VOT 74021

DESIGN AND DEVELOPMENT OF AN AUTOMATIC FINGERPRINT VERIFICATION SYSTEM

(REKABENTUK DAN PEMBANGUNAN SISTEM PENGECAMAN DAN VERIFIKASI CAP JARI)

GHAZALI BIN SULONG

FAKULTI SAINS KOMPUTER DAN SISTEM MAKLUMAT
UNIVERSITI TEKNOLOGI MALAYSIA

2005

ABSTRACT

Data security is an important part of internetworking. It prevents fraudulent users from accessing an individual personal data. Biometrics is one such authentication method used in a wide range of application domains such as e-commerce and automated banking. Biometrics is more reliable and more capable of differentiating between an authorised person and a fraudulent impostor than traditional methods such as passwords and PIN numbers. There are a number of biometrics technologies being researched and under development such as fingerprint identification, face recognition, iris recognition, etc. However, fingerprint identification is one of the most reliable biometrics technologies. Generally, there are two approaches to fingerprint identification, namely conventional and bypass. In the former approach, fingerprint images have to go through several processes including noise removal, segmentation, thinning and finally minutiae extraction. Whereas, in the latter approach, the minutiae are directly extracted from a greyscale image and bypassing all the above processes. However, the minutiae extraction is an error prone process, depending on quality of the fingerprint images. A low quality image will generate many false minutiae that eventually lead to errors in fingerprint identification. This research focuses on design and development of an automatic fingerprint verification system that capable of handling a wide variety of fingerprints. Here, the biggest challenge is to develop a technique that can enhance or improve a low quality image which contains scars, sweat spots and broken ridges. Our proposed framework is started with noise removal, and followed by image enhancement, directional image computation, fingerprint reconstruction, segmentation, thinning, minutiae extraction, and finally fingerprint matching. In this study, 500 fingerprints were tested and the percentage of successful matches was 91 percent. This achievement was directly attributable to our new enhancement technique’s excellent performance in the fingerprint reconstruction.

.

ABSTRAK

Keselamatan data merupakan isu yang amat penting dalam jaringan rangkaian komputer. Ia dapat membendung pencerobohan terhadap data individu. Biometrik merupakan satu daripada kaedah pengesahan identiti yang digunakan secara meluas dalam pelbagai bidang seperti e-dagang dan e-bank. Kaedah biometrik ini amat diyakini dan berkemampuan untuk membezakan kesahihan identiti seorang individu dengan penyamarnya. Ia adalah lebih baik daripada kaedah tradisional yang menggunakan katalaluan dan nombor PIN. Kini, terdapat beberapa teknologi biometrik yang sedang giat dikaji dan dibangunkan, termasuklah sistem pengecaman cap jari, pengecaman muka, pengecaman mata dan lain-lain. Walaubagaimanapun, pengecaman cap jari adalah lebih diyakini. Lazimnya, terdapat dua pendekatan pengecaman cap jari, iaitu kaedah konvensional dan kaedah pintas. Dalam kaedah konvensional, imej cap jari terlebih dahulu diproses melalui penghapusan hingar, dikuti oleh segmentasi, penipisan, dan akhirnya proses pengekstrakan *minutiae*. Sementara itu, bagi kaedah pintas, *munitiae* diekstrak secara langsung daripada imej cap jari berskala kelabu tanpa perlu melalui proses-proses diatas. Walaubagaimanapun, pengekstrakan *minutiae* adalah satu proses yang amat sensitif yang sering kali menghasilkan ralat, dan sangat bergantung kepada kualiti imej cap jari. Bagi imej berkualiti rendah, ianya mudah terdorong kepada penghasilan ralat dan seterusnya mengakibatkan kesilapan dalam pengecaman cap jari. Penyelidikan ini memfokus kepada rekabentuk dan pembangunan satu sistem verifikasi cap jari automatik yang berkeupayaan untuk mengendalikan pelbagai rupa bentuk cap jari. Disini, cabaran terbesar adalah untuk menghasilkan satu teknik yang mampu untuk memperbaiki dan mempertingkatkan mutu imej berkualiti rendah yang mengandungi parut, liang peluh dan batas terputus Rangkakerja yang dicadangkan adalah dimulai dengan penghapusan hingar, diikuti oleh pembaikan imej, penjanaan imej terarah, pembangunan semula imej cap jari, segmentasi, penipisan, pengekstrakan *minutiae,* and akhirnya pemadanan cap jari. Dalam kajian ini, sebanyak 500 cap jari telah diuji dengan ketepadan padanan yang terhasil adalah 91 peratus. Pencapaian ini adalah ekoran daripada prestasi cemerlang teknik baru kami dalam pembangunan semula imej cap jari.

TABLE OF CONTENTS

LIST OF APPENDICES

LIST OF TABLES

LIST OF FIGURES

We declare that this report entitled "*Design and development of an automatic fingerprint verification system*" is the result from our own research except as cited in the references.

Signature	:	.....
Name	:	Prof. Dr. Ghazali bin Sulong
Date	:	.....

To all my fellow researchers,

"I give greater respect to knowledge rather than to people, for it is they who are in need of the knowledge and it is they who should seek it."
(Imam Bukhari (R.A))

ACKNOWLEDGEMENTS

I wish to express my deep and lasting appreciation to my friends and associates who have assisted me and involved in this research, especially Mohamad Khairulli Othman; Dzulkifli Mohammed, PhD.; Jumail Taliba; Mohd Sufian Jusoh; Leong Chung Ern; Siti Masrina Sulong; and Ra Delina Patail.

I would also like to thank the Universiti Teknologi Malaysia (UTM); The Ministry of Science, Technology and Innovation (MOSTI); Research Management Centre (RMC); Fakulti Sains Komputer & Sistem Maklumat (FSKSM); Abdul Hanan Abdullah, PhD; Siti Mariam Shamsudin, PhD.; Naomie Salim, PhD.; Rose Alinda Alia, PhD.; Safaai Deris, PhD.; and Daut Daman, without whom a project of this scope could never have been completed.

CHAPTER 1

INTRODUCTION

1.1. Overview

This chapter discusses each of the topics below:

1. Background of the Problem
2. Statement of Problem
3. Objective of research
4. Scope of research

Previous and proposed framework of research will also be discussed. In the biometric sub-topic, the biometric technology based on physical characteristic and behavioural aspect such as face, fingerprint, iris, voice and signature will be covered.

1.2. Background of the Problem

Today, fingerprints are widely used as identification in many types of applications. The fingerprint identification system matches two fingerprint based on their macro-features, which are commonly recognized as fingerprint minutiae. Most of the time, these minutiae are corrupted during the scanning process and would produce false minutiae. The fingerprint can be corrupted in two ways: first, during the fingerprint scanning process; and second, fingerprints may contain scars. During the scanning process, a few factors can influence the quality of the image. These factors are pressure, moisture and the scanner quality. The corrupted fingerprint image can produce false minutiaes and probably lead to a wrong identification match.

This situation has led us to belief that using a fingerprint reconstruction technique to create a new image of higher quality without removing an original feature will lead to correct identification of individual.

1.3. Statement of Problem

This research looks into fingerprint reconstruction from original image to remove the noise during scanning process and reconstruct scar to perfect ridges and furrows. This research will also cover fingerprint pre-processing, directional image, segmentation, thinning, features extraction and fingerprint matching technique to complete the fingerprint identification system.

1.4. Objectives of the Study

This research covers various existing methodology or algorithm in each stage of the fingerprint identification system. The research starts from fingerprint pre-processing, enhancement, directional image, segmentation, thinning, feature extraction and then matching.

In this research, experiments on all stages and a comparative study with several methods in each stage were carried out.

The objectives of the study are to find the best method and suggestions based on the results of the experiments and comparative study. An automated fingerprint identification prototype from suggested techniques will also be developed.

1.5. Scope

This thesis focuses on using greyscale fingerprint images obtained through an optical scanner. The dataset will consist of 500 fingerprint images in various classes and noise.

This research involves the complete process of fingerprint identification starting from the image acquisition, pre-processing, segmentation, thinning, feature extraction and matching them with the records in the database. A one-to-many technique (1-M) during the matching stage was used.

1.6. Framework of Previous Studies

Figure 1.1 below shows a flow-chart of summarized framework using the traditional method in developing the fingerprint identification system. The traditional method of fingerprint identification system consists of seven major stages. The process starts by obtaining the fingerprint images using inked or inked-less device. The obtained fingerprint image normally contains noise, which needs to be discarded. The fingerprint quality will be enhanced during pre-processing and enhancement stage.

The identification system generates directional image from the enhanced image. Using directional image, the greyscale image is then converted into a binary image. The binary image is divided into foreground, which consists of ridges and background that consists of furrows. The binary image is then converted into skeleton images during the thinning process. Fingerprint feature extraction from the skeleton image happens during the minutiae extraction stage. Using the extracted feature during the matching stage, the fingerprint can be identified.

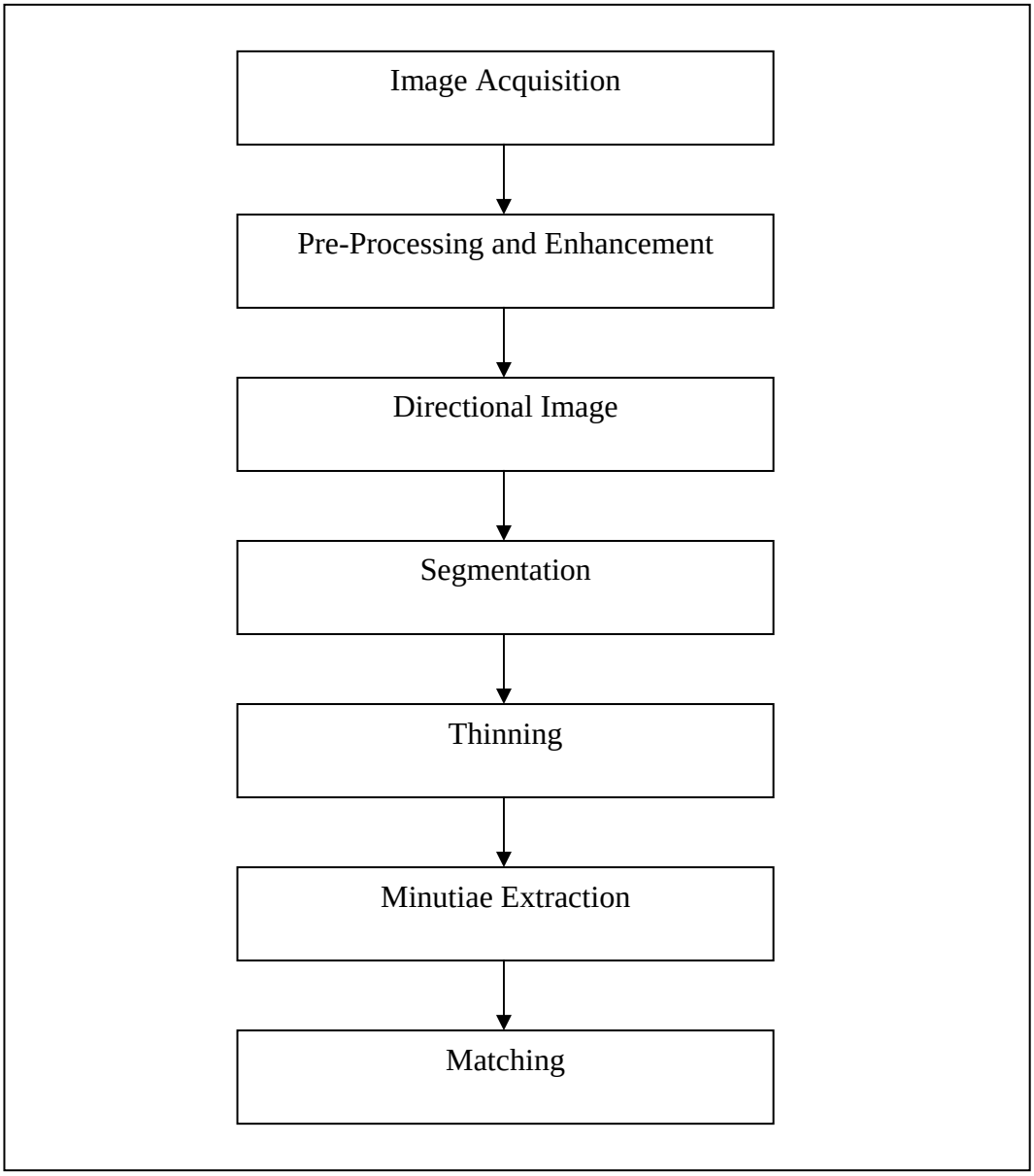


Figure 1.1 Conventional frameworks in the fingerprint identification system.

