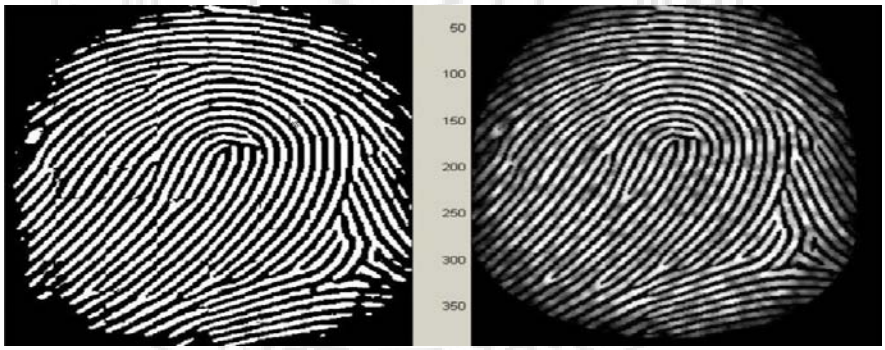


Figure 1.1.10 (a) Binarized Image after FFT, (b) Image before binarization

(a) (b)

1.4.3. Fingerprint Image Segmentation

Before extracting the feature of a fingerprint, it is important to separate the fingerprint regions (presence of ridges) from the background. This limits the area to be processed and therefore increase the processing time and wrong feature extraction. In some cases, a correct segmentation may be very hard, especially in low quality fingerprint image or

noisy images, such as presence of latents. The same information used for quality extraction, such as contrast, ridge orientation and ridge frequency can be used for the segmentation or covering the quantified region quality may be used directly by considering the regions with quality below some threshold as background. Normally, the segmentation is also calculated by block in the same way as the quality extraction. In the above figure the partition of the segmented region overlapped over the original image.

Figure 1.1.11 Image Segmentation

1.5. MINUTIAE EXTRACTION

While researcher has improved the image and segmented the required area, the function of minutiae extraction consists of two operations: Ridge Thinning, Minutiae Marking.

1.5.1. Fingerprint Ridge Thinning

Exact Thinning is to remove the useless pixels of ridges till the ridges are just one pixel wide. Parallel thinning algorithm uses an iterative. In scanning of the full fingerprint image, the algorithm shows useless pixels in each small image window (3x3). And finally after many scans removes all those marked pixels. Such an iterative thinning algorithm although it can get an ideal thinned ridge map after enough scans, but it has bad efficiency. Researcher can use a one-in-all method to directly extract thinned ridges from gray-level fingerprint images. Their method, tracing along the ridges, have maximum gray intensity value. However, binarization is absolutely enforced because only pixels with maximum gray intensity value are stayed. While testing, although it does not require the movement of pixel by pixel as in other thinning algorithms, but the improvement of each trace step still has big computation complexity. Thus the third method is bid out and uses the built-in Morphological thinning function in MATLAB.

1.5.2. Minutiae Marking

After the fingerprint ridge thinning, marking minutiae points is easy. It is still not an easy task as most literatures declared, since at least one special case grab my attention during the minutiae marking stage,.

Generally, 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 [Figure 1.12]. If the central pixel is 1 and has only 1 one-value neighbor, then the central pixel is a ridge ending [Figure1.13].

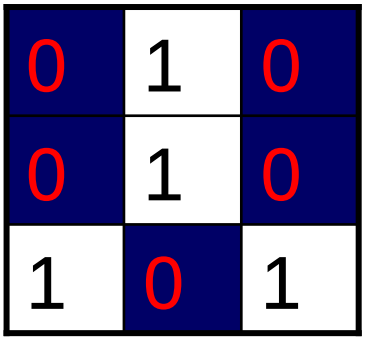


Figure 1.1.12 Bifurcation

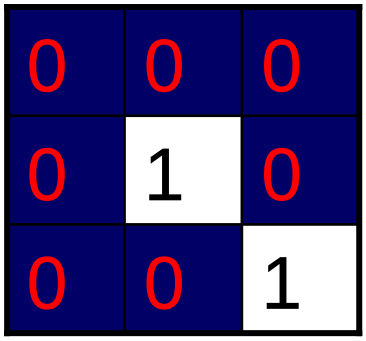


Figure 1.1.13 Termination

. Imagine 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 only one branch is located in the small region. Therefore a check routine requires none of the neighbors of a branch are branches that are added.

Also the average inter-ridge width D is estimated at this stage. The average inter-ridge width means the average distance between two neighboring ridges. The way to estimate the D value is simple. Researcher can 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 sum total to get an inter-ridge width. To do the best, such kind of row scan is performed upon several other rows and column scans are also conducted, finally all the inter-ridge widths are averaged to get the D.

All thinned ridges in the fingerprint image are labeled with a unique ID for further operation together with the minutiae marking. The labeling operation is realized by using the Morphological operation: BWLABEL.

Figure 1.1.14 Enhanced image Thinned image

1.6. MINUTIAE POSTPROCESSING

1.6.1. False Minutiae Removal

The preprocessing level does not completely cure the fingerprint image. For example, false ridge breaks because of inadequate amount of ink and ridge cross-connections due to over inking are not totally removed. In fact all the earlier stages occasionally introduce some artifacts which later lead to false minutiae. This false minutiae if they

are simply regarded as genuine minutiae, will significantly affect the accuracy of matching. Therefore some mechanisms of removing false minutiae are essential to keep the fingerprint verification system effective.

Seven types of false minutiae are specified in following diagrams:

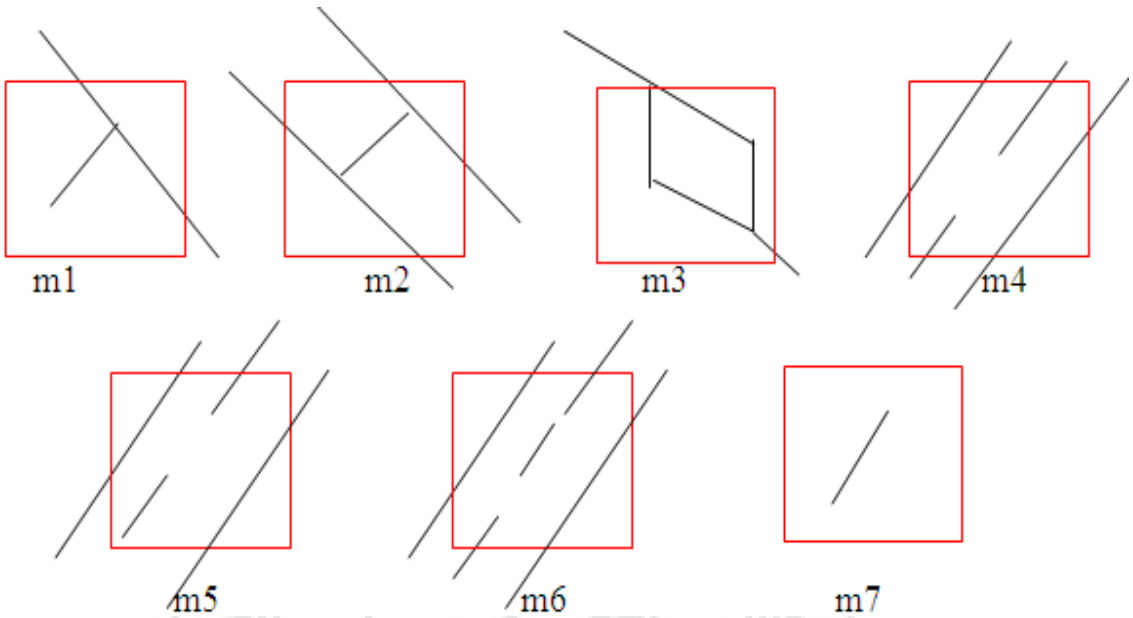


Figure 1.1.15 False Minutiae Structures

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

The procedure for the removal of false minutiae is:

1. Both of them are removed, if the distance between one bifurcation and one termination is less than D and the two minutiae are in the same ridge (m1 case), Where D is the average inter-ridge width represents the average distance between two parallel neighboring ridges.

2. The two bifurcations are removed, if the distance between two bifurcations is less than D and they are in the same ridge, (m2, m3 cases).

3. The two terminations are regarded as false minutiae derived from a broken ridge and are removed, if two terminations are within a distance D and their directions are coincident with a small angle variation and they satisfy the condition that no other termination is located between the two terminations. (Case m4, m5, m6).

4. Researcher removes the two terminations, if two terminations are located in a short ridge with length less than D, (m7).

2.MINUTIAE MATCH.

2.1. Minutiae Representation

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

So each minutiae is completely characterized by the following parameters

1) x-coordinate, 2) y-coordinate, 3) orientation and 4) ridge associated with it

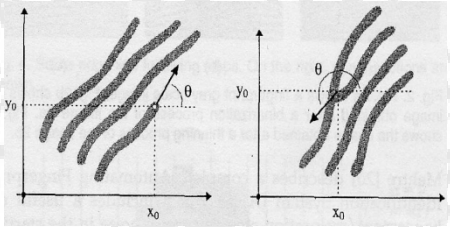


Figure 2.1 ridge ending and bifurcation

In fact a bifurcation can be broken down into 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. Researcher should 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 then divides above sum total with D to get sx. Then get sy using the same way.

Get the direction from:

$$\theta(A) = \begin{cases} 90: \ x(A) = x(Af), y(A) > y(Af) \\ -90: \ x(A) = x(Af), y(A) < y(Af) \\ 0: \ x(A) = x(Af), y(A) = y(Af) \end{cases}$$

$$\theta(A) = \left\{ a\ tan \frac{y(A) - y(At)}{x(A) - x(At)} \right\} \times \frac{180}{\pi} + 180: \ x(A) < x(At)$$

$$\theta(A) = \left\{ a\ tan \frac{y(A) - y(At)}{x(A) - x(At)} \right\} \times \frac{180}{\pi}: \ x(A) > x(At), y(A) >= y(At)$$

$$\theta(A) = \left\{ a\,tan\frac{y(A) - y(At)}{x(A) - x(At)} \right\} \times \frac{180}{\pi} + 360:\ x(A) > x(At), y(A) < y(At)$$

Results after the minutiae extraction stage (Figure 2.2 – 2.4)

Figure 2.2 Thinned image

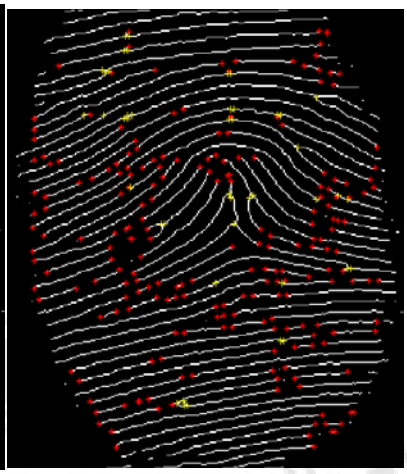


Figure 2.3 Minutiae after marking

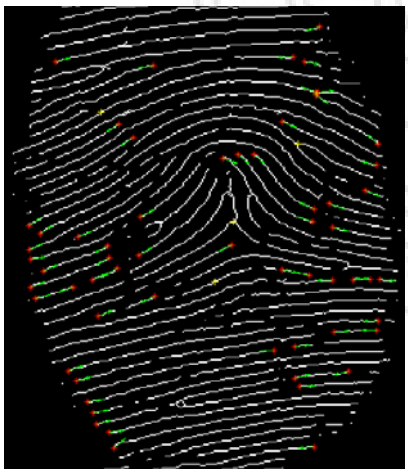


Figure 2.4 Real Minutiae after false removal

2.2. Minutiae Matching

The matching process is including the comparison of one set of minutiae data with another set. In most cases, this process compares an input data collection to a previously stored dataset with a known identity, defined as a template. The template is created during the enrollment process, when a user presents a finger for the system to collect the data from. This information is then classified as the defining characteristics for that particular user.

First researcher need to compare only two images since the whole process is done in MATLAB. If both of the images were from the same fingerprint they will be matched otherwise they are unmatched.

The following are the steps involved in the matching process:

1. First of all the two fingerprints are load into the matching function which are to be compared.
2. Minutiae points i.e. ridge ending and ridge bifurcation are loaded into the function that is extracted from both fingerprint images.
3. A built in MATLAB function ‘is equal’ is used to compare the two minutiae points i.e. ridge ending and ridge bifurcation of both the images with each other.
4. Only the fingerprint images are matched otherwise they are unmatched, if both the ridge endings and ridge bifurcations of the two fingerprint images matches with each other.