1.7. Proposed Framework

Figure 1.2 below graphically shows the framework of this research. It has ten major stages. This framework is based on the conventional framework. In this framework, three stages are added, namely: transforming original images into frequency domain, fingerprint reconstruction and median filtering. The reconstructed

image is free from noise and is better than the original image, ensuring a better fingerprint identification system.

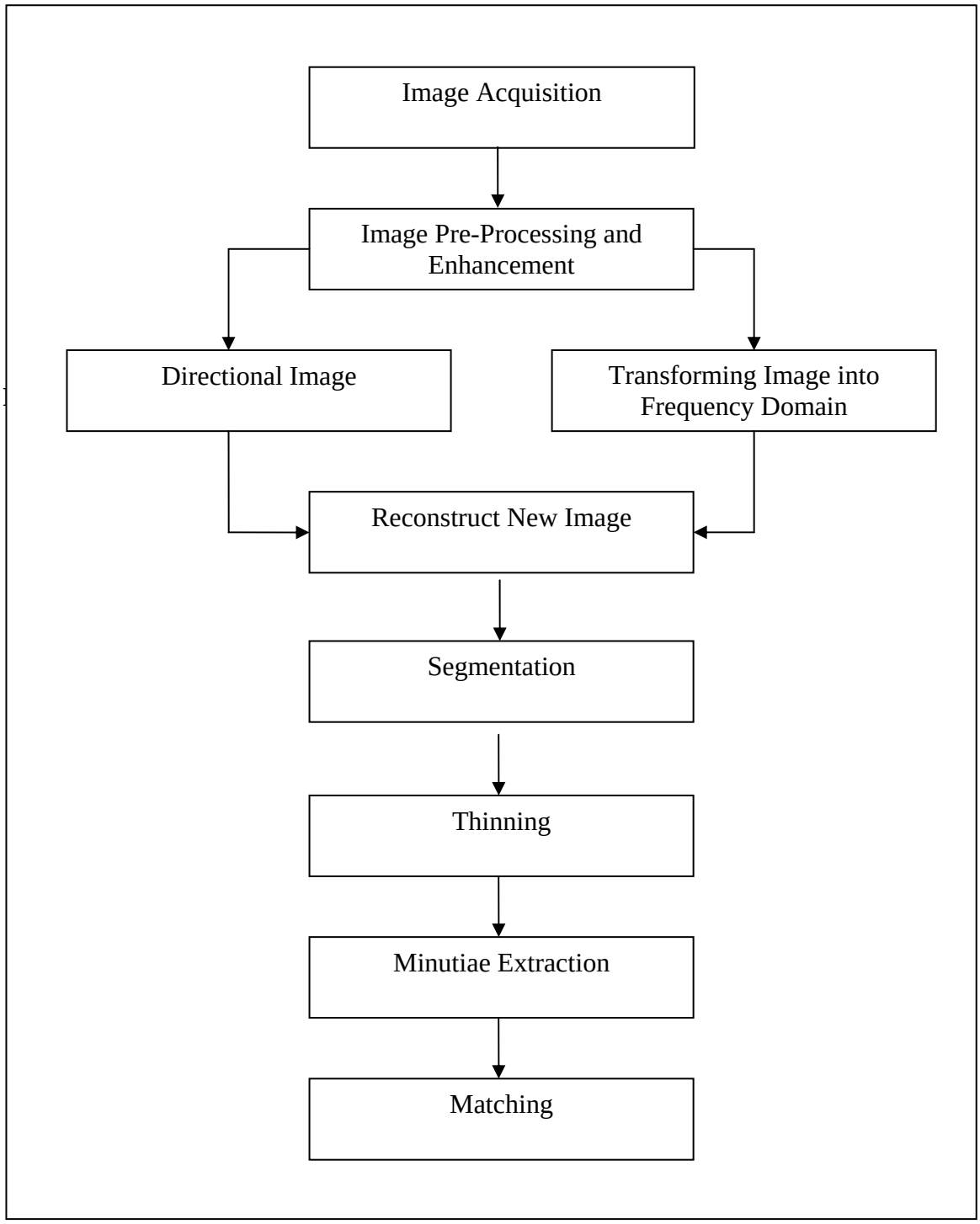


Figure 1.2: Proposed frameworks in fingerprint identification system.

1.8 Biometric: An Overview

Biometric is a science in which physical characteristics and behaviours are used to identify an individual or to verify a human identity. In other words, biometric is a unique identification, which can be used to identify a person automatically.

With biometric technology, we can verify the owner attendant when transaction occurs. Today, biometric technology spreads parallel with information technology. Biometric technologies are very important in security to control illegal transactions.

Biometric technology can be divided into two categories: i) based on physical characteristics such as face, fingerprint, iris, and DNA, or ii) based on behavioural aspects such as voice and signature.

The fingerprint method has already been used in human identification for a century. It has been successfully implemented in forensic, administrative, banking and in commerce sectors. Fingerprint contains unique feature called minutiae and it is different for every each individual.

The generated wave frequency of human voice system is unique and different for every individual. With this unique characteristic, a voice recognition system is developed. This technology is widely used in text processing and has the potential to be used in many fields. Unfortunately, when the number of samples increases, it is very hard to differentiate the frequencies because the voice frequency is very sensitive to human emotion and surrounding.

Deoxyribose Nucleic Acid (DNA) provides unique genetic information of the human cell. Linus Pauling and Robert Corey first introduced DNA in 1951. Because of the uniqueness, it has been widely used in forensic science to solve criminal cases. DNA recognition is very different from other biometric technologies. It needs true physical samples for making a comparison. To create DNA profile for each individual is very difficult and takes time. Usually DNA profiling uses hair, bone, blood or human tissue as the DNA sample. This technology is very accurate but costly and hard to used widely in commerce and security fields.

Another method of identification of individuals is through iris recognition. Iris recognition method is based on identifying the unique features of the tissues (iris) of the eyes of individual. There is usually a brown, black or blue small circle shape in the human eyes called iris. It is built from small tissues, which controls the entrance of light into the human eye. This process is similar to the aperture function in cameras. Iris map is obtained using special infrared camera. Although, this technology has high accuracy rate, it is not very practical to be used widely.

Face recognition is a process to identify unique features on human face such as the distance between the two eyes, the location of the mouth, ears, eyes, nose and face sizes. This biometric technology has a very big potential in the commercial security system. Unfortunately, to develop this system is a very difficult task. Even today, scientists are still trying to explore how the human brain identifies human. The development task becomes even more complex when dealing with natural mutable features such as moustache, hair and beard.

The acceptance of human signature as an identification system has been accepted for a long time, even though the signature technology is new and its reputation does not equal the manual signature. Recognition process is very complex because the stroke or signature shape is mutable. The accuracy of using this technology is very low and slow.

CHAPTER 2

LITERATURE REVIEW

2.1 Overview

This literature review covers the fingerprint history, fingerprint identification system and fingerprint analysis. In the fingerprint identification system, previous methods and methodology of research are discussed briefly. Different methods for each stage of the fingerprint identification system are also reviewed in this literature review section. The stages are:

1. Fingerprint pre-processing and enhancement
2. Directional image
3. Fingerprint reconstruction
4. Fingerprint segmentation
5. Fingerprint thinning
6. Minutiae extraction
7. Fingerprint matching

2.2 Fingerprint History

Humans have used fingerprints for a very long period of time (Lee et al, 1991). Human fingerprints have been discovered on a large number of archaeological artefacts and historical items. These historical artefacts provide sufficient evidence to show that ancient people used fingerprint such as in Babylon and Assyria.

In 1684, Dr. Nehemiah Grew (in Cummins et al, 1961) published a scientific paper report regarding his study on the ridge, furrow and pore structure in fingerprints. He described fingerprints structure based on his illustration. Figure 2.1 shows his illustration.

Figure 2.1 Fingerprints illustration drawn by Grew (Cummins et al, 1961)

Since then, a large number of researchers have invested a huge amount of effort on the studies of fingerprint. In 1788, a detailed description of anatomical formations of fingerprints was made by Mayer (Moenssens, 1971) in which a number of fingerprints ridge characteristics were identified and characterized (Figure 2.2).

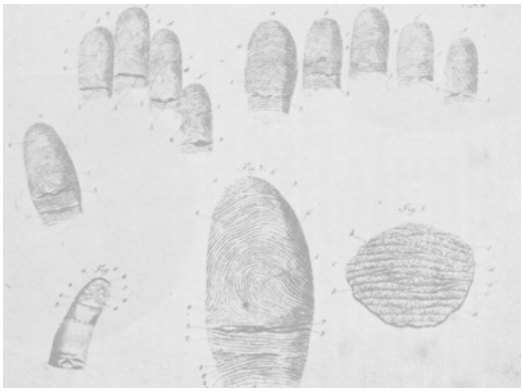


Figure 2.2 Mayer's drawing of fingerprints (Moenssens, 1971)

Starting in 1809, Thomas Bewick (Chapel, 1971) began to use his fingerprints as his trademark, which is believed to be one of the most important milestones of the scientific study of fingerprint identification. Figure 2.3 show Thomas's trademark.

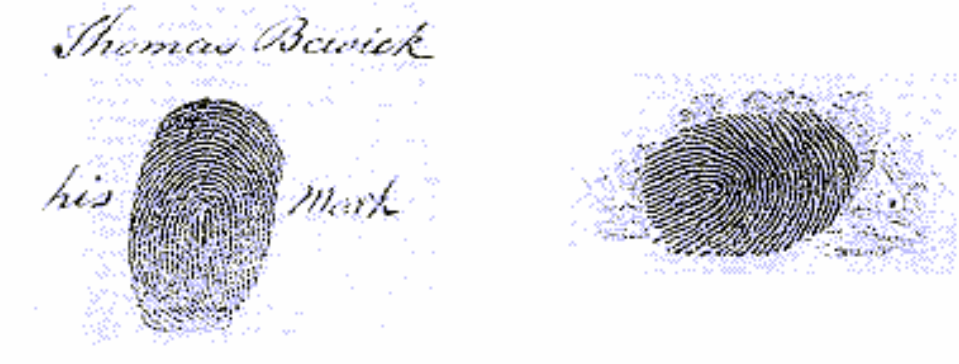


Figure 2.3 Trademarks of Thomas Bewick (Chapel, 1971)

In 1823, Purkinje (Moenssens, 1971) proposed the first fingerprint classification scheme. He classified the fingerprints into nine categories according to the ridge configurations. Figure 2.4 shows Purkinje classification.