Then researcher can check our algorithm in Database.

2.3. National Institutes of Standards and Technology (NIST)

This report documents is the open source releases of the biometric picture software distribution developed by the National Institute of Standards and Technology (NIST) for the Federal Bureau of Investigation (FBI) and Department of Homeland Security (DHS). Its content and format is one of user's reference and guide manual. The cited references contain more details of how these fingerprint software technologies work, while some algorithmic overview is provided.

2.3.1 MINDTCT

The Home Office's Automatic Fingerprint Recognition System inspired the algorithms used in MINDTCT; specifically the suite of algorithms commonly called as "HO39."
The NIST software is a completely original implementation that exceeds the capabilities of HO39. It integrates new algorithms, a modular design, dynamic allocation, and flexible parameter control, which provide a framework to support future enhancement and adaptation of the technology. It should be considered that the algorithms and software parameters have been designed and set to perfect process images scanned at 19.69 pixels per millimeter (ppmm) (500 pixels per inch) and quantized to 256 levels of gray.

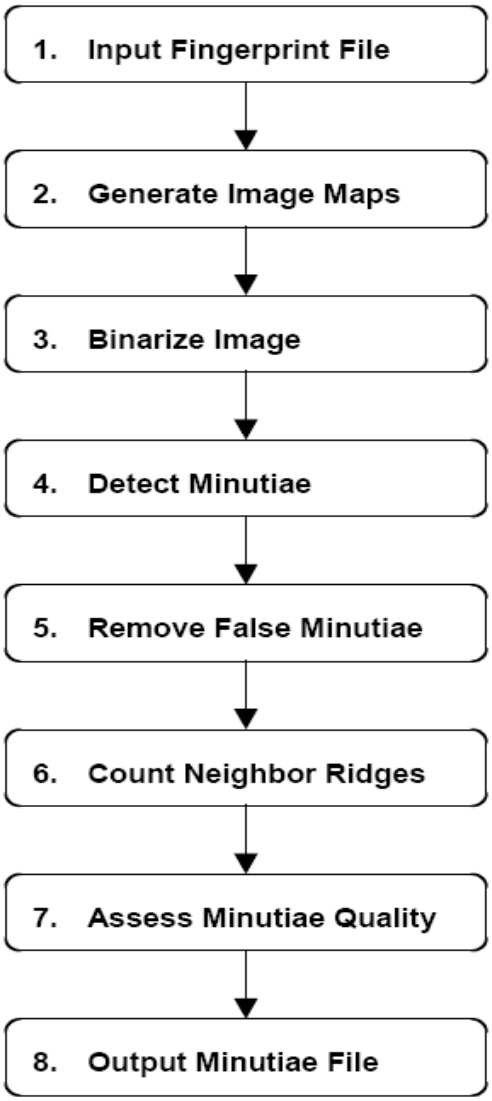


Figure 2.5 Minutiae detection process.

When the software is successfully installed and assembled, the program, MINDTCT, is available for detecting minutiae in a fingerprint image. This part describes each of the major steps in the minutiae detection process. It should be considered that two generations of minutiae detection have been developed before the release of this software.

2.3.1.1 Input Fingerprint Image File

MINDTCT bring in a fingerprint image and it automatically detects minutiae on the fingerprint. The algorithms and parameters have been developed and set for images scanned at 19.69 ppmm and quantized to 256 levels of gray. The application can read files in ANSI/NIST, WSQ, JPEGB, JPEGL, and IHEAD formats.

In ANSI/NIST formatted files it searches the file structure for a grayscale fingerprint record. When found, the fingerprint image in this record is processed.

MINDTCT has an option that will allow it to enhance very low contrast images. MINDTCT will evaluate the histogram of the input image, if the option is selected. It is enhanced to improve the contrast otherwise it is not modified, if the image is a very low contrast image.

2.3.1.2 Generate Image Quality Maps

It is critical to be able to analyze the image and determine areas that are declined and likely to cause problems, because the image quality of a fingerprint may vary, especially in the case of latent fingerprints. In this case several characteristics can be measured that are designed to convey information considering the quality of localized regions in the image. These include determining the directional flow of ridges in the image and finding out of regions of low contrast, low ridge flow, and high curvature. These three conditions show unstable areas in the image where minutiae detection is not reliable, and they can be used to show levels of quality in the image.

Each of these characteristics is discussed below.

2.3.1.2.1 Direction Map

A fundamental step in this minutiae detection process is deriving a directional ridge flow map, or *direction map*. The purpose of this map is to show areas of the image with sufficient ridge structure. These well-formed and clearly visible ridges are essential to reliably detecting points of ridge ending and bifurcation. Moreover, the direction map records the general orientation of the ridges as they flow across the image.

The image should be divided into a grid of *blocks* to locally analyze the fingerprint. All the pixels within a block are assigned to the same results. So, in the direction map, all the pixels in a block will be assigned the same ridge flow direction. Several considerations must be taken into account when using a block-based approach. First, it must be determined to reliably derive the desired characteristic, how much local information is required. This area is called *window*. The characteristic measured within the window is then assigned to each pixel in the block. It is normally desirable to share data used to compute the results assigned to neighboring blocks. In this case some of the image that dedicated to one block's results is included in the neighboring block's results as well. It cause minimize the discontinuity in block values as you cross the border from one block to its neighbor. This "smoothing" can be implemented by using a system where a block is smaller than its surrounding window, and windows overlap from one block to the next. This is shown in Figure 2.6

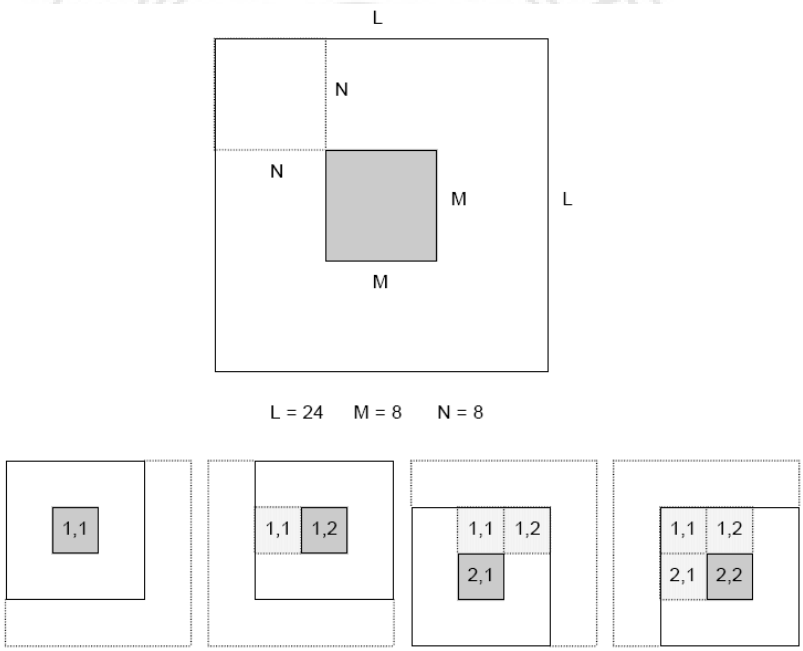


Figure 2.6 Adjacent blocks with overlapping windows.

Each of its window orientations is analyzed, when determining the direction of ridge flow for a block. Within an orientation, the pixels along each rotated row of the window are gathered together, forming a vector of 24 pixel row sums. The 16 orientations produce 16 vectors of row sums.

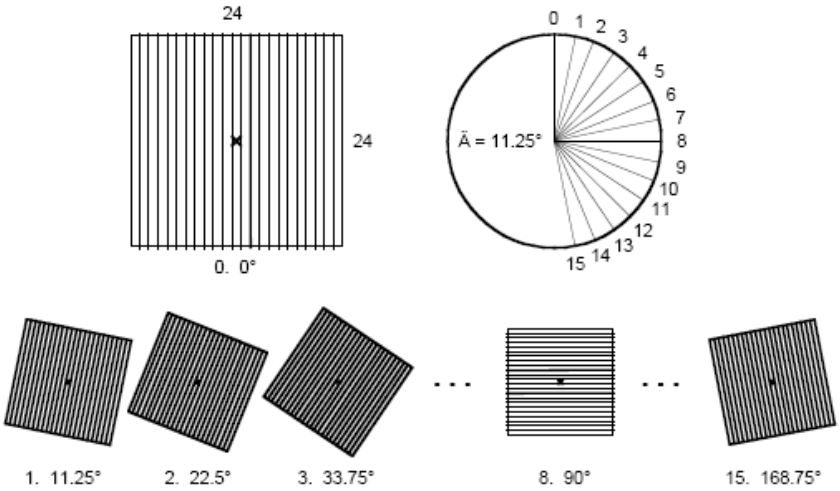


Figure 2.7 Window rotation at incremental orientations

2.3.1.2.2 Low Contrast Map

It is somehow difficult to accurately determine a prominent ridge flow in certain portions of a fingerprint image. This is low contrast areas that contain image background and smudges. It is favorable to detect these areas and prevent artificially assigning ridge flow directions where there are really no obviously defined ridges. Deriving an arbitrary ridge flow strictly from the data within these areas is somehow problematic.

An image map called as *low contrast map* and it is computed where blocks of sufficiently low contrast are flagged. This map separates the background of the image from the fingerprint, and it maps out smudges and lightly-inked areas of the fingerprint. In this case minutiae are not detected within low contrast blocks in the image.

One way to recognize a low contrast block from a block containing well-defined ridges is to compare their pixel intensity distributions. There is little dynamic range in pixel intensity in a low contrast area, so the distribution of pixel intensities will be very narrow. On the other hand, a block containing well-defined ridges will have a considerably wider range of pixel intensities as there will be pixels ranging from very light in the middle of valleys to very dark in the middle of ridges.

In order to determine if a block is low contrast, the software computes the pixel intensity

Distribution within the block's surrounding window. A specified percent of the distribution's high and low tails are arranged and the width of the remaining distribution is measured. The block is flagged in the map as having low contrast, if the measured width is sufficiently small.

Figure 2.8 Low contrast map results.

2.3.1.2.3 Low Flow Map

When deriving the initial direction map for some blocks it is possible to have no dominant ridge flow. These blocks normally correspond to low-quality areas in the image. At first these blocks are not assigned to an orientation in the direction map, but subsequently some of these blocks may be assigned to an orientation by inserting the ridge flow of neighboring blocks. The *low flow map* marks the blocks that could not initially be assigned to a prominent ridge flow.

In this case that minutiae are detected in these blocks, because they have been detected within a less reliable part of the image, their assigned quality is reduced. The white cross marks in the fingerprint image in Figure 2.9 label blocks with no prominent ridge flow.

Figure 2.9 Low flow map results.

2.3.1.2.4 High Curve Map

Another section of fingerprint image that is somehow problematic when it detects minutiae reliable is in areas of high curvature. This is true of the core and delta regions of a fingerprint. The *high curve map* marks blocks that are in high-curvature areas of the fingerprint. Two measures are used in this case. The first, called *vortices*, that measures the cumulative change in ridge flow direction around all the neighbors of a block. The second called, *curvature*, that measures the largest change in direction between a block's ridge flow and the ridge flow of each of its neighbors. The details located in the source code. In this case that minutiae are detected in these blocks, their assigned quality is reduced because they have been detected within a less reliable part of the image. The white cross marks in the fingerprint image in Figure 2.10 label blocks with high-curvature ridges.

Figure 2.10 High curve map results.

2.3.1.2.5 Quality Map

A *quality map* is the final image map produced by this package. As mentioned before, the low contrast map, low flow map, and the high curve map all of them point to different low quality regions of the image. The data in these maps is integrated and converted into one general map, as shown in Figure 2.11 and contains 5 levels of quality. The quality assigned to a specific block is determined based on its proximity to blocks flagged in these various maps. The details are in the source code.

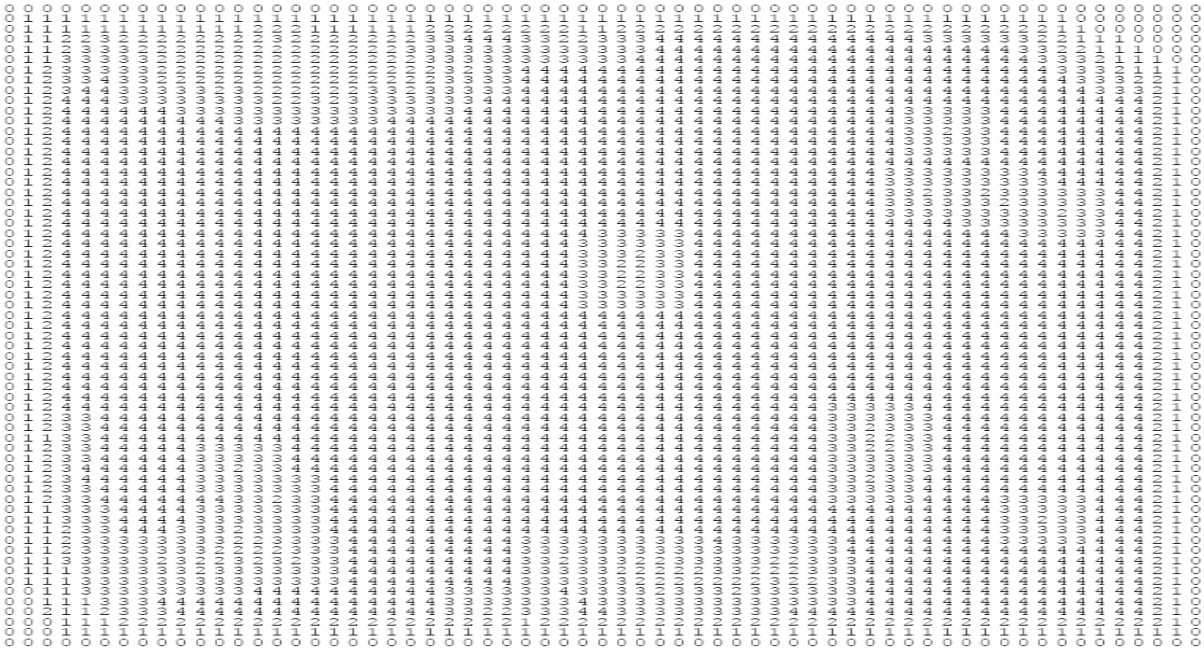


Figure 2.11 Quality map results

.

2.3.1.3 Binarize Image

The minutiae detection algorithm is designed to operate on a bi-level (or binary) image where black pixels show ridges and white pixels show valleys in a finger's friction skin. For creating this binary image, every pixel in the grayscale input image must be analyzed to identify if it should be assigned a black or white pixel. This process is called image *binarization*.

A pixel is assigned to a binary value based on the ridge flow direction that is associated with the block that is within a pixel. The pixel is set to white if there was no detectable ridge flow for the current pixel's block. The pixel intensities surrounding the current

pixel are analyzed within a rotated grid, if there is detected ridge flow, as shown in Figure 2.12.

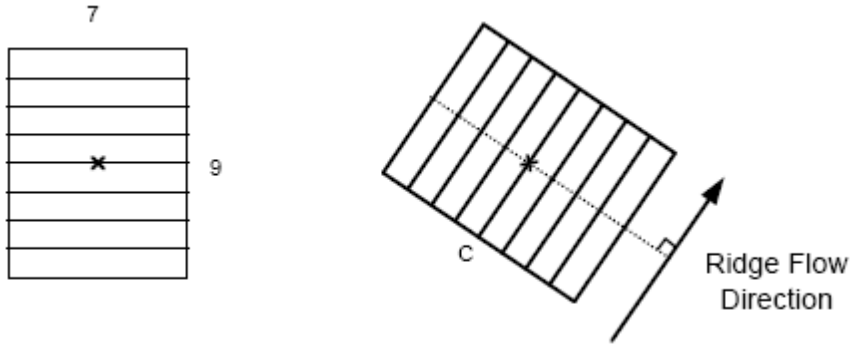


Figure 2.12 Rotated grid used to binarize the fingerprint image.

2.3.1.4 Detect Minutiae

This step scans the binary image of a fingerprint, and identifying localized pixel Patterns that indicate the ending or splitting of a ridge. The left-most pattern contains six binary pixels in a 2×3 configuration. This pattern may show the end of a black ridge inflicting into the pattern from the right. The same pattern is true for the next 2×4 pattern. The difference between this pattern and the first one is that the middle pixel pair is repeated. Also, this is true for all the patterns illustrated. This "family" of ridge ending patterns can be shown by the right-most pattern, where the middle pair of pixels may repeat one or more times.

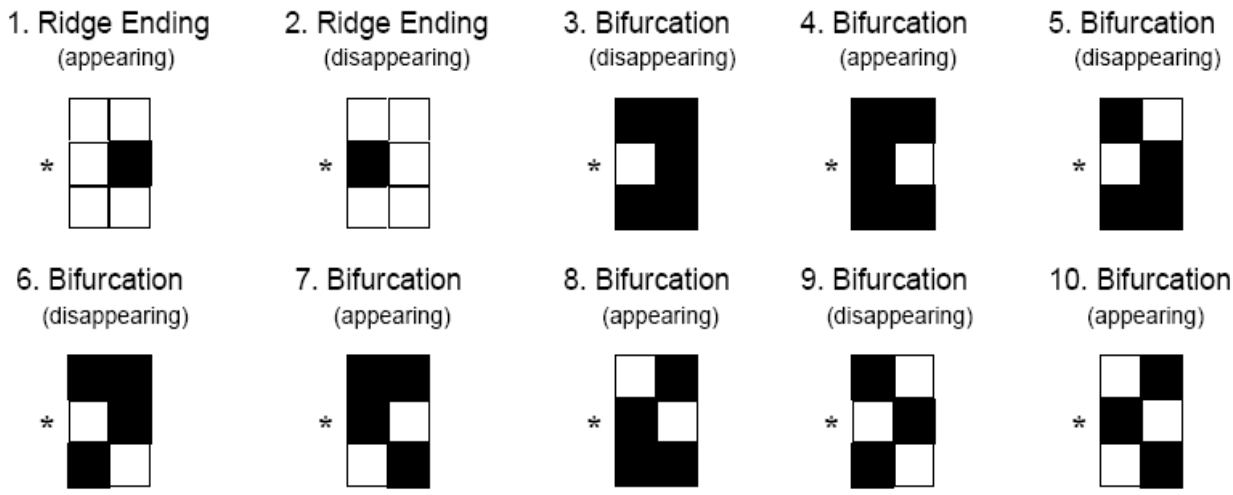


Figure 2.13 Pixel patterns used to detect minutiae.

Candidate ridge endings are detected in the binary image by scanning consecutive pairs of pixels in the image searching sequences that match this pattern. Pattern scanning is conducted both vertically and horizontally. The pattern as shown is configured for vertical scanning as the pixel pairs are stacked on top of each other. To conduct the horizontal scan, the pixel pairs are unstacked, rotated 90° clockwise, and placed back in sequence left to right.

2.3.1.5 Remove False Minutiae

With using the patterns in Figure 2.13, candidate minutiae points are detected with as few as six pixels.

This can facilitates an especially greedy detection scheme that minimizes the chance of missing true minutiae; anyhow, many false minutiae are included in the candidate list. Because of this matter, much effort is spent on removing the false minutiae. These steps include removing islands, lakes, holes, minutiae in regions of poor image quality, side minutiae, hooks, overlaps, minutiae that are too wide, and minutiae that are too narrow (pores). A short description of each of these steps is provided in the way that they are executed.

2.3.1.6 Count Neighbor Ridges

Matchers of fingerprint minutiae often use information in addition to only the points themselves.

Additional information usually includes the minutiae's direction, its type, and it may include information relating to minutiae neighbors. There are no standard neighbor schemes beyond a minutiae's position, direction, and type. Different AFIS systems use different neighbor topologies and attributes. One of the common attributes is the number of intervening ridges (called ridge crossings) between a minutiae and each of its neighbors. For example, the FBI's IAFIS uses ridge crossings between a minutiae and its 8 nearest neighbors, where each neighbor is the closest one within a specified octant.

The neighbor scheme that is distributed with this system, has been directly inherited from HO39.Up to 5 nearest neighbors (MAX_NBRS) are reported till now. In a minutiae point, the closest neighbors below (in the same pixel column), and to the right (within entire pixel columns) in the image are selected. These nearest neighbors are sorted based on their direction, starting with vertical and working clockwise. Applying this topology, ridge counts are computed and recorded between a minutiae point and each of its nearest neighbors.

2.3.1.7 Assess Minutiae Quality

One of the objectives of developing this software package was to compute a quality/reliability that is associated with each detected minutia point. False minutiae potentially remain in the candidate list, even with the lengthy list of removal steps above. A high quality measure can help manage the false minutiae that should be assigned a lower quality than true minutiae. A tradeoff between retaining false minutiae and throwing away true minutiae may be determined through dynamic thresholding. In this case, MINDTCT, computes and reports minutiae qualities.

There are two factors that are combined to produce a quality measure for each detected minutia point. The first factor, *L*, is taken directly from the location of the minutiae point within the quality map described in Section 5.2.1.6. One of the five quality levels is initially assigned, with 4 being the highest quality and 0 being the lowest.

The second factor is based on simple pixel intensity statistics (mean and standard deviation) within the immediate neighborhood of the minutiae point. The size of the neighborhood is set to 11 pixels (RADIUS_MM). This is sufficiently large to contain numerous portions of an average ridge and valley. A high quality region within a fingerprint image will have significant contrast that will cover the full grayscale spectrum. As a result, the mean pixel intensity of the neighborhood will be very close to 127. Similarly, the pixel intensities of an ideal neighborhood will have a standard deviation >= 64.

2.3.1.8 Output Minutiae File

When it is completed, MINDTCT, takes the resulting minutiae and outputs them to a file. If the input file was an ANSI/NIST formatted file, MINDTCT adds two new records and writes a new ANSI/NIST formatted file to <oroot>.mdt, where <oroot> is passed as a parameter to MINDTCT.

3. Literature Review

3.1. A Matching Algorithm of Minutiae for Real Time Fingerprint Identification System Shahram Mohammadi, Ali Frajzadeh

In recent years a lot of matching algorithms with different characteristics have been introduced. In real time systems these algorithms are usually based on minutiae features. Here researcher introduce a novel approach for feature extraction in which the extracted features are not dependent to shift and rotation of the fingerprint and by the way the performance of matching operation is easy and fast and accurate. In this new approaches first for any fingerprint a reference point and a reference orientation is determined and then they are converted into polar coordinates based on this information features. This approach is the most appropriate for real time applications,
due to high speed , accuracy of this approach , small volume of extracted features and easily execution of matching operation .
The main image (Figure 3.1(a)) will be enhanced by using Gabor filter and a binary image will result in (Figure 3.1(b)) considering a disk neighborhood centered at reference point which has a radius proportional to image dimensions and reference point position (Figure 3.1(c)). If intensity of up core=0(up core on the ridge) (Fig 3.1 (d)); then intensity of every pixel in the previous step is totalized (Figure 3.1(e)). For the consequent image from previous step a stretch conversion is performed by using a squared neighborhood centered at reference point with dimension of 15×15 i.e. Morphological operations with structuring element=5 (Figure 3.1(f)). Now from the consequent image all connecting points with up core are obtained (Fig 3.1 (g)).