Figure 2.4 The nine patterns illustrated in Purkinje's thesis (Moenssens, 1971)

In 1880, Henry Fauld (in Lee et. al, 1991) first scientifically suggested the individuality of fingerprints based on his observation. At the same time, Herschel ( ) asserted that he had practiced fingerprint identification for about 20 years. This discovery established the foundation of modern fingerprint identification. In the late 19th century, Sir Francis Galton (1961) conducted an extensive study of fingerprints. In 1888, he introduced the minutiae features for single fingerprint classification. In his work on fingerprinting, he named ridges characteristic known as Galton Details, which are ridge endings, bifurcations, lakes, independent ridges or islands, spurs and crossovers. He proved that every human being has different fingerprints.

Sir Edward Richard Henry (in Newham, 1995) who proposed the Henry System of Classification made an important advancement in fingerprint identification in 1899. The principles of this system are still in use today.

By the early 20th century, researchers started to understand the fingerprint formations. The biological principles for fingerprints are as follows:

1. Individual epidermal ridges and furrows have different characteristics for different fingerprints.
2. The configuration types are individually varied, but they vary within limits, which allow for a systematic classification.

3. The configurations and minutiae details of individual ridges and furrows are permanent and unchanging.

2.3 Fingerprint Analysis

The literature on the works of fingerprint features and characteristics have previously been presented in the references of Galton (1961), Henry (1905), Cherrill (1954) and the Federal Bureau of Investigation (1984).

Sir Francis Galton was the first person who undertook a substantial study of the features of fingerprints. In fact, most people refer to the features of fingerprints as Galton Details in honour of his contributions to fingerprint research. Galton defined fingerprints with their ridge endings, bifurcations, lakes, independent ridges, spurs and crossovers.

In fingerprints, there are singular points called deltas and cores. These features are very important in fingerprints classification and counting ridges. Fingerprints can be divided into several categories. The classifying system which is widely known as the Henry System who was first devised and attempted by Sir Edward Richard Henry is still being used today.

2.4 Fingerprint Features

Fingerprints contain ridges and furrows. There are two major attributes of the fingerprints. They are known as local and global attributes. For global attributes, the

cores and deltas are used for fingerprint classification. The local attributes extract fingerprint ridges information known as minutiae and are used in fingerprint identification (Kasei et al. 1997). Figure 2.5 shows the fingerprint ridges and furrows.

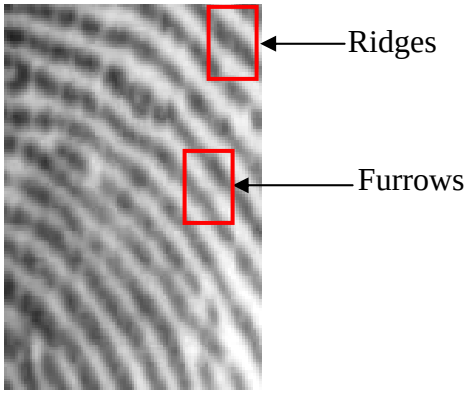


Figure 2.5 Fingerprints ridges and furrows

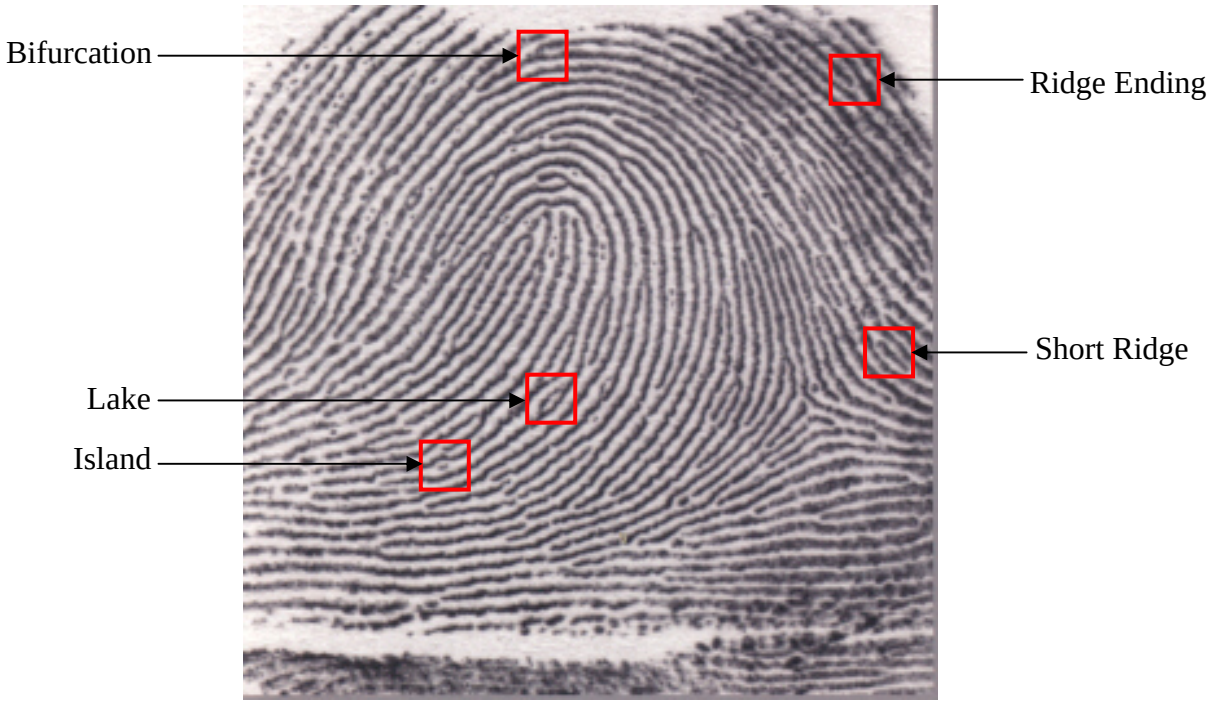


Figure 2.6 Fingerprint's minutiae

The different features of a fingerprint consist of bifurcation, lake, island, ridge ending and short ridge. A bifurcation is a ridge divided or forked into two or more parallel ridges. A lake is the joining of two bifurcations where one forms the left side and the other forms the right side. An island is a very short and independent ridge. A ridge ending is where a ridge begins and ends abruptly. A short ridge is a short and independent ridge, but not shorter than an island. From the Figure 2.6, all minutiaes are derived from a basic minutiae, ridge ending and bifurcation.

2.5 Automated Fingerprint Identification System

One of the fingerprint identification systems available is the Automated Fingerprint Identification System (AFIS). AFIS remains as one of the most prominent and reliable biometric identification method. AFIS determines whether two fingerprints are from the same finger.

There are two types of AFIS; the first type is one-to-one matching (1-1) and the second type is one-to-many matching (1-M). In a one-to-one matching environment, the person has a token such as PIN number, identity card or smart card for accessing the system. The system will match the live fingerprint with a copy within the existing database interactively. This method is faster compared to the one-to-many method.

In a one-to-many environment, the user does not have any token to identify himself. In this case, the processing time is slower because the system needs to find all the data in the database one by one.

AFIS is already in use in several law enforcement applications. However, the technology is still developing and there are still many unresolved research issues.

Automatic and reliable extraction of minutiae from a digital fingerprint image is an extremely difficult task. The performance of a current available minutiae extraction algorithm depends heavily on the quality of digital fingerprint images. Figure 2.7 shows the basic processes involved in the AFIS.

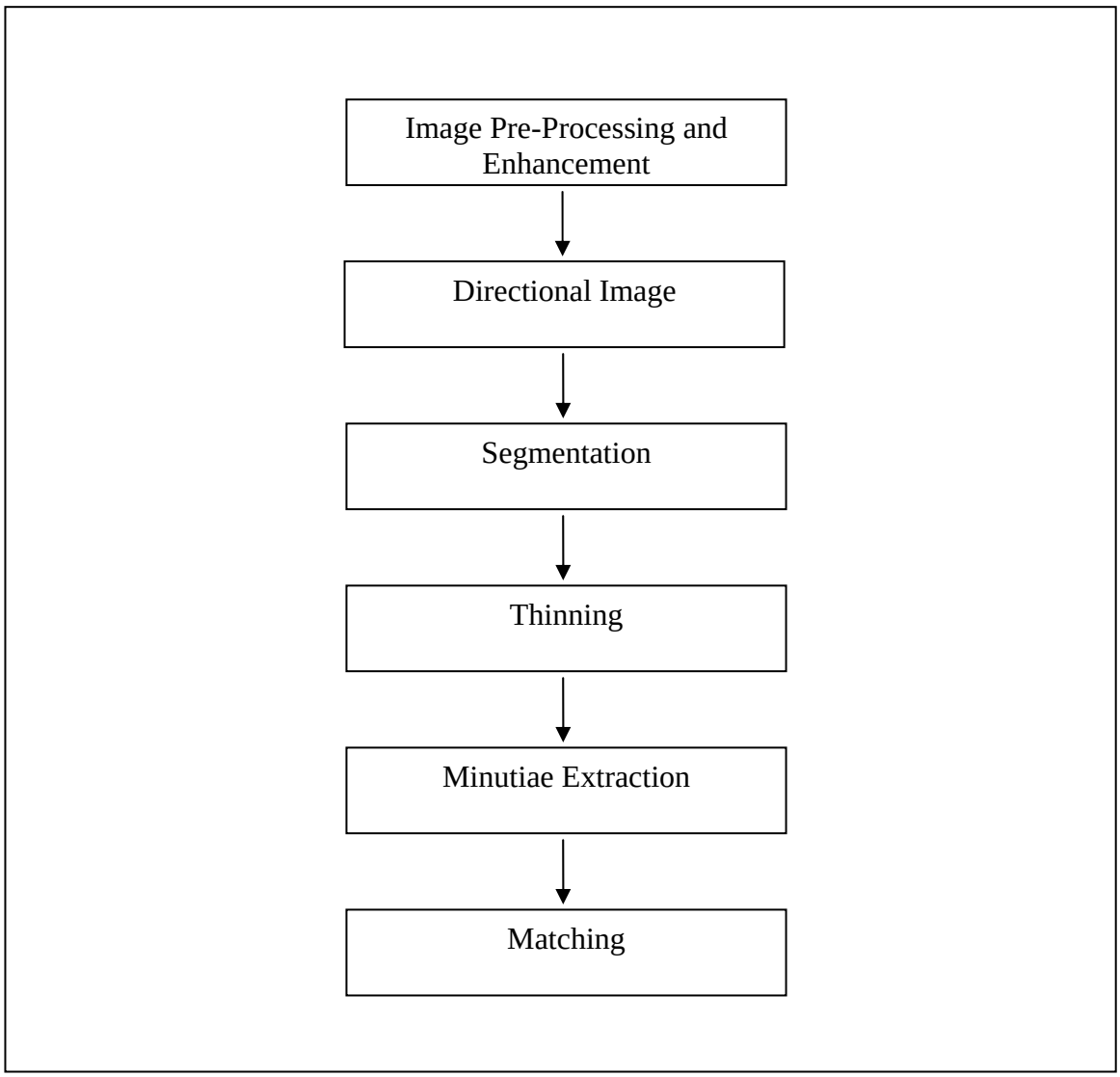


Figure 2.7 Basic processes involved in the Automated Fingerprint Identification System (AFIS)

2.6 Image Pre-Processing and Enhancement

Noise always exists in digital images. Noise verification technique in a digital image is not always accurate because of its random behaviour. Noise can damage structures. This could lead to a failure or a false minutiae extraction. There are two approaches to remove the noise and enhance the fingerprint image. They are the spatial domain approach and frequency domain approach.