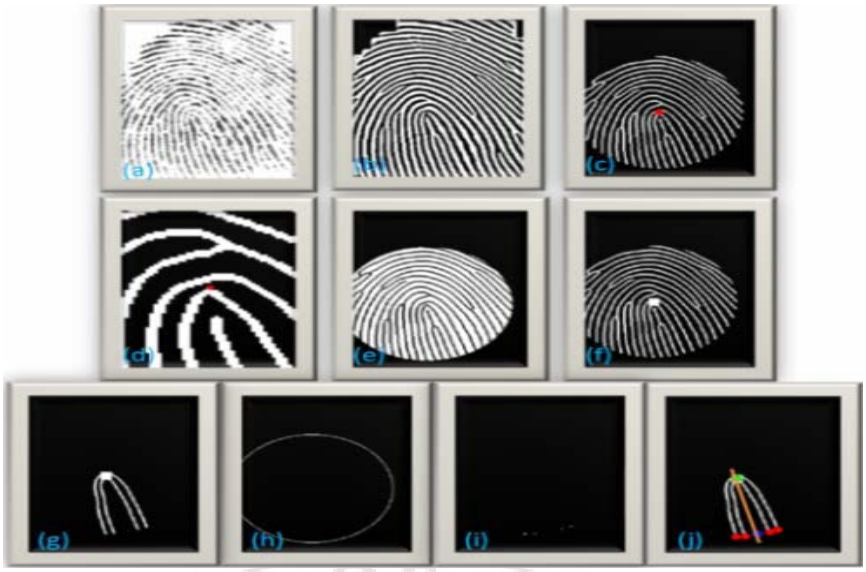


Figure 3.1 Reference orientation computation

Increase in the resulting image with a circular mask with previous radius of one pixel thickness that in Fig 3.1 (i) will be obtained, provides the approximate position of ends of ridges or valleys. Mean value of lengths and widths of each position will provide the exact position for each of them (red points in Figure 3.1(g)).Getting the mean value of widths and lengths of red points will result in a blue point. Angle of this point considering reference point (green point) is the reference orientation of the image (Figure 3.1(j)).

Hence during scanning the fingerprint it may has been translated and/or oriented, the extracted features must be separate from these operations. In this new method the features are extracted somehow that are separate from these operations. Features are extracted based on the reference point and reference orientation computed in previous steps. For every minutiae of each fingerprint e.g. nth minutiae, feature vector is extracted as follows:

$$(d_{n,}t_{n,}\theta_{1(n),}\theta_{2(n)})$$

d Is the distance of nth minutiae from reference point (Fig 3.2). If characteristics of the reference point are $(X_{rp}, Y_{rp})$and $(X_{m(n)}, Y_{m(n)})$characteristics of nth minutiae are

$$d_{n=}\sqrt{(X_{rp-}X_{m(n)})^2+(Y_{rp-}Y_{m(n)})^2}$$

t n is the type of minutiae which is ridge ending or ridge bifurcation . For ridge ending number 1 and for bifurcation number 2 has been considered. θ 1 (*n*) is:

θ 1 (*n*) =θ *n* −θ *r*

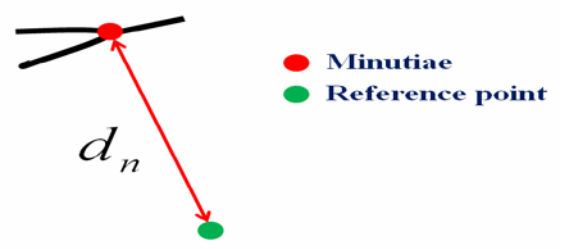


Figure 3.2 Distance of nth minutiae from reference point.

In the above relation, θ *n* is the angle of the line that connect reference point and the nth minutiae according to horizontal axis and θ *r* is the angle of the line that connect the reference point and reference orientation according to horizontal axis and both of them are between 0 and 2π radians (Fig 3.3). A minutiae directional angle is defined for every minutiae. This angle can be between 0 and 2π radians (Fig 3.4). θ 2(*n*) is assumed as follows:

$$\theta_{2(n)} = \theta_{m(n)} - \theta_r$$

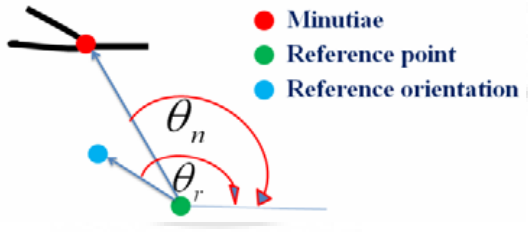


Figure 3.3 Angle of minutiae respect to reference point and reference orientation.

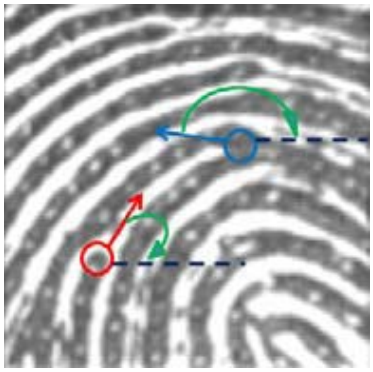


Figure 3.4 Minutiae directional angle (red circle is ridge ending and blue circle is ridge bifurcation)

Here θ *m*(*n*) is the minutiae directional angel according to horizontal axis and θ *r* is the angle of the line that connect the reference point and reference orientation according to horizontal axis and θ *m*(*n*) and θ *r* are between 0 and 2π radians(Fig 3.5).

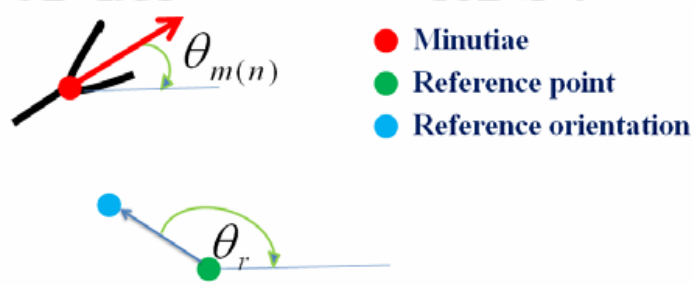


Figure 3.5 Minutiae directional angel with respect to reference point and reference orientation.

Moreover mentioned features which are known as local features, also there are other features which are known as global features (such as singular points) and the classification of them is based on type and position of the singular points. These increase accuracy and speed of operation and decrease the complexity of the algorithm. Here all images are classified into five classes (Fig 3.6).

Figure 3.6 Example Fingerprint image classes: (a) Whorl, (b) Right Loop, (c) Left Loop, (d) Arch, and (e) Tent Arch

3.2. Orientation feature for fingerprint matching Jayant V. Kulkarni∗, Bhushan D. Patil, Raghunath S. Holambe

A fingerprint verification algorithm based on the orientation field is described here. The orientation field of a fingerprint image has also been used for image alignment. Area around the core point has been occupied as an area of interest for determining the orientation feature map.

The orientation field of a fingerprint image shows that ridges are directional. Fingerprint image normally divided into number of non-overlapping blocks and an orientation representative of the ridges in the block is assigned to the block based on grayscale gradients in the block. The block size depends on the inter-ridge distance, i.e. it should cover at least one ridge and one valley in a block. The block orientation can be recognized from the pixel gradients by averaging or voting (optimization). The orientation field of block *(i, j)* is given by:

$$\theta(i,j) = 0.5tan^{-1}\left(\frac{V_x(i,j)}{V_y(i,j)}\right),$$

Where

$$V_x(i,j) = \sum_{u=i-w/2}^{i+w/2} \sum_{v=j-w/2}^{j+w/2} (G_x(u,v)G_y(u,v))$$

And

$$V_y(i,j) = \sum_{u=i-w/2}^{i+w/2} \sum_{v=j-w/2}^{j+w/2} (G_x^2(u,v)G_y^2(u,v))$$

Where *w* is the size of block, *Gx* and *Gy* are the gradient magnitudes in *x* and *y* directions. Sobel operators have been used for calculations of *Gx* and *Gy* .

3.3. A fast and elastic fingerprint matching algorithm using minutiae-centeredcircular regions Haiyong Chen, Hongwei Sun, Kwok-Yan Lam

One of the challenging problems in a fingerprint verification system is the reliable and fast matching fingerprints. A minutiae matching algorithm uses minutiae-centered circular regions to ensure the speed of matching and the strongness to non-linear distortion. In this method, a circular region is constructed around each minutiae, which can be considered as a secondary feature. By using the constructed regions, the suggested algorithm can find matched minutiae fast via regional matching. Hence each minutiae's region is consist of only a small area of the fingerprint, this algorithm is more enduring to non-linear distortion when it is compared to global matching approaches. From a different angle, the area of the constructed region is much larger than that of local neighborhood in local matching approaches, which means that circular region, including a larger subnet of minutiae, is more reliable and distinct feature.

Experiment results show that this algorithm has good performance on processing speed and accuracy.

Fingerprint minutiae should be extracted first before the matching phase. In this method, minutiae is represented by only two attributes: location and orientation. A minutiae is represented as (x, y, α) where (x, y) is the position and α is the orientation.

This method includes the following stages:

Firstly, researcher build a circular region with the same radius around each minutiae and each region is identified by the centered minutiae.

Secondly, researchers go through regional minutiae matching until he find the first pair of corresponding regions.

Thirdly, since circular region is identified by the centered minutiae, it can identify more pairs of matched regions by using matched minutiae. Then, researcher can go through regional minutiae matching between newly matched regions repeatedly until he find all pairs of minutiae (regions).

Finally researcher can determine whether the two fingerprints are from the same finger according to the number of all matched minutiae.

Researchers build a circular region around each minutiae. Each region is identified by the centered minutiae, and has the same radius which is adjustable parameter based on experimental needs. The quantity of circular region we build equals to the minutiae in a fingerprint. Region information of each minutiae consists of minutiae in its region. Different circular regions have different quantity of minutiae based on the random distribution of minutiae. Minutiae near the border of fingerprint may have a partial circular region. For example, let {A1,A2,…,Am} define the set of *m* minutiae in a fingerprint, and K(Ai) defines the corresponding circular region in which minutiae Ai is the center. Any built region is represented by the set of regional minutiae. In figure 3.7.A1 is a minutiae, and K(A1) is the corresponding region. 6 minutiae including A1 are in this region. In general, the radius of our region has both upper limit and lower limit. An entire region is about 15%~35% of the fingerprint according to the change of radius. Because circular region is determined by a larger part of the fingerprint when it compared to local structure of local matching methods, the circular region is more distinctioning.

From a different angle, circular region is also a small part of the global fingerprint, so it is huge to non-liner distortion and has high processing speed.

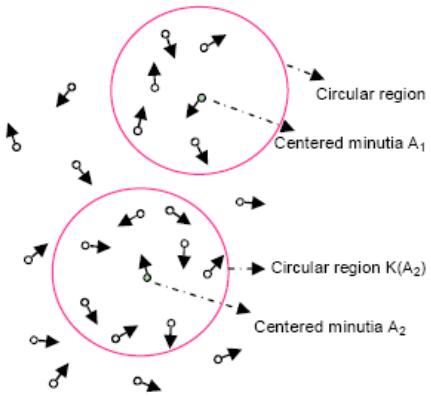


Figure 3.7 minutiae and circular regions

Now researchers go through regional minutiae matching until finding the first pair of corresponding regions. In this step it is crucial to find other pairs of corresponding minutiae (regions) based on it. For ensuring the accuracy of the entire matching process, finding the first pair of corresponding regions is very important.

Let {A0,A1,…,Am-1}define the set of *m* minutiae in the template and {B0,B1, …, Bk-1} define the set of *k* minutiae in the input image. So {K(A0), K(A1), …,K(Am-1)} is the set of circular region in the template, and {K(B0), K(B1), …, K(Bk-1)} is the set of circular region in the input image. The algorithm of finding the first pair of corresponding regions is described as follows:

Algorithm:

Find the first pair of corresponding regions

Input:

{K(A0), K(A1), …, K(Am-1)}

{K(B0), K(B1), …, K(Bk-1)}

Output: A pair of corresponding regions

Declaration:

Regional Match(K(Ai),K(Bj))–a regional minutiae matching function to calculate the matched minutiae between region K(Ai) and region K(Bj) which assumes that Ai and Bj are reference minutiae as default.

threshold(reg_matched_num)–the least number of matched minutiae of two corresponding regions.

Method:

for (i=0;i<k;i++)

for (j=0;j<m;j++)

if (RegionalMatch(K(Ai),K(Bj))>=

threshold (reg_matched_num))

return (i, j);

return (-1,-1)

In any regional minutiae matching process, for example Regional Match (K(Ai),K(Bj)), Ai and Bj are reference minutiae as default . Let A be a minutiae in K(Ai), let B be a minutiae in K(Bj). The process in which determines whether A and B are matched minutiae can be described as follows:

A(x(A), y(A), α(A)) is a minutiae of region K(Ai) in the template and B(x(B), y(B), α(B)) is a minutiae of region K(Bj) in a sample fingerprint. Researchers align and convert each minutiae to the polar coordinate system due to the corresponding reference minutiae.

Let A(r(A),θ(A), β(A)) and B(r(B),θ(B), β(B)) be the aligned representation in polar coordinate system (r represents the radial distance, θ represents the radial angle, and β represents the orientation of the minutiae with respect to the reference minutiae).