2.6.1 Spatial Domain Approach

The term spatial domain refers to the aggregate of pixels in composing an image. Spatial domain methods are procedures that operate directly on these pixels. The three elements in the spatial domain approach are smoothing filter, sharpening filter and histogram modelling.

Smoothing filter is used for blurring and reducing noise. Blurring is a common pre-processing step to remove small details in an image. Examples of smoothing filter technique are average, minimum and maximum filter.

Sharpening filter is used to highlight fine details or to enhance the details of fingerprint images. A sharpening filter seeks to emphasize changes to the original images. The example of sharpening filter is high-pass and high boost filter.

Low-contrast images occur often due to poor or non-uniform lighting conditions or non-linearity or small dynamic range of the imaging sensor. One way to resolve this is to use Histogram Modelling. The histogram of an image represents the relative frequency of occurrence of the various grey levels in the image. Histogram Modelling is a technique to modify an image so that the histogram has a desired shape. This is useful in stretching the low-contrast level of images with narrow histograms (Fundamental of Digital Image Processing). Histogram equalization and histogram stretching are two techniques based on histogram modelling.

2.6.2 Frequency Domain Approach

Gonzales and Woods (1992) describe the process of enhancement in frequency domain involves the following steps: i) the Fourier transformation of image is obtained, ii) the result of transformation by a filter function is multiplied, and iii) the inverse transformation to produce the enhanced image is taken. In this domain, three methods are reviewed, namely are i) Local Fast Fourier Transformation, ii) Directional Fourier Filtering and iii) Fast Enhancement Fingerprint Algorithm.

DeLaRue Printrak Inc. (Printrak, 1985) developed a fingerprint pre-processing filter using local Fast Fourier Transform (FFT) technique. In this technique, the fingerprint image is divided into 32 by 32 tiles starting from the upper left hand corner and each tile is subsequently processed by the filter. Once a tile is processed, the filter shifts right 24 pixels to obtain the next 32 by 32 tile. In the new tile, the first 8 columns of the tiles become the last 8 columns of the previous tiles. After reaching the right side of the image, the filter shifts down 24 pixels. In the new tiles, the first 8 rows of tiles become the last 8 rows of the previous vertically adjacent tiles. The process continues until the whole image is processed in this manner. Each tile is filtered individually by an FFT. The FFT of each tile is computed, and very low and very high spatial frequencies are set to zero. Then the power spectrum of the FFT is computed as:

$$P_{jk} = X^2_{jk} + Y^2_{jk}$$

where X^2_{jk} is the real part of the FFT and Y^2_{jk} is the imaginary part of the FFT. The elements of *P* are raised to a power of α and multiplied by FFT elements $X + iY$ to obtain new elements $U + iV$ as follows:

$$U_{jk} = P^{\alpha}_{jk} X_{jk} \text{ , } \mathrm{V}_{jk} = P^{\alpha}_{jk} Y_{jk}$$

To obtain the filtered tile, the inverse FFT of $U + iV$ is computed. The real part is used to reconstruct the filtered tiles. In an image reconstruction, the center 24 x 24 pixels is saved and the outer 4 rows and columns are discarded from each filtered 32 x 32 pixel tile.

Sherlock et al (1994) first defined fingerprint enhancement by directional Fourier filtering. Sherlock used a combination of a spatial filter and a directional filter, which multiplies the spatial and the directional filters together. Sherlock defined the filter expression as $H(\rho,\phi) = H_{spatial}(\rho) * H_{angle}(\phi)$. Ikonomopoulo et al (1984, 1985) and Kunt et. al (1985) introduced a directional filtering approach for texture discrimination. In their work, they isolated the edge information belonging to a limited number of directions using a directional filter such that its frequency response covers a set of frequencies, which are within a directional range. They used $G_i = H_i * W$ as directional filter.

Hong (1998) introduced a fast fingerprint enhancement algorithm, which can improve the clarity of ridge and furrow structures of input fingerprint images based on the estimated local ridge orientation and frequency. A grey level fingerprint image, *I*, is defined as *N x N* matrix, where *I (i, j)* represents the intensity of the pixel at the *ith* row and *jth* column. It is assumed that all the images are scanned at a resolution of 500 dots per inch (dpi), recommended by FBI.

2.7 Directional Image

The directional image describes the basic shape of the fingerprint and represents an intrinsic nature of the fingerprint images. It defines the local orientation of the ridge-valley structures. The directional image takes into account the locations of ridges and valleys as well as their primary direction. Directional computation is

particularly important for fingerprint images as they consist of ridges and valleys. In a fingerprint directional image, each pixel represents the local orientation of the ridges. Here, we review Mehtre based algorithm, Iterated Least Square and Directional Image.

2.7.1 Mehtre

Mehtre and Murthy (1987) introduced the concept of directional image for fingerprints. They also investigated other images that consist of only background and foreground. In their research, they developed a segmentation method for directional images. They concluded that the directional image is an image transformation, where each pixel of the image represents the directions of the local grey level uniformity. The directional image is a transformed image of the original. It represents the local orientations of the ridges.

2.7.2 Iterated Least Mean Square Orientation Estimation Algorithm

The orientation field of a fingerprint image represents an intrinsic nature of the fingerprint image and defines invariant coordinates for ridges and furrows around each local neighbourhood, which plays a very important role in fingerprint image analysis. Hong (1998) introduced the iterated least mean square orientation estimation by viewing the fingerprint image as an oriented texture. An orientation image, O defined as N x N image, where $O(i, j)$ represents the local ridge orientation at pixel (i, j). Local ridge orientation is usually specified for block rather than at

every pixel: an image is divided into a set of *w x w* non-overlapping blocks and a single local ridge orientation is defined for each block.

2.7.3 Directional Mask

Stock and Swonger (1969) originally proposed the algorithm, which initially was used to change a grey scale image to a binary image, which has been converted into a detector of ridge orientation at each pixel. Approaches that utilise a scheme similar to Stock and Swonger's (1969) algorithm include Candela *et al*. (1995), Karu and Jain (1996), and Capelli *et al*. (1998).

Figure 2.8(a) shows an example of a direction mask used in Candela *et al*. (1995), which has a size of 9×9, centred at *C*. This mask can estimate eight directions for each pixel. The slit sums for each pixel in an image s_i, $i = 1 \ldots 8$ are computed as in the example provided. Each s_i is the sum of the value of the slit of four pixels labelled *i* in the figure. Let $0 \le p, q \le 7$ be indices such that:

$$s_p = \min_{i=0\ldots7} s_i,$$
$$s_q = \max_{i=0\ldots7} s_i.$$

The direction at a pixel defined to be *p* if the centre pixel is located on a ridge (dark area) and *q* if the centre pixel is located in a valley. If the centre pixel has *C* value, then its direction is given by:

$$d = \begin{cases} p & if \left(4C + s_p + s_q\right) > \frac{3}{8} \sum_{i=0}^{7} s_i \\ q & otherwise. \end{cases}.$$

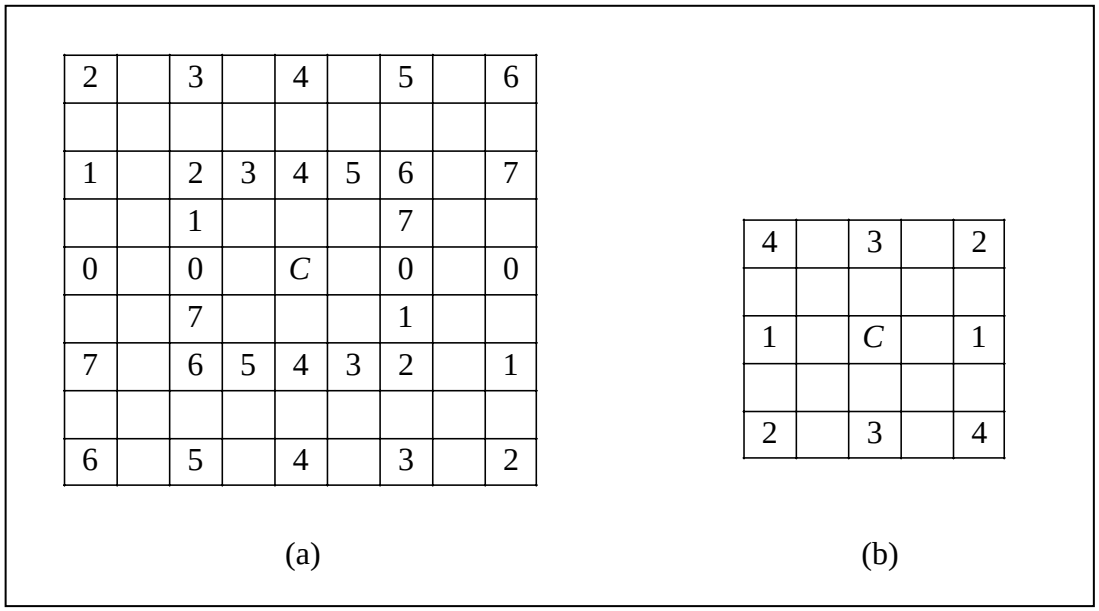


Figure 2.8 Direction masks. Each number represents an angle. (a) 8-way direction mask. (b) 4-way direction mask

2.8 Segmentation

Fingerprint segmentation is a process to separate the foreground image (ridges) from the background image (furrows) and to produce a binary image from the grey scale image. Segmentation also removes noise in the process. Segmentation is defined as a process to identify and group pixels with the same attributes in a region or block. There are a few methods to produce segmentation, such as statistical classification, thresholding, edge detection and region detection. The simplest and most recognized method of segmentation is thresholding. Thresholding involves looking at each pixel and deciding whether it should be converted into black or white pixel. This decision is made by comparing each pixel value with a constant called a threshold level or threshold value. If the pixel value is less than the threshold value,

the pixel is set to black; otherwise, it is set to white. Four techniques will be reviewed here, which are i) Global Thresholding, ii) Regional Average Thresholding, iii) Histogram Based Thresholding and iv) Niblack Binarization.

2.8.1 Global Thresholding

Global Thresholding technique is also known as Simple Thresholding. This technique is the easiest compared to the other existing techniques. It is based on one single and permanent threshold value, which is applied to all the pixels in an image. For example, if the threshold value is set to 128, then any pixel value that is less than 128 will be converted into black (or white).

2.8.2 Regional Average Thresholding

The Regional Average Thresholding (Emiroglu, 1997) divides the original image into any one of the 16 x 16, 16 x 8, 8 x 8, 8 x 4, 4 x 4 or 4 x 2 regions of windows sizes. The Regional Average Thresholding is applied on each region by using the grey level average of the related region as a threshold value.

2.8.3 Histogram-based Thresholding

Histogram-based image thresholding or segmentation is one of the simplest and most often used segmentation techniques. Histogram based segmentation depends on the histogram of the image. It uses the histogram to select the grey levels for grouping pixels into regions. There are two entities in a simple image: the background and the object. The background is generally one grey level and occupies most of the image. Therefore, its grey level forms the largest peak in the histogram.

The object or subject of the image forms another grey level, which shapes a smaller peak in the histogram. Therefore, the histogram must be designed first.

2.8.4 Niblack Binarization

Niblack Binarization algorithm is a robust thresholding in the presence of shadows and other image defects (D. Trier, 1997). The idea of this method is to vary the threshold values over the image based on local mean and standard deviation. The local mean and standard deviation are calculated and then added to the product of predefined weight constant and the standard deviation.

2.9 Thinning

An effective edge thinning algorithm is very important in an image segmentation and object identification. It increases the possibility of success by detecting the objects in the image and saves the processing time in the subsequent steps such as labelling and image transformation. The aim of the thinning algorithm is to make the fingerprint image much simpler in feature extraction. The thinning algorithm produces an image where all ridges are one pixel wide (known as skeletons). This process acquires a binary image as input. One major advantage of the thinning process is the reduction of memory space required for storing the essential structural information presented in a pattern. Moreover, it simplifies the data structure required in a pattern analysis. There are four techniques for thinning algorithm, namely are i) Safe Point Thinning Algorithm, ii) Two-Way Pass, iii) Fast Thinning Algorithm and iv) Ridge Line Following Thinning Algorithm.

Connectivity is an important property that must be preserved in the thinned object. Border pixels are removed in such a way that the object connectivity is still maintained. Thinning algorithms satisfy the following constrains:

1. They maintain connectivity at iteration. They do not remove border pixels that may cause discontinuities.
2. They do not shorten the end of thinned shape limbs.

2.9.1 Safe Point Thinning Algorithm

Nachacce and Shinggal (1984) proposed the Safe Point Thinning Algorithm (SPTA) technique, which consists of executing many passes over the pattern, where in each pass, a few dark points are flagged. A flagged point is an edge-point, but not an end-point, nor a break--point and nor must its deletion cause excessive erosion in the pattern. Edge-point is a dark point that has at least one white 4-neighbour. On the other hand, end-point is a dark point that has at most one dark 8-neighbour. Break-point is a dark point, where the deletion which would break the connectivity of the original pattern.

SPTA technique must satisfy the different conditions in order to delete certain points. This depends on the following conditions:

1. an edge-point,
2. an end-point, and
3. it satisfies the pre-defined template match windows (break-point).

SPTA has a few steps as follows:

1. Test whether a point is an edge-point or not.
2. Test whether an edge-point is an end-point or not,
3. Test whether its deletion will cause disconnectedness or not (if not, flag it).
4. At the end of each pass, delete all flagged points.

For step one, SPTA tries to identify an edge-point as one or more of the following four types:

1. A left edge-point having a white left neighbour,

2. A right edge-point having a white right neighbour,
3. A bottom edge-point having a white bottom neighbour,
4. A top edge-point having a white top neighbour.

Here, an edge-point can be more than one type. A pass in the SPTA consists of two scans instead of one single scan for each pass, where a scan examines every point in the pattern. In the first scan, all left edge-points and all right edge-points that are not respectively left safe-points and right safe-points are flagged. Similarly, in the second scan, the corresponding top edge-points and bottom edge-points are also flagged.

2.9.2 Two-way Pass

The Two-way Pass technique requires two successive iterative passes which are described in Zha and Gon (in Gonzalez, 1992). In step 1, a logical rule P_1 is applied locally in a 3x3 neighbourhood to flag border pixels that can be deleted. These pixels are only flagged, and not deleted, until the entire image is scanned. Deletion of all flagged pixels is performed afterwards. In Step 2, another logical rule P_2 is applied locally in a 3x3 window to flag border pixels for deletion. After the entire image has been scanned, the flagged pixels are then deleted. This procedure is applied iteratively, until no more thinning can be performed.

2.9.3 Fast Thinning Algorithm

The main objective of using the Fast Thinning Algorithm (FTA) is to obtain the skeleton of an image. There are two main steps in the FTA that are repeated until the obtained image approaches the medium axis of the original image. First, the

contour of the image is marked. The contour of an image is formed by a pixel that is found in the innermost and most distant position of this image. In the second step, the marked contour is analyzed to verify which pixels belonging to this contour that should be deleted.

2.9.4 Ridge Line Following Algorithm

Emiroglu (1998) proposed the thinning algorithm based on the Ridge Line Following Algorithm, which is used only on threshold fingerprint images. The algorithm uses black pixels as the ridges of the fingerprints. The thinning algorithm presented here is designed particularly for fingerprint images. Before the thinning operation starts, it is necessary to obtain a block directional image which produces the direction of each pixel value in a fingerprint image. The aim of the thinning algorithm is to remove redundant black pixels in the image and to produce a thinned image.

2.10 Minutiae Extraction

Fingerprint feature extraction is also known as minutiae extraction. In this stage, the minutiae are extracted from the thinned fingerprint images. In this minutiae extraction stage, only the bifurcations and ridge endings known as basic or primitives from thinned images. Other minutiae such as lakes, pores and hooks are disregarded since they are essentially a combination of these basic minutiae. Each extracted minutiae have four attributes, which are i) x-coordinate, ii) y-coordinate, iii) minutiae direction and iv) minutiae type. There are two methods mainly used for fingerprint feature extraction purpose: Crossing Number and Template.

2.10.1 Crossing Number

The Crossing Number (CN) determines the ridge ending and ridge bifurcation using the equation below (Mehtre, 1993).

$$CN = 0.5\sum_{i=1}^{8}\left|p_i - p_{i+1}\right|, \quad p_9 = p_1$$

These technique works on a 3x3 regions. With the value of CN, the property of CN is determined. CN has three major properties: i) ridge ending, ii) ridge bifurcation, and iii) connecting ridges.

2.10.2 Template

Emiroglu (1998) and Hong (1998) used a template in minutiae extraction. Using template or mask window of size 3x3 pixels centered at the black pixel, the algorithm finds the number of pixels, *N*, within the window. The value *N* will determine the features of the minutiae.

2.11 Matching

Given two minutiae patterns, an input and a template, the minutiae matching algorithm determines whether they are from the impressions of the same finger. There are three approaches for this purpose, which are Template Matching, Alignment Based Algorithm, and The Flexible Matching Algorithm (The Matcher).

2.11.1 Template Matching

The template matching algorithm attempts to match a set of minutiae obtained from a scanned fingerprint with a previously stored template. In this algorithm, the classic ridge counting is not performed as this also increases the possibility of false rejection without seriously affecting the false acceptance rate. The matching is based on the minutiae features (Sun et al, 1996).

2.11.2 Alignment Based Algorithm

Hong (1998) developed an alignment-based matching algorithm, which is simple in theory, efficient in discrimination and fast in speed. The algorithm decomposes the minutiae matching into two stages: (i) alignment stage and (ii) matching stage. In the alignment stage, an alignment hypothesis, including translation and rotation between the input and the template are first generated and the input minutiae are aligned with the template minutiae according to the hypothesis. In the matching stage, the input minutiae and the template minutiae are first converted to a string representation in the polar coordinate system. An elastic string matching algorithm is used to evaluate the similarity between the two strings. The hypothesis that results in the largest similarity value is determined as the optimal alignment. The corresponding minutiae pairs are determined based on this optimal alignment.

2.11.3 The Flexible Matching Algorithm (The Matcher)

A flexible matching algorithm approach was described by Akhan and Emiroglu (1995a, 1995b). This matching algorithm allows the adjustment of the degree of matches. The extracted features from the fingerprint image, such as location and angle are allowed to deviate by user definable margins. The algorithm offers likely matches to the extracted features of the compressed database using pre-set margins.

CHAPTER 3

METHODOLOGY

3.1 Overview

This chapter provides detailed discussion and explanation of selected methods from the literature review. In this chapter, the discussion goes through each stage of the fingerprint identification process. The discussion involves fingerprint pre-processing and enhancement, directional image, fingerprint reconstruction, fingerprint segmentation, fingerprint thinning, fingerprint minutiae extraction and fingerprint matching.

3.2 Fingerprint Pre-Processing and Enhancement

Noise effect always exists in digital images. The noise in digital images cannot be verified 100% because of its random behaviour. In fingerprint images, ridges and furrows are very sensitive, and the distance between them are very small.

Noise damages the structures of images, which could lead to a failure or false minutiae extraction.

Filter processing is used to remove the noise and enhance the fingerprint images. In this stage, fingerprint images are filtered using the spatial domain technique. The use of spatial mask for image processing is usually called spatial filtering. The mask itself is called spatial filter. There are a few techniques to remove noise, such as sharpening filter, smoothing filter and histogram modelling. Figure 3.1 shows the original fingerprint image that is used for fingerprint pre-processing and enhancement.

Figure 3.1 Original fingerprint image

3.2.1 Smoothing Filter

Smoothing filter is used for blurring and noise reduction. Blurring is a common pre-processing step to remove small details when the aim is the location of a large object. The smoothing filter technique involves:

1. Averaging
2. Minimum

3. Median
4. Maximum
5. Low Pass
6. Hexagonal Grid

In averaging filtering, centered pixels in spatial mask are replaced with the average value of spatial mask. The sizes of spatial mask are usually 3x3, 5x5 and 9x9. Figure 3.2 shows the original pixel values before averaging filter.

64	64	64
64	255	255
64	64	255

Figure 3.2 Original pixel values before averaging filter

The new value obtained for the center pixel using 3x3 average filters is:

$$(64+64+64+64+255+255+64+64+255)/9=128$$

Figure 3.3 shows the new pixel value and new fingerprint image after the average value is obtained.

64	64	64
64	**128**	255
64	64	255

Figure 3.3 New pixels values and new fingerprint image after the average value is obtained.

Median Filter may be used when the aim is to achieve noise reduction with a minimum amount of blurring. This means that the grey level of each pixel is replaced by the median of the grey levels in a neighbourhood of that pixel, instead of by averaging. Median Filter method is particularly effective when the noise pattern consists of strong and spike like components and the characteristic to be preserved is edge sharpness. In order to perform median filtering in a neighbourhood of a pixel, the values of the pixel and its neighbours are sorted first, the median is determined, and the value to the pixel is assigned. For example in a 3x3 neighbourhood, the median is the 5^{th} largest value; while in a 5x5 neighbourhood, the median is the 13^{th} largest value, and so on. In Figure 5.2, the 3x3 neighbourhood has a value of 64, 64, 64, 64, 255, 255, 64, 64, and 255. These values are sorted as 64, 64, 64, 64, **64**, 64, 255, 255, 255, which result in a median of 64. Figure 3.4 shows the pixel values and new fingerprint image after median filtering.

4	64	64
64	**64**	255
64	64	255

Figure 3.4 Pixel values and the new fingerprint image after median filtering

Minimum filtering has the same process as median filtering. Median filter determines the median value within a neighbourhood while the minimum filter determines the first value in the neighbourhood. For example, in Figure 3.2, the 3x3 neighbourhood has a value of (64, 64, 64, 64, 255, 255, 64, 64, 255) and after sorting, this neighbourhood has a value of (**64**, 64, 64, 64, 64, 64, 255, 255, 255) and the minimum value is 64 (Figure 3.5). Maximum filtering will also go through the similar process. However, this technique only determines the last value in the neighbourhood. For example, the sorted 3x3 neighbourhood, the maximum value is 255 (Figure 3.6).

64	64	64
64	**64**	255
64	64	255

Figure 3.5 Pixel values and the new fingerprint image after minimum filtering

64	64	64
64	**255**	255
64	64	255

Figure 3.6 Pixel values and the new fingerprint image after maximum filtering

Low-Pass filtering allows low spatial frequency to pass unchanged but high frequency is suppressed. Low-Pass produces blurred image and reduces noise but

obscure fine details. Low-Pass is suitable for images where the noise has strong high frequency component. Figure 3.7 shows the mask filter for Low-Pass filtering.