Assume:

Da=α(Bj)-α(Ai) , then $r(A) = \sqrt{[x(A) - x(A_i)]^2} + [y(A) - y(A_i)]^2$

$$\theta(A) = \begin{cases} 90: \ x(A) = x(Af), y(A) > y(Af) \\ -90: \ x(A) = x(Af), y(A) < y(Af) \\ 0: \ x(A) = x(Af), y(A) = y(Af) \end{cases}$$

$$\theta(A) = \left\{ a \ tan \frac{y(A) - y(At)}{x(A) - x(At)} \right\} \times \frac{180}{\pi} + 180: \ x(A) < x(At)$$

$$\theta(A) = \left\{ a \ tan \frac{y(A) - y(At)}{x(A) - x(At)} \right\} \times \frac{180}{\pi}: \ x(A) > x(At), y(A) >= y(At)$$

$$\theta(A) = \left\{ a\ tan \frac{y(A) - y(At)}{x(A) - x(At)} \right\} \times \frac{180}{\pi} + 360:\ x(A) > x(At), y(A) < y(At)$$

$$\beta(A) = a(A)$$

$$r(B) = \sqrt{\left[x(B) - x(B_f)\right]^2 + \left[y(B) - y(B_f)\right]^2}$$

Assume:

$$temp = \begin{cases} 90:\ x(B) = x(Bf), y(B) > y(Bf) \\ -90:\ x(B) = x(Bf), y(B) < y(Bf) \\ 0:\ x(B) = x(Bf), y(B) = y(Bf) \end{cases}$$

$$temp = \left\{ a\ tan \frac{y(B) - y(Bf)}{x(B) - x(Bf)} \right\} \times \frac{180}{\pi} + 180:\ x(B) < x(Bf)$$

$$temp = \left\{ a\ tan \frac{y(B) - y(Bf)}{x(B) - x(Bf)} \right\} \times \frac{180}{\pi}:\ x(B) > x(Bf), y(B) >= y(Bf)$$

$$temp = \left\{ a\ tan \frac{y(B) - y(Bf)}{x(B) - x(Bf)} \right\} \times \frac{180}{\pi} + 360:\ x(B) > x(Bf), y(B) < y(Bf)$$

$$temp = temp - Da; then$$

$$\theta(B) = \begin{cases} temp - 360; & temp \geq 360 \\ temp + 360; & temp < 0 \\ temp; & 0 \leq temp < 360 \end{cases}$$

$$\beta(B) = \begin{cases} \alpha(B) - Da; & \alpha(B) - Da \geq 0 \\ \alpha(B) - Da + 360; & \alpha(B) - Da < 0 \end{cases}$$

If the following in equations can be satisfied at the same time, A and B are matched minutiae pairs.

$$|r(A) - r(B)| < threshold(r)$$

$$|\theta(A) - \theta(B)| < threshold(\theta)$$

$$|\beta(A) - \beta(B)| < threshold(\beta)$$

$$\sqrt{(r(A)-r(B))^2+(\theta(A)-\theta(B))^2+(\beta(A)-\beta(B))^2} < threshold(T)$$

Since circular region is identified by the centered minutiae, then researcher can identify more pairs of matched regions using matched minutiae. Then researcher can go through regional minutiae matching between newly matched regions repeatedly until he fined all pairs of minutiae (regions). This process is very fast since it is very easy to find newly matched minutiae in small matched regions.

Finally, researcher can determine whether the two fingerprints are from the same finger according to the number of all matched minutiae.

3.4. A Fingerprint Matching Algorithm Based On Delaunay Triangulation Net1 Ning Liu, Yilong Yin, Hongwei Zhang

Fingerprint matching is one of the key issues in an automatic fingerprint identification system. Due to the Delaunay triangulation (DT) in computational geometry, researcher should suggest a fingerprint matching algorithm based on DT net in this paper. It uses DT in fingerprint matching, and then develops a matching algorithm based on DT net to find reference minutiae pairs (RMPs). If researchers use DT on the topological structure of minutiae set, he found that DT net is formed with minutiae as vertexes. From the nets of the input minutiae set and template minutiae set, researcher select a certain pairs of minutiae which have similar structures as RMPs for aligning, and then matching is carried out based on point pattern.

Minutiae set that is extracted from fingerprint, can be considered as scattered points on the plane. Triangulation is an effective tool for working with scattered data set. In this case the net produced by triangulation with a suitable rule, has good performance, for example, the uniqueness (assuming no degeneracy) and the good local stability.

So the triangulation can be applied to fingerprint matching. Here, the DT net is used to find RMPs, which are used for aligning. DT algorithm is used to form a DT net of a minutiae set. The time complexity of the algorithm is O (N log$N$) and N is the number of minutiae. Fig3.9 shows the process of forming DT net of minutiae set and in this

case the DT net of minutiae set formed by triangulation using Delaunay criterion is shown in Fig3.8 .The algorithm consists of four steps described as follows:

1. Preprocessing the data: calculate the uniform grid and then put each minutiae in one and only one grid cell.

2. Finding the first edge: Find a start minutiae by searching from center of grid or from neighbor of center of grid. Then, find second minutiae, which is closest minutiae among the neighbors of the start point. Finally, link the two minutiae to form the first edge.

3. Forming triangles: Search on the right-hand side of the first edge and find a minutiae, which gives the largest angle. Then, take the circle defined by the original triangle and set the min-max box of the circle. Then select the minutiae with the largest angle in the min-max box and repeat the same as above. Then stop the search when there are no unvisited rows and columns inside the current min-max box.

4. Putting triangles together: Add triangles continuously so that the added triangles are simply connected. So there are no holes and bridges with the already chosen triangles. The algorithm proceeds and considers all the cases to ensure that the triangulation is complete and accurate.

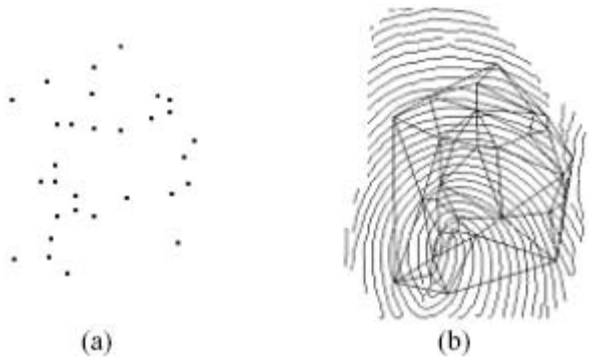


Figure 3.8 DT net of minutiae set. (a) Minutiae set, (b) The DT net taking thinning fingerprint image as background

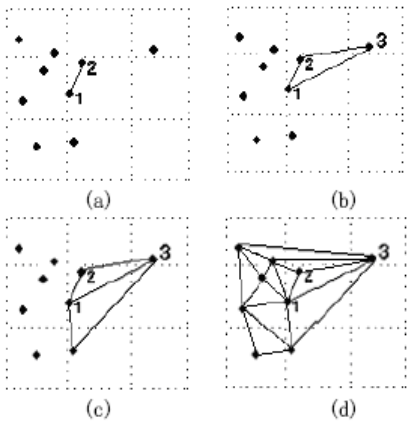


Figure 3.9 Forming DT net of a minutiae set. (a) Finding the first edge, (b) Forming the first triangle, (c) Adding the second triangle on clockwise direction and (d) Putting triangles together

In fingerprint matching, triangulation net of scattered points on the plane has three characteristics described as follows:

1. Triangulation net is unique (The scattered points is assumed non-degenerate). This guarantees that a minutiae set can always get same triangulation net.

2. Triangulation net has very good local stability. This guarantees that triangulation nets of fingerprints obtained from the same finger are basically same.

3. Triangulation algorithm of 2D data points has a tested linear time complexity. This is a suitable algorithm for an on-line automatic fingerprint identification system.

Matching the DT nets is used for finding several RMP. At first, some pairs of similar edge pairs (SEPs) can be found via matching. With having the SEPs, the corresponding similar triangle pairs (STPs) can be found quickly. The three pairs of vertexes of each STP are the three RMPs that is needed in alignment.

The procedure of seeking RMPs is schemed. Figure 3.10 shows local structures and features, fig3.11 shows the procedure of matching as follows:

1. Two corresponding DT nets have produced by the algorithm with a template image and input image. Minutiae can be described as a vector $(X_t, Y_t, \theta_t)^T$in template image or as$(X_i, Y_i, \theta_i)^T$in input image. However, edge can be described as a vector $(\theta_{th}, \theta_{tb}, Y_t, l_t)^T$in template DT net or$(\theta_{ih}, \theta_{ib}, Y_i, l_i)^T$in input DT net. In vectors, x, y define x, y coordinates of minutiae, and θ, y define minutiae angle and orientation of edge, and l defines length of edge. In, $(\theta_{th}, \theta_{tb}, Y_t, l_t)^T \theta_{th}$corresponds to the vertex

with smaller x coordinates and θ_{tb}corresponds to the vertex with larger x coordinates. The same to vector$(\theta_{ih}, \theta_{ib}, Y_i, l_i)^T$.

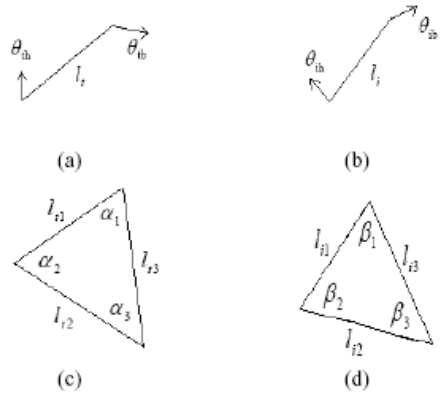


Figure 3.10 Local structures and features. (a),(b) DT edges,(c), (d) DT triangles

2. Edge matching is used to find SEP. Choose two edges $T(\theta_{th}, \theta_{tb}, Y_t, l_t)^T$ and $I(\theta_{ih}, \theta_{ib}, Y_i, l_i)^T$from the template and input DT net. Only when the equations (1), (2) and (3) are completed, T and I are considered SEPs. The equations, (2) and (3) are defined as follows:

$$|l_t - l_i| < T_1 \quad (1)$$

$$\big||\theta_{th} - \theta_{ih}| - |\theta_{tb} - \theta_{ib}|\big| < T_2 \quad (2)$$

$$|Y_t - Y_i| < T_3 \quad (3)$$

Where, $T_1, T_2\ and\ T_3$ are thresholds of length of edge, minutiae angle and orientation of edge. If template DT net has M edges and input DT net has N edges, the total number of edge comparison will be M*N. If there is no similar edge , the local matching is rejected, which means the two fingerprints are not generated from the same finger.

3. Triangle matching is used to find STPS, to choose a SEP “T” and “I” and to search a triangle on the right-hand side of them. Researcher search triangle on the left-hand side of them if they are edges on right border. Researcher search two sides of them if they are edges on right border. So, every pair of similar edges can produce one or two pairs of triangles.

Let vector $(l_{t1}, l_{t2}, l_{t3}, \alpha_1, \alpha_2, \alpha_3)^T$and $(l_{i1}, l_{i2}, l_{i3}, \beta_1, \beta_2, \beta_3)^T$ define two triangles found on the right-hand side of T and I. In vectors, $l_{t1}, l_{t2}, l_{t3}, \alpha_1, \alpha_2, \alpha_3$ define length

of six edges of two triangles and $l_{i1}, l_{i2}, l_{i3}, \beta_1, \beta_2, \beta_3$ define six angles of two triangles. According to suffix, these weights of two vectors are one to one correspondence. If the two triangles can satisfy the following in equations (4) and (5), they will be considered as a STP.

$$\left|l_{tj} - l_{ij}\right| < T_4, j = 2,3 \quad (4)$$

$$\left|\alpha_j - \beta_j\right| < T_5, j = 1,2,3 \quad (5)$$

Where, $l_{t1} = l_t, l_{i1} = l$, T_4 *and* T_5 are thresholds of length of edge and angle. Vertexes of STPs are used as RMPs. If there is no STP after intersecting all pairs of triangles, the matching should be rejected.

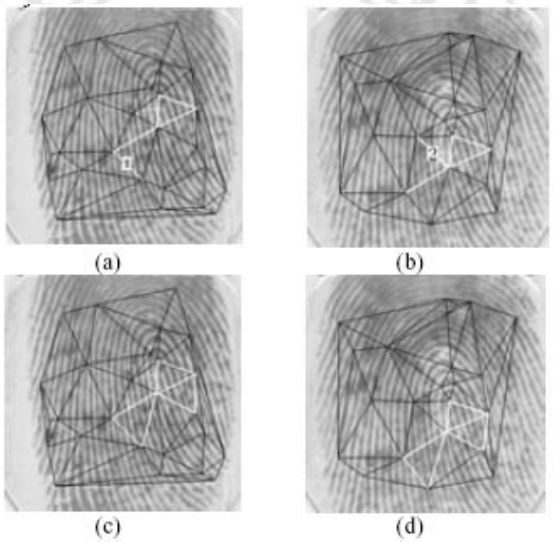


Figure 3.11 The procedure of matching two fingerprints. (a)and (b) are DT nets of two fingerprints, white lines are five SEPs, but edge 1 and 2 are a pseudo pair, (c) and (d) are three genuine STPs and six RMPs according to the similar lines in (a) and (b),

3.5. Delaunay Triangulation Algorithm for Fingerprint Matching Chengfeng Wang and Marina L. Gavrilova

The aim of fingerprint identification is to determine whether two fingerprints are from the same finger or not. For this reason, the input fingerprint needs to be aligned with the template fingerprint represented by its minutiae pattern. The following hard transformation can be performed:

$$F_{s\Delta\theta\Delta x\Delta\gamma}\begin{pmatrix}X_{tempt}\\Y_{tempt}\end{pmatrix} = s\begin{pmatrix}cos\Delta\theta - sin\Delta\theta\\sin\Delta\theta - cos\Delta\theta\end{pmatrix}\begin{pmatrix}X_{input}\\Y_{input}\end{pmatrix} + \begin{pmatrix}\Delta x\\\Delta y\end{pmatrix} \quad (1)$$

where $(s\Delta\theta\Delta x\Delta\gamma)$show a set of hard transformation parameters: (scale, rotation, translation). Through simple transformation, a point can be transformed to its corresponding point after rotation $\Delta\theta$ and translation$(\Delta x, \Delta y)$.

Since both images are captured with the same device, researchers assume that the scaling factor between input and template images is same.

Let $Q = (\left(x_1^Q, y_1^Q, \theta_1^Q, t_1^Q\right) \dots \left(x_n^Q, y_n^Q, \theta_n^Q, t_n^Q\right))$define the set of *n* minutiae in the input image ((x,y): location of minutiae θ : orientation field of minutiae; *t*: minutiae type, end or bifurcation;) and $P = \left(\left(x_1^p, y_1^p, \theta_1^p, t_1^p\right) \dots \left(x_m^p, y_m^p, \theta_m^p, t_m^p\right)\right)$define the set of m minutiae in template image.

Feature	Fields					
Minutiae Point	x (Y)	y (Y)	θ (Y)	Type (N)		
Triangle Edge	Length (N)	$\theta 1$ (N)	$\theta 2$ (N)	Type1 (N)	Type2 (N)	Ridge Count (N)

Table 1. DT data structure used for comparing fingerprint images: Y: Dependent on fingerprint transformation; N: Independent of it.

Table 1 shows features that researcher can use in local and global matching of fingerprints. In the above table, *Length* is the length of edge; $\theta 1$, is the angle between the edge and the orientation field at the first minutiae point; *Type1* define minutiae type of the first minutiae; *Ridge count* is the number of ridges that these two minutiae points cross. If researchers use triangle edge as comparing index we get many advantages. For local matching, the first compute the Delaunay triangulation of minutiae sets Q and P. Second is using triangle edge as comparing index. For comparing two edges, parameters such as *Length*, $\theta 1$,$\theta 2$, *Type1*, *Type2*, and *Ridge count* are used. Consider these parameters are equal to the translation and rotation (see Table 1). It means their values do not change when there is only hard transformation of fingerprint. A sample set of conditions for determining whether two edges match or not can be found in it . If the threshold is selected successfully, transformation$(\Delta\theta, \Delta x, \Delta y)$ is obtained by aligning the input and template images. Matching using Delaunay triangulation edge is realized as follows:

If one edge from an input image matches two edges from the template image, then researcher need to consider the triangulation to which this triangle edge belongs and then compare the triangle pair for a certain range of translation and rotation scattering ,detect the peak in the transformation space, and record transformations that are neighbors of the peak in the transformation space. Remember those recorded transformations are close to each other, but not the same. Also remember elastic deformations should be done separately as such deformations cannot be ignored. Figure 3.12 shows the successfully matched Delaunay triangle edge pairs. Algorithm shows the pseudo code for local matching.

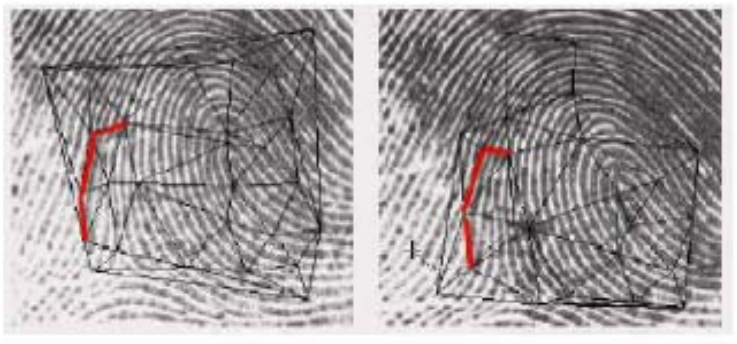


Figure 3.12 Delaunay triangle edges are matched, but not a single

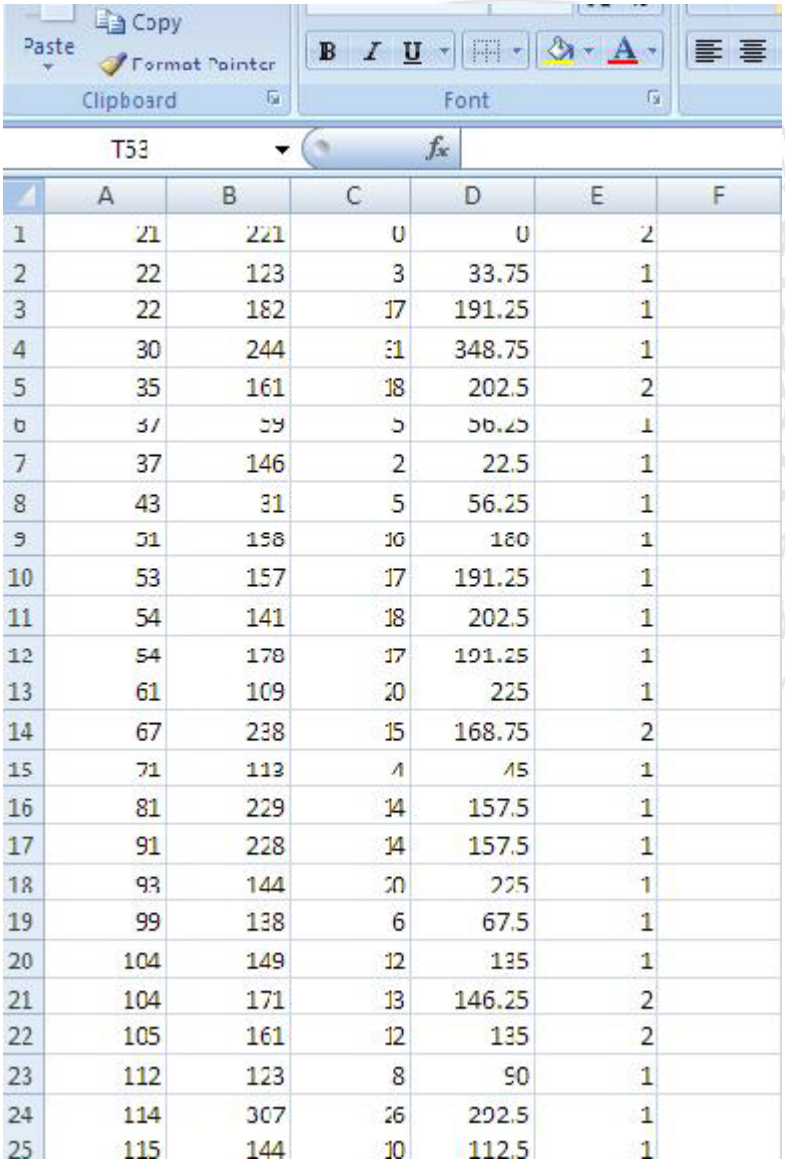
Algorithm Local Matching

(Triangle Edge Q, Triangle Edge P)

1. K=0
2. For i=1: n // Delaunay triangle edge in Q
3. For j=1: m // Delaunay triangle edge in P
4. If Triangle Edge Q(i) match Triangle Edge P(j)
5. Calculate the transformation parameters (θ, x, y)
6. And record this match as A[k];
7. k++;
8. Discrete (θ, x, y) parameters into Hough space(a three dimension array count);
9. For I=1: k
10. Obtain Hough space A[i] falls into ,for example
    $(\theta[m1], x[m2], y[m3])$
11. Count$[m1][m2][m3] + +$;
12. Detect the maximum value in Hough space as $(\theta m, xm, ym)$
13. For i=1: k
14. If A[i] are round equal $(\theta m, xm, ym)$
15. Record A[i] as effective local matching and Corresponding control points;

4.METHODOLOGY

We will explain how the experiments were performed in this chapter. First, a description of first matching algorithm. In this experiment we tried two finger print image. For minutiae extraction we used NBIS software provided by NIST, we just using information that contain minutiae X coordinate and Y coordinate and orientation define by θ and type of minutiae that is ridge bifurcation shows with 2 or ridge ending that shows with 1.



	A	B	C	D	E	F
1	21	221	0	0	2	
2	22	123	3	33.75	1	
3	22	182	17	191.25	1	
4	30	244	31	348.75	1	
5	35	161	18	202.5	2	
6	37	39	5	56.25	1	
7	37	146	2	22.5	1	
8	43	31	5	56.25	1	
9	51	158	16	180	1	
10	53	157	17	191.25	1	
11	54	141	18	202.5	1	
12	54	178	17	191.25	1	
13	61	109	20	225	1	
14	67	238	15	168.75	2	
15	71	113	4	45	1	
16	81	229	14	157.5	1	
17	91	228	14	157.5	1	
18	93	144	20	225	1	
19	99	138	6	67.5	1	
20	104	149	12	135	1	
21	104	171	13	146.25	2	
22	105	161	12	135	2	
23	112	123	8	90	1	
24	114	307	26	292.5	1	
25	115	144	10	112.5	1	

Figure 4.1

After getting such information using that in MATLAB to build Delaunay triangle, means each minutiae be as vertex so each 3 minutiae cause 1 triangle, so we will apply this for both fingerprint.

And how we get these triangles in 4 steps

1. Uniform grid and then put each minutiae in one and only one grid cell.

2. Find start minutiae by searching from center of grid or from neighbor of center of grid. Then, find second minutiae, which is closest minutiae among the neighbors of the start point. Finally, link the two minutiae to form the first edge.

3. Search on the right-hand side of the first edge and find a minutiae and make them together be one triangle

4. Add triangles continuously so that the added triangles are simply connected and triangles are together.

The figures below demonstrate Delaunay triangles for one fingerprint image.