1/16

1	2	1
2	4	2
1	2	1

Figure 3.7 Low-Pass filter masks

For example, filter 3x3-neighbourhood value (64, 64, 64, 64, 255, 255, 64, 64, and 255) with Low-Pass mask:

64*1+64*2+64*1+64*2+255*4+255*2+64*1+64*2+255*1=144

Figure 3.8 shows the pixel values and the new fingerprint image after Low-Pass filtering.

64	64	64
64	**144**	255
64	64	255

Figure 3.8 Pixel values and the new fingerprint image after Low-Pass filtering

3.2.2 Sharpening Filter

Sharpening filter is used for highlighting fine details or enhancing the details of a fingerprint image. A sharpening filter seeks to emphasize changes. The sharpening techniques are:

1. High Pass
2. High Boost

High pass filtering is accomplished using a kernel containing a mixture of positive and negative coefficients. An omni-directional high pass filter, specifically one whose response is the same in whatever the direction in which grey level varies, should have positive coefficients near its centre and negative coefficients in the periphery of the kernel. Figure 3.9 shows the classic implementation of 3x3 sharpening filter.

1/9

-1	-1	-1
-1	8	-1
-1	-1	-1

Figure 3.9 A High-Pass spatial mask filter

Note that the sum of the coefficients is 0. This means that when the kernel is over an area of constant or slowly varying grey level, the result of convolution is zero or some very small number. However, when the grey level varies rapidly within the neighbourhood, the result of convolution can be a large number. This number can be positive or negative, because the kernel contains both positive and negative coefficients. Figure 3.10 shows the fingerprint image after High Pass filter.

Figure 3.10 Fingerprint image after applying the High Pass filter

Using a high boost approach, we can compute a weighted sum of the original image and the output from a high pass filter. The result is an image in which high spatial frequencies are emphasised relative to lower frequencies. The degree of emphasis achieved depends on the weight given to the original and high pass filtered images. This high boost filter can be used to sharpen an image. Be informed that high boost filtering in a single convolution operation can be performed using the kernel shown in Figure 3.11.

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & c & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad (c > 8)$$

Figure 3.11 A High Boost spatial filter

When the central coefficient *c* is large, the convolution will have little effect on an image. As *c* gets closer to 8, the degree of sharpness increases. If *c* = 8, the kernel becomes the high pass filter as described earlier in Figure 3.9. Figure 3.12 shows the fingerprint image after applying the High-Boost filter where c=9.

Figure 3.12 Fingerprint image after applying High Boost Filter where c=9

3.2.3 Histogram Equalization

Histogram Equalization is a technique to modify an image so that its histogram has a desired shape. This is useful in stretching the low-contrast levels of images with narrow histograms.

A useful approach to digital image processing is to consider image intensities $f(i, j)$ as being random variables having probability density function

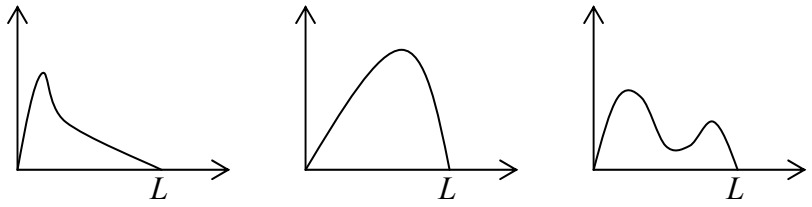
(pdf) $p_f(f)$. Pdf image carries valuable global information about image content. However, the pdf is generally not available and must be estimated from the image itself by using the empirical pdf, usually called the histogram. Let us assume that a digital image has L discrete grey levels (usually from 0 to 255) and that $n_k, k = 0,...,L-1$ is the number of pixels having k intensity. The histogram $\hat{p}_f(f)$ is given by the equation below:

$$\hat{p}_f(f) = \frac{n_k}{n} \qquad k = 0,1,...,L-1$$

(**Equation 3.1**)

where n is the total number of image pixels. The image histogram can be calculated easily, as can be seen in Equation 3.1.

The image histogram carries important information about the image content. If its pixel values are concentrated in the low image intensities, as in Figure 3.13(a), the image is *dark*. A *bright* image has a histogram that can concentrate in the high image intensities, shown in Figure 3.13(b). The histogram of Figure 3.13(c) reveals that the image contains two objects with different intensities (or possibly one object being clearly distinguished from its background). If the image histogram is concentrated on a small intensity region, the image contrast is poor and the subjective image quality is low. Image quality can be enhanced by modifying its histogram. This can be performed by a technique called *histogram-equalization*.



(a) (b) (c)

Figure 3.13 (a) Histogram of dark image; (b) histogram of bright image; (c) histogram of an image containing two regions with different distributions

Histogram equalization is a technique where the histogram of the resultant image is as flat as possible. With the histogram stretching, the overall shape of the histogram remains the same. The theoretical basis for histogram equalization involves probability theory where the histogram is treated as the probability distribution of the grey levels. This technique consists of four steps:

1. Finding the running sum of the histogram values.
2. Normalizing the values from step (1) by dividing by the total number of pixels.
3. Multiplying the values from step (2) by maximum grey level value and round.
4. Mapping the grey level values to the results from step (3) using a one to one correspondence.

Example: Given a 3 bit image, the possible range of values is 0 to 7. Suppose the image has the following histogram:

Table 3.1 Histogram of image

Grey Level	0	1	2	3	4	5	6	7
No of Pixels	10	8	9	2	14	1	5	2

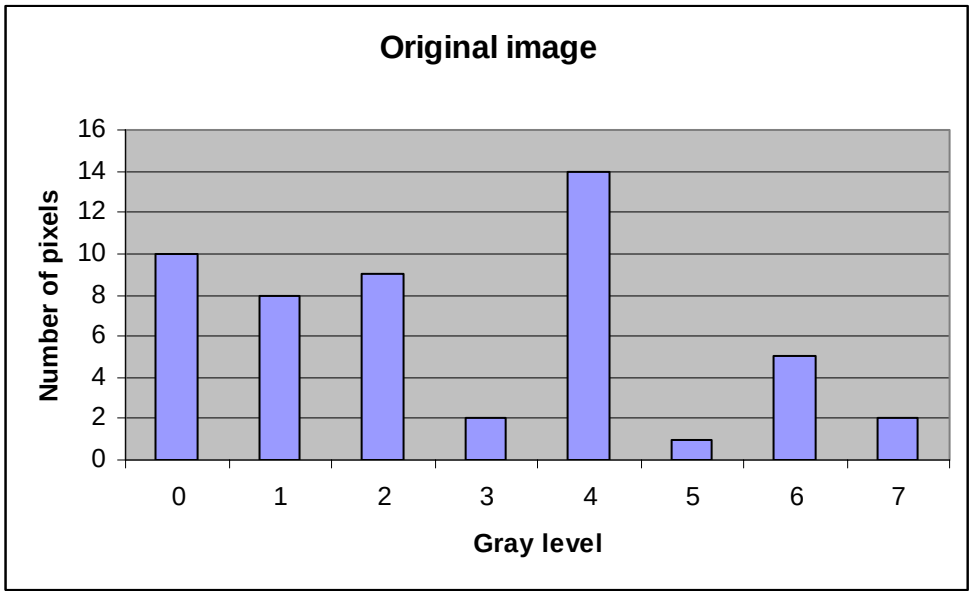


Figure 3.14 Histogram of original image.

The following Table 3.2 shows the steps to find the histogram equalized values.

Table 3.2 Histogram equalized values

Grey Level	0	1	2	3	4	5	6	7
No of Pixels	10	8	9	2	14	1	5	2
Run Sum	10	18	27	29	43	44	49	51
Normalized	10/51	18/51	27/51	29/51	43/51	44/51	49/51	51/51
Multiply 7	1	2	4	4	6	6	7	7

In order to get the histogram equalized image, all the pixels in the original image with grey level 0 are set to 1, value of 1 are set to 2, 2 set to 4, 4 set to 4 and so on; as indicated in Table 3.3. Figure 3.15 shows the histogram of the histogram equalized image.

Table 3.3 Histogram of the histogram equalized image

Grey Level	0	1	2	3	4	5	6	7
No of Pixels	0	10	8	0	11	0	15	7

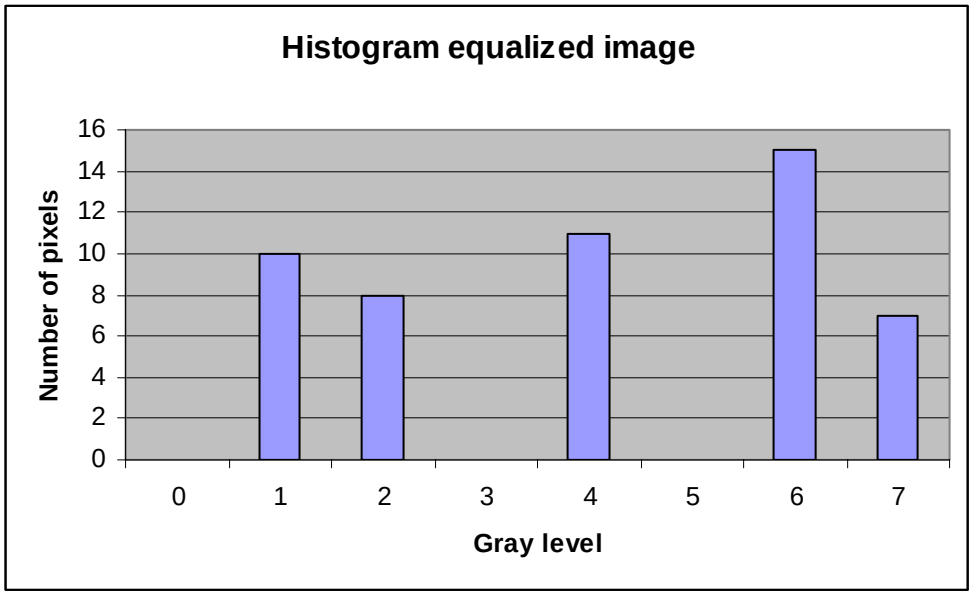


Figure 3.15 Histogram of the histogram equalized image

Figure 3.16 shows a histogram before applying histogram equalization. Figure 3.17 shows the histogram after applying the histogram equalization, and Figure 3.18 displays the fingerprint image after the histogram equalization.

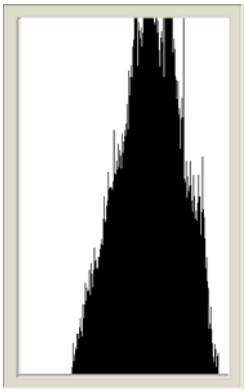


Figure 3.16 Original Histogram

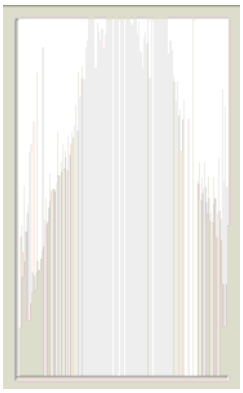


Figure 3.17 The histogram after histogram equalization

Figure 3.18 Fingerprint image after histogram equalization

3.3 Directional Image

The directional image describes the basic shape of the fingerprint and represents an intrinsic nature of the fingerprint image. It is defined as the local orientation of the ridge-valley structures. The directional or orientation field defines invariant coordinates for ridges and furrows around each local neighbourhood, which plays a very important role in the fingerprint image analysis (Hong, 1999). By viewing a fingerprint image as an oriented texture, a number of methods have been proposed to estimate the orientation field of fingerprint images (Kass & Witkin, 1987), (Kawagoe & Tojo, 1984). The directional image takes into account the

locations of ridges and valleys as well as their primary direction. Directional computational is particularly important for fingerprint images as the fingerprint image consists of ridges and valleys. In a directional fingerprint image, each pixel represents the local orientation of the ridges. The directional image can be used for image filtering, segmentation and classification. In this research, we used directional image for image enhancement and reconstruction. To produce the directional image two methods were utilised: i) Mehtre based technique and ii) Iterated Least Mean Square Orientation Estimation Algorithm.