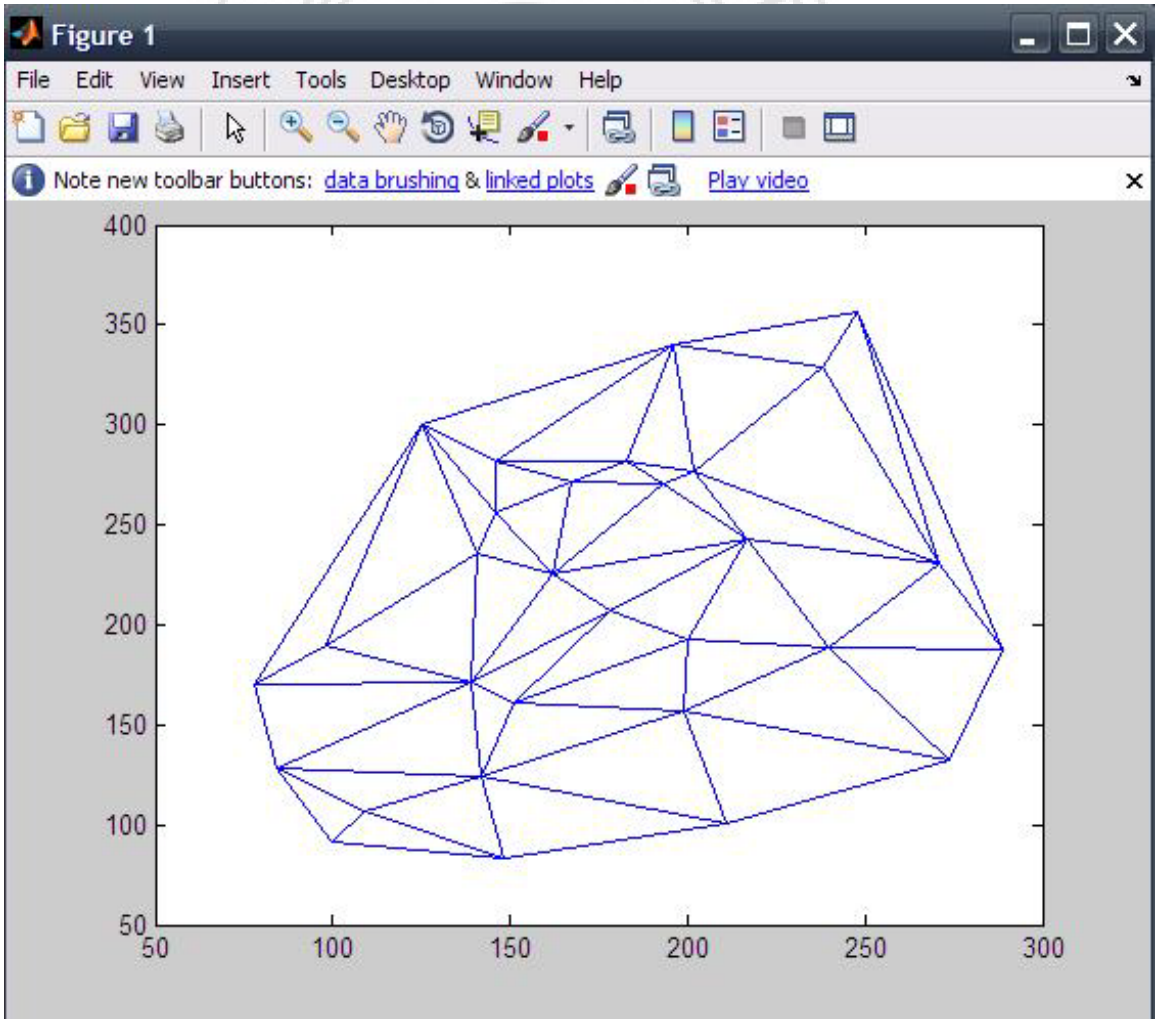


Figure 4.2

So after getting these triangles In first part of algorithm we compare three vertexes that make one triangle, and then get array of all minutiae that make triangles(specific minutiae with numbering them) the fig below shows the array of minutiae ,for example minutiae 17 ,22 and 19 these three are made one triangle .

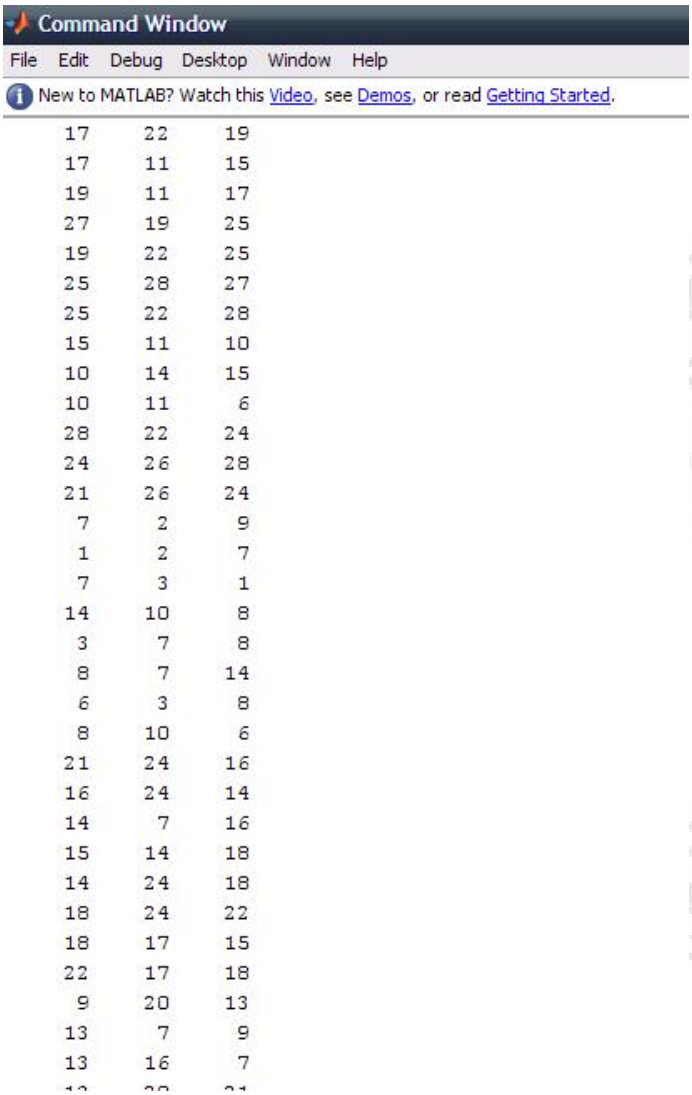


Figure 4.3

So for comparing that two fingerprint we are checking these two array that are equal or not.

If it is equal means, these two finger print is same otherwise no. the figure below shows the result of checking these two fingerprint. First one is equal the second one is not equal.

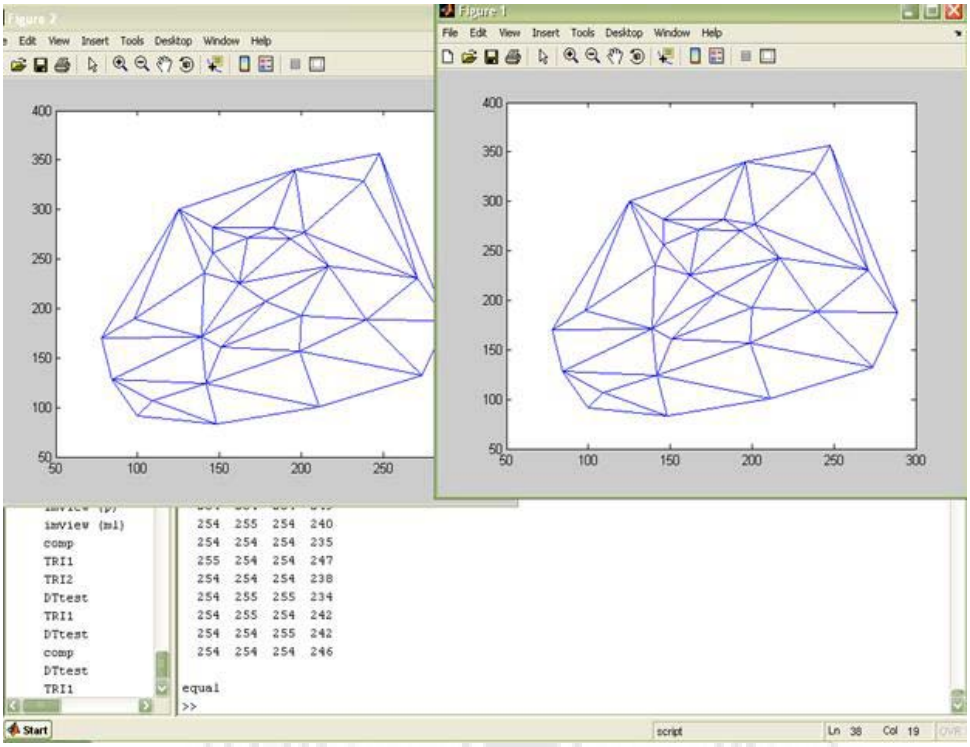


Figure 4.4

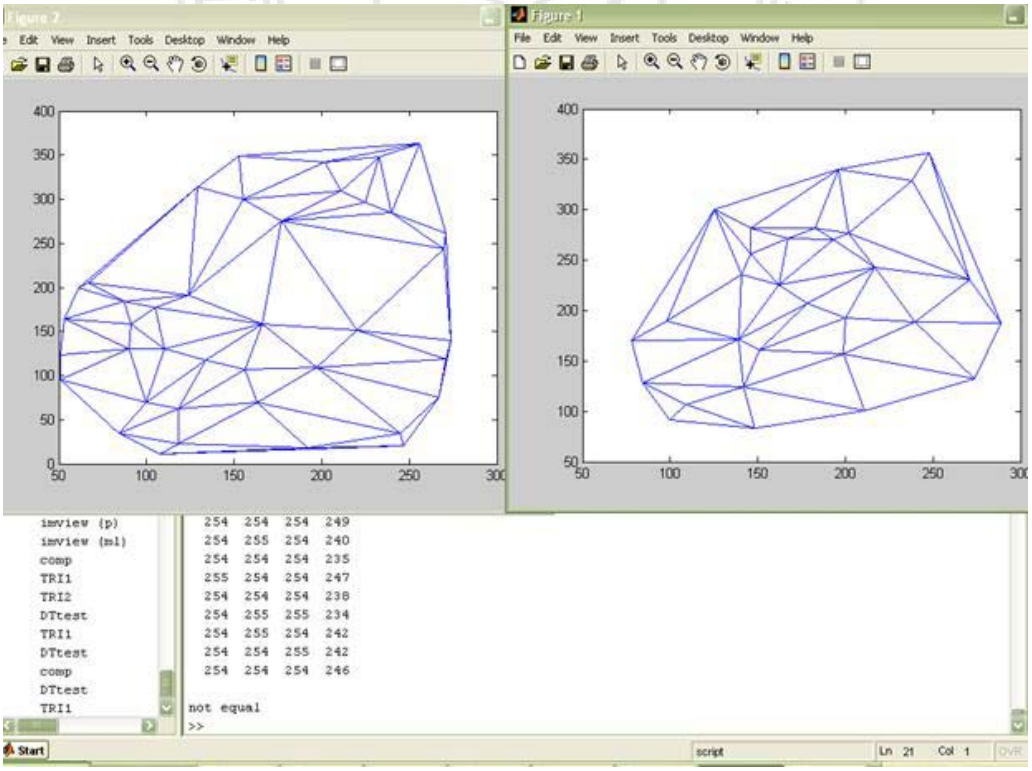


Figure 4.5

So in this algorithm we should check in each array three columns so instead of comparing three column of array we just compare one column of array so we should find the perimeters of triangles so in this way, we are just checking the 1 column in each array, so for finding the perimeters we should find the length of each triangle first by

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

So after finding the lengths try to getting the perimeter of each triages, then finding the percentage similarity of these two finger print with counting how many triangles are similar in each fingerprint. The figure below shows the result after finding the perimeters and checking two images.