3.3.1 Mehtre

Mehtre et al (1987) developed the concept of directional image computation as follows:

$$K(i,j) = Min\sum_{k=1}^{n} | C(i,j) - C_d(i_k, j_k) | \qquad \text{d=0,1,2,3,...N-1}$$

(**Equation 3.2**)

where *K(i, j)* is an output image at location (i, j). *C(i, j)* is grey level value for pixel (i, j). $C_d(i_k,j_k)$, is grey level value of input image for each pixel in a particular direction. *N* is the number of directions and *n* is the number of pixels chosen for this particular direction, *d*. The directional image computation is graphically summarized in Figure 3.19.

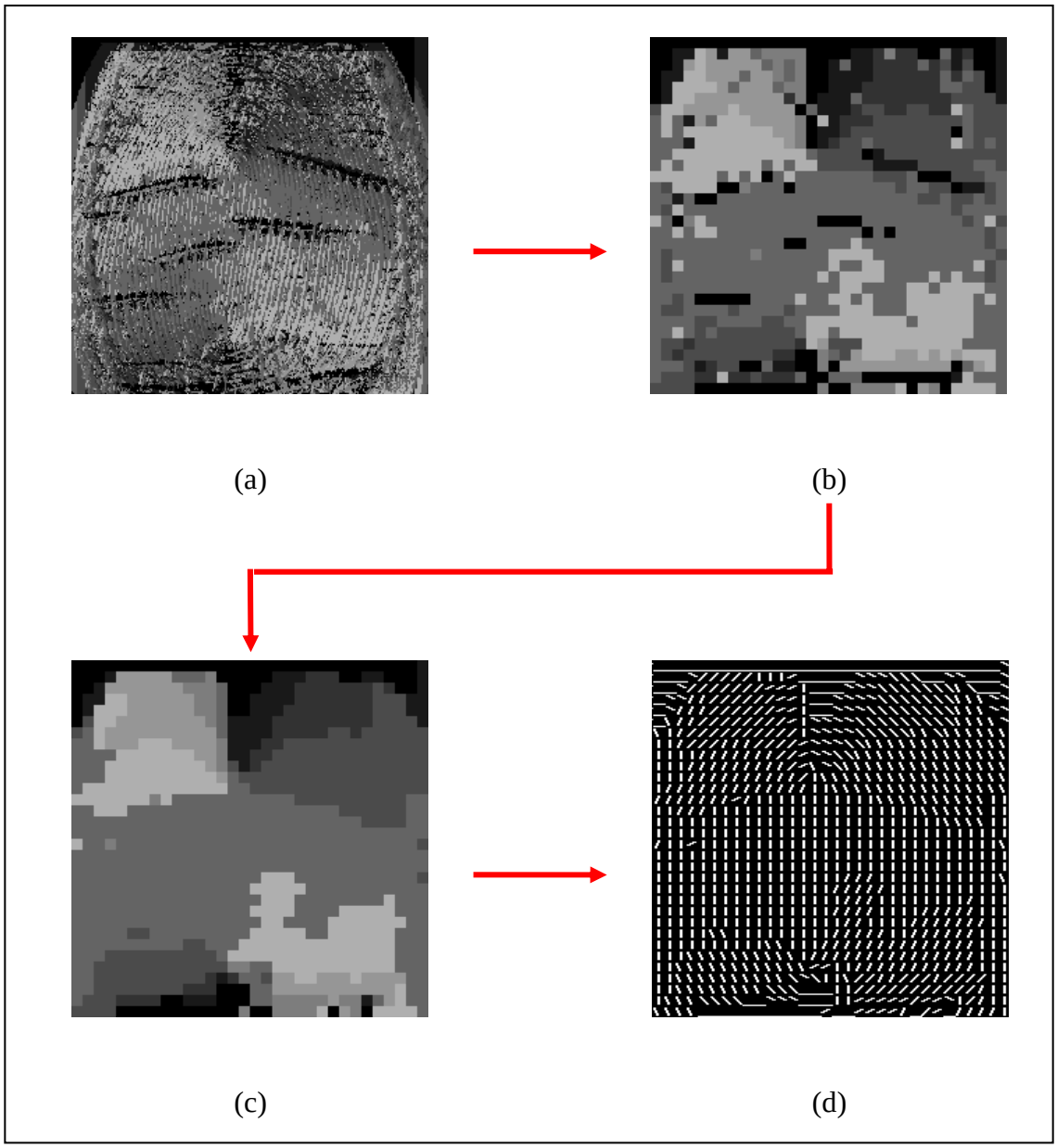


Figure 3.19 Directional image generation summarized graphically

Firstly, using Equation 3.2, an output image in Figure 3.19(a) is obtained. In this stage, the direction for each pixel is computed. This process is known as the "pixel-wise directional image". To make the filter less computationally intensive, the pixels are grouped into small region or block, either 4x4, 8x8, 16x16 or 32x32. For each region, a new direction is calculated. The new direction is obtained by using histogram given by Equation 3.1. In each region, the direction which occurs the maximum number of times or maximum value of histogram is chosen as the new direction for the region. This process is known as "block directional image". Figure

3.19 (b) shows the new direction obtained from “pixel-wise directional image”. Using the median filtering technique, the noisy regions are removed and a new directional block image which is free from noise is obtained. Figure 3.19 (c) shows the new block directional image after image filtering. Figure 3.19 (d) shows the final output or fingerprint directional image.

For the purpose of this study, the directions of the fingerprint have been computed in N=8 directions as shown in Figure 3.20, and for each direction, n=10 pixel value have been used. In this study, experiments were conducted using 4x4, 8x8, 16x16 and 32x32 sized blocks to find the best region size to produce the directional image.

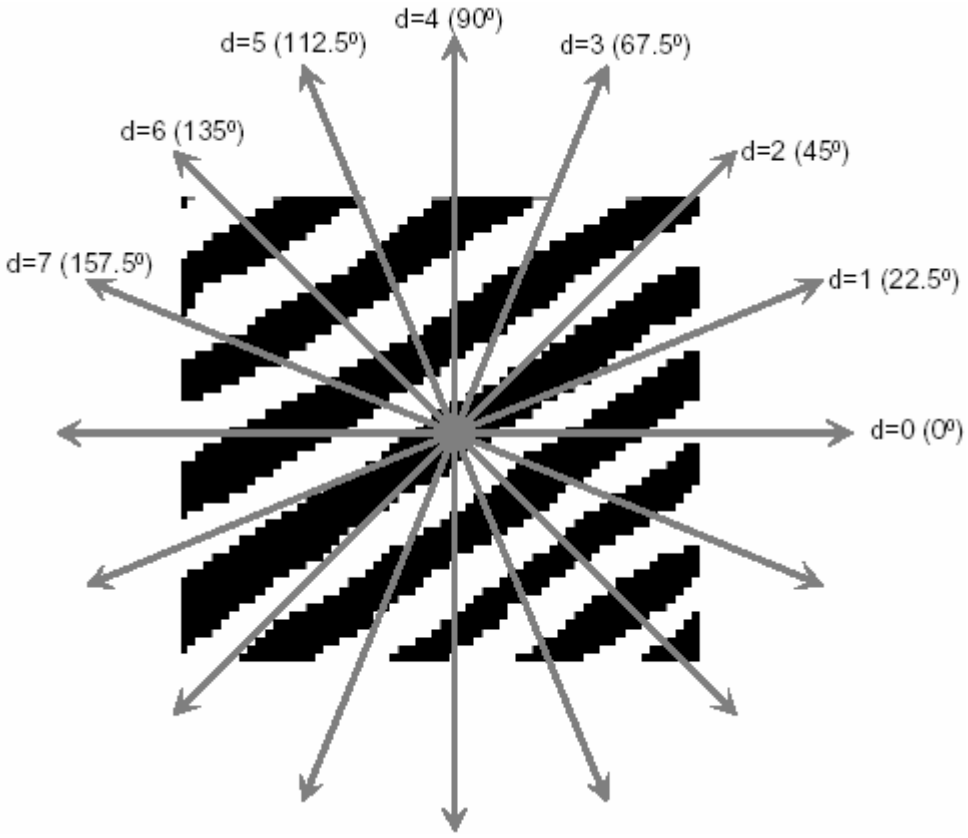


Figure 3.20 Direction computations in 8 directions

Figure 3.21 shows an original fingerprint image after the pre-processing and enhancement stage and its pixel wise directional image in Figure 3.22, while noisy

block before block filtering in various size of region is shown in Figure 3.23. Figure 3.24 shows the block directional images in various region sizes after block filtering stage. Figure 3.25 shows the extracted directional elements of fingerprint image. While Figure 3.26 shows the direction of each 8x8 region which is mapped on to the original fingerprint image before block filtering and after block filtering.

Figure 3.21 An original fingerprint image after pre-processing and enhancement stage

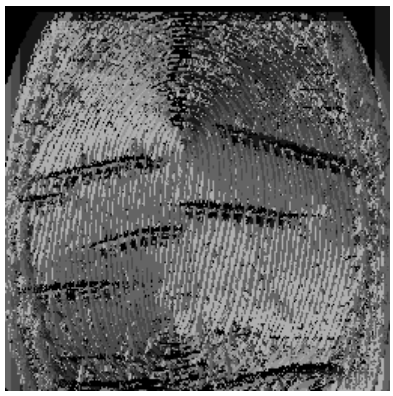
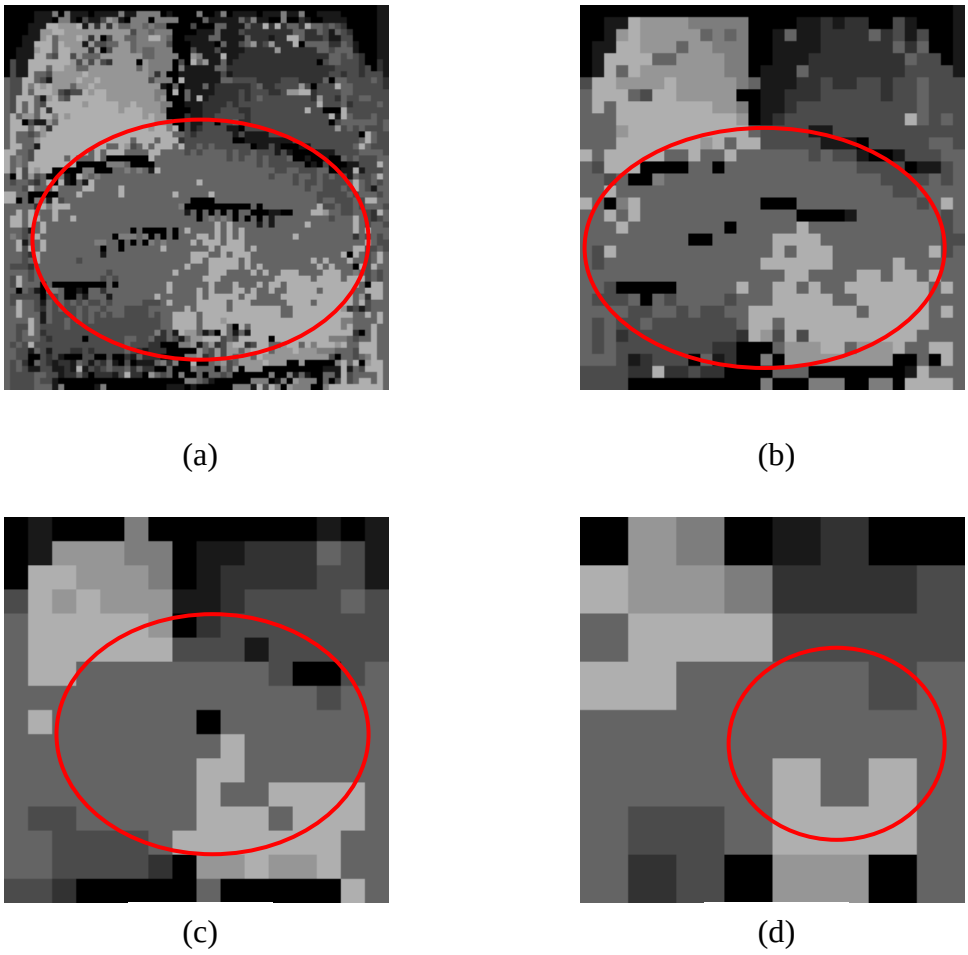