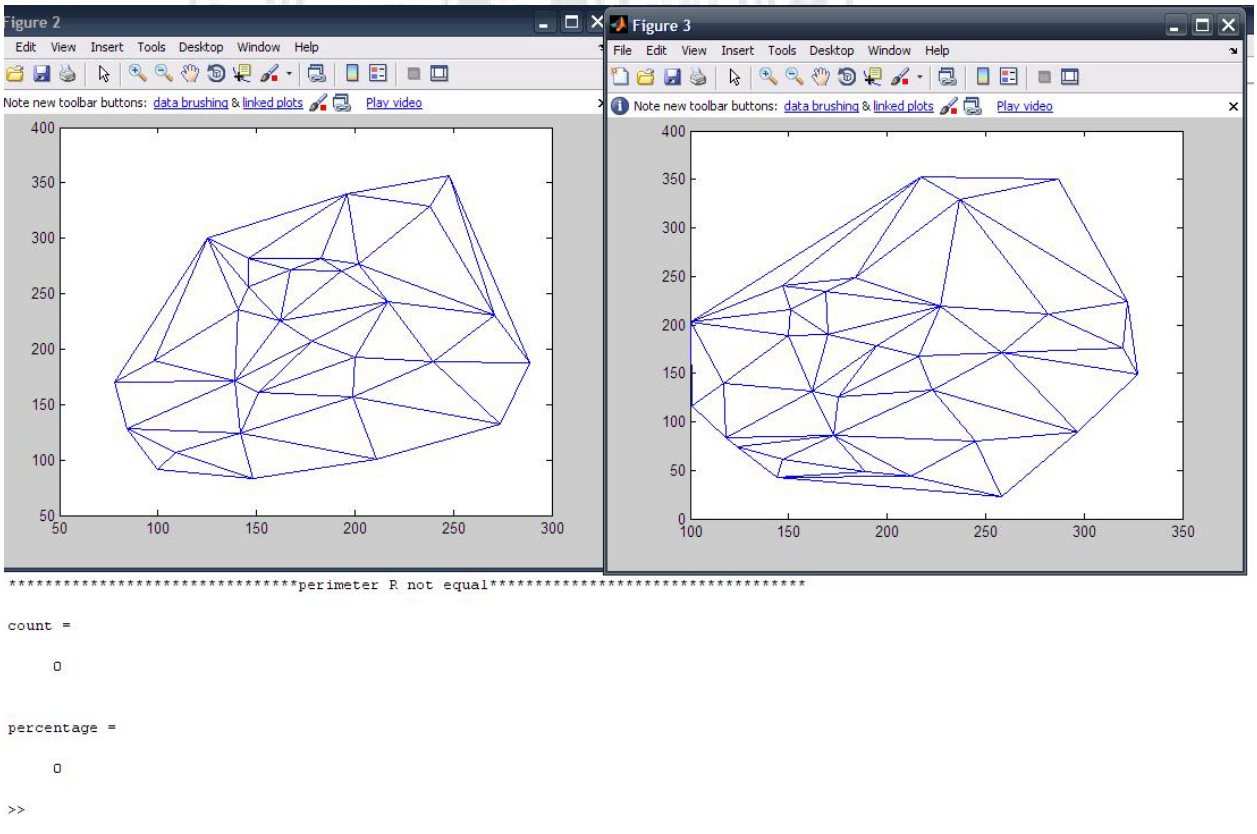


Figure 4.6

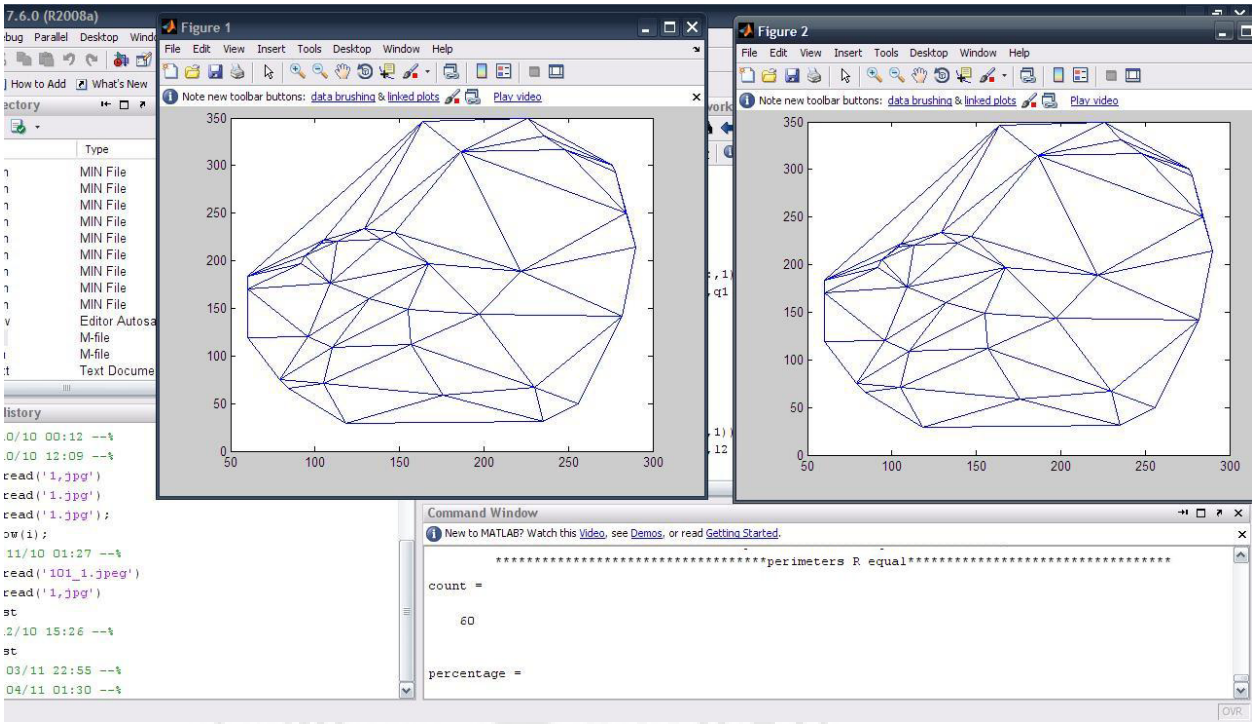


Figure 4.7

After checking the perimeters if the similarity of this two fingerprint is more than 60 percent then we check the FAR and FRR, but if it is less than 60 percent we check the edges because sometimes 2 triangles are not same but two edges out of three is same so in these case after all we checking the number of edge similarity, means we are going through the array checking for how many same edges we have.

5.EXPERIMENTATION RESULTS

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.

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.

For FAR comparing one fingerprint out of database we check how many similar one we get ,these are all false accept rate .

$$False\ Acceptance\ Rate = \frac{number\ of\ \ false\ Acceptances}{total\ number\ of\ \ attempts} \times 100\ \%$$

And for FRR in Database for each finger we have 8 image so checking first image through all then second one and go on ,and check how many are not match these are false reject rate

$$False\ Rejection\ Rate = \frac{number\ of\ \ false\ Rejections}{total\ number\ of\ \ attempts} \times 100\ \%$$

Database	Image size	FAR	FRR
FVC2002 DB1,DB2	388 x 374	0.0563	0.1648

Comparison of proposed algorithm with others

Algorithms	FAR	FRR
Delaunay Triangulation Algorithm for Fingerprint Matching	0.0018	0.0848
A fast and elastic fingerprint matching algorithm using minutiae-centered circular regions	0.026	0.021

And this is a result after checking the algorithm in database and finding FAR and FRR.

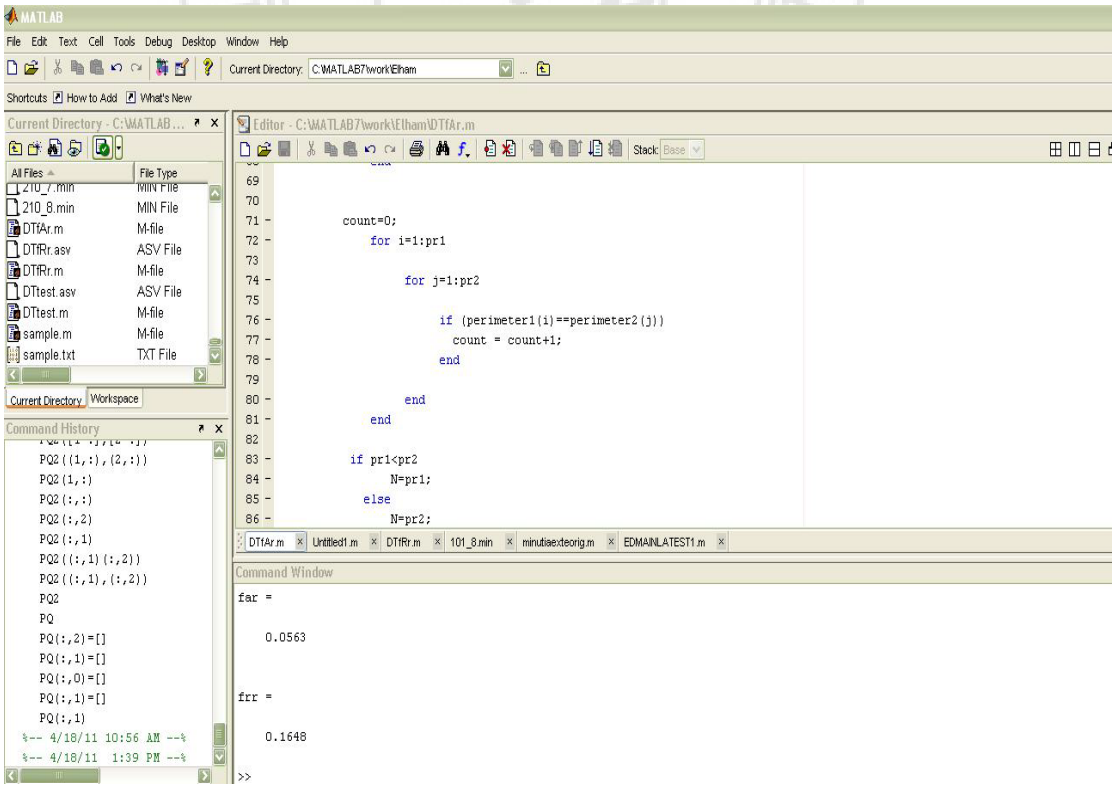


Figure 5.1

Conclusion

Fingerprint matching is the most important step in fingerprint identification, the purpose of fingerprint matching is to compare two fingerprint images and return a similarity score that represents to the probability of match between the two fingerprints so most of the fingerprint matching algorithms have been developed using minutiae based matching. In this project we proposed a novel fingerprint matching algorithm using descriptors and minutiae. For minutiae extraction we used NBIS software provided by NIST. This fingerprint matching algorithm using Delaunay Triangle.First, we extracted the minutiae and next made triangle then finding the perimeters and compare the similarity between perimeters in template and input fingerprint images and if the similarity is less it's going to compare edge similarity. The effectiveness of the proposed algorithm is tested on a public database FVC2002 DB1, DB2 and FAR, FRR is computed.

6. Bibliography

[1] Chen, H., Sun, H., & Lam, K.-Y. (2007). A fast and elastic fingerprint matching algorithm using minutiae-centered. *International Conference on Emerging Security Information, Systems and Technologies* (pp. 211-215). Washington: IEEE Computer Society .

[2] Farina, A., Kovas-Vajna, Z. M., & Leone, A. (25th june 1998). Fingerprint minutiae extraction from skeletonized binary images. *Pattern Recognition 32* , 877-889.

[3] Feng, J. (24th April 2007). Combining minutiae descriptors for fingerprint matching. *Pattern Recognition 41* , 342-352.

[4] *FVC2002*. (2002). Retrieved from http://bias.csr.unibo.it/fvc2002/download.asp

[5] Jiea, Y., fanga, Y., Renjiea, Z., & Qifab, S. (16th june 2005). Fingerprint minutiae matching algorithm for real time system. *Pattern Recognition 39* , 143 – 146.

[6] Kulkarni, J. V., Patil, B. D., & Holambe, R. S. (2006). Orientation feature for fingerprint matching. *Pattern Recognition 39* , 1551 – 1554.

[7] Liu, N., Yin, Y., & Zhang, H. (2005). A Fingerprint Matching Algorithm Based On Delaunay Triangulation Net. *The Fifth International Conference on Computer and Information Technology (CIT'05)* (p. 591). IEEE.

[8] Mohammadi, S., & Frajzadeh, A. (2009). A Matching Algorithm of Minutiae for Real Time Fingerprint Identification System. *World Academy of Science, Engineering and Technology 60* .

[9] *NBIS: fingerprint*. (n.d.). Retrieved from fingerprint web site: http://fingerprint.nist.gov/NBIS

[10] Qi, J., Yang, S., & Wang, Y. (2005). Fingerprint matching combining the global orientationfield with minutiae. *Pattern Recognition Letters 26* , 2424–2430.

[11] Tico, M., & Kuosmanen, P. (2000). Algorithm for fingerprint image post processing. *IEEE* , 1735-1739.

[12] Wang, C., & Gavrilova, M. L. (2006). Delaunay Triangulation Algorithm for Fingerprint Matching. *The 3rd International Symposium on Voronoi Diagrams in Science and Engineering (ISVD'06)* (p. 208). IEEE.

[13] Z.Li, S., & K.Jain, A. (2009). *Encyclopedia of biometrics.* Bejin China: Library of Congress Control .