Figure 3.22 Pixel wise directional image of the original fingerprint image

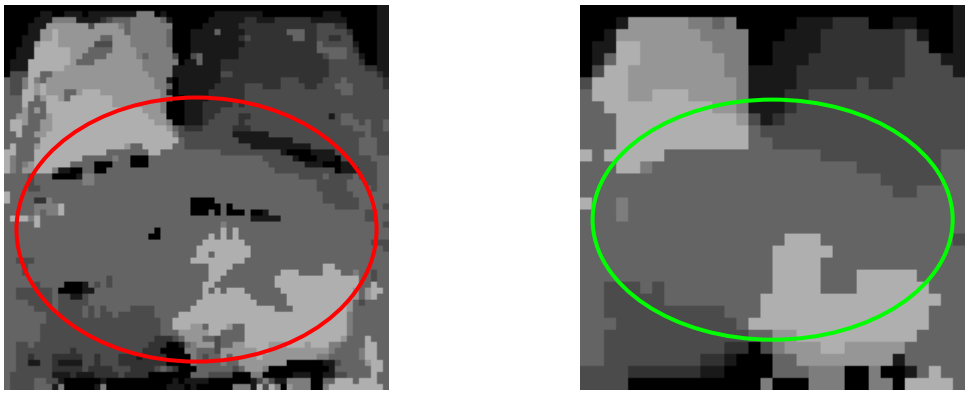


(a) (b)

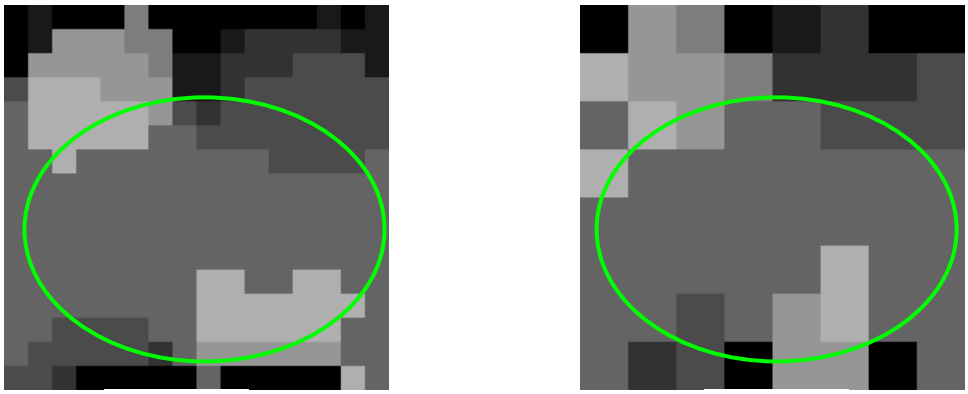
(c) (d)

Figure 3.23 Noisy block directional image: (a) 4x4 region, (b) 8x8 region, (c) 16x16 region, and (d) 32x32 region

In Figure 3.23 above, the circles represent noise area in each directional image. These wrong elements are caused by existing scars in fingerprint image. To remove this noise and fixed the wrong elements, block filtering using median technique as a noise filter is used. Figure 3.24 below shows the block directional image after block filtering.

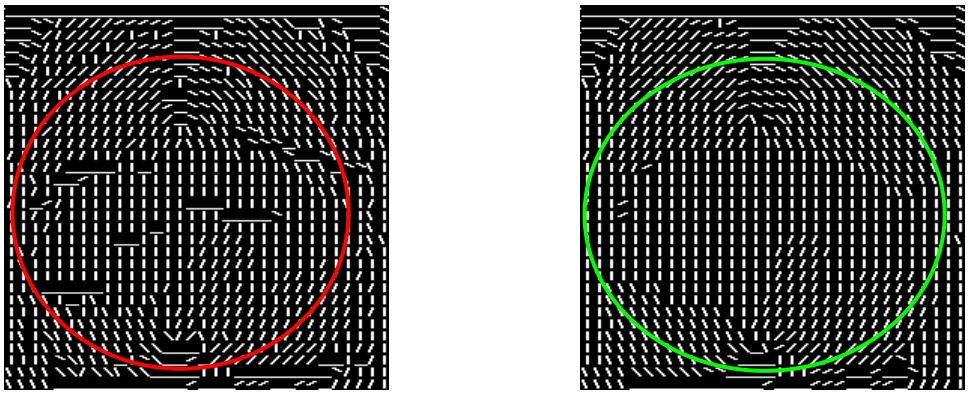


(a) (b)



(c) (d)

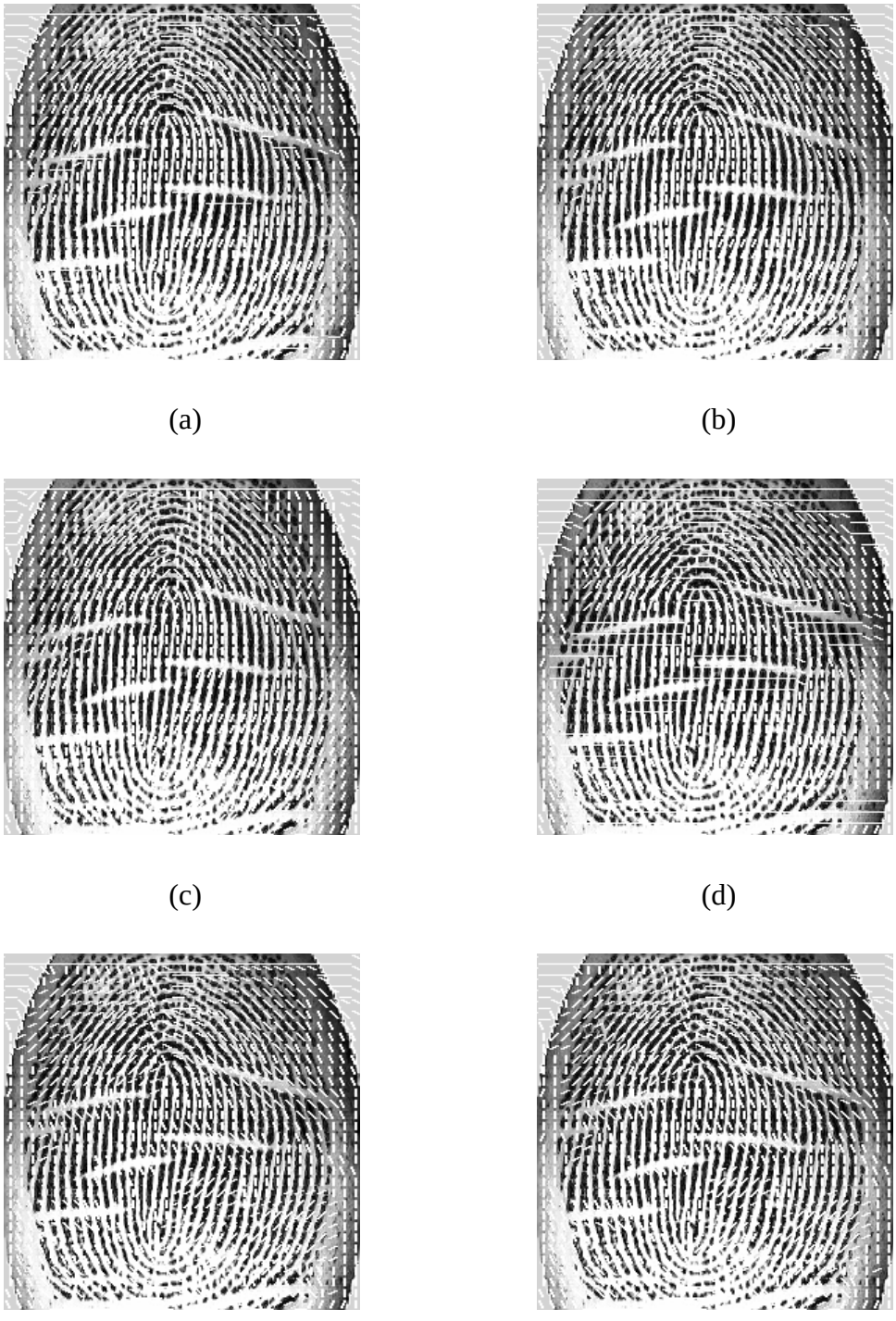
Figure 3.24 Block directional image after block filtering: (a) 4x4 region, (b) 8x8 region, (c) 16x16 region, and (d) 32x32 region.



(a) (b)

Figure 3.25 The directional element of fingerprint image: (a) before block filtering and (b) after block filtering

Figure 3.25 shows the directional image before and after applying the block filtering. Before applying the block filtering, the directional image contains many false directional elements as shown in the red circle. Using median filtering as block filtering, these false minutiae are discarded as shown in the green circle.



(a) (b)

(c) (d)

(e) (f)

(f)

Figure 3.26 The direction mapped on the original images: (a) None block filtering, (b) Median Filtering, (c) Maximum Filtering, (d) Minimum Filtering, (e) Average Filtering, (f) Low-Pass Filtering, (g) High-Pass Filtering.

Figure 3.26 above shows the directional elements mapped on the original fingerprint image using different filter block filtering techniques.

3.3.2 Iterated Least Mean Square Orientation Estimation Algorithm

Hong et al (1998) developed an iterated least mean square orientation estimation algorithm. The main steps of the algorithm are as follows:

1. Dividing the input fingerprint image into blocks size *w x w*. For 500 dpi images, the initial value of *w* is 16.
2. Computing the gradients $\partial x(i, j)$ and $\partial y(i, j)$ at each pixel (i, j). Depending on the computational requirement, the gradient operator may vary from the simple Sobel operator to the more complex Marr-Hildreth operator (D. Marr, 1982).
3. Estimating the local orientation of each block centered at pixel (i, j) using the following equation (A. Rao, 1990).

$$v_x(i,j) = \sum_{u=i-\frac{w}{2}}^{i+\frac{w}{2}} \sum_{v=j-\frac{w}{2}}^{j+\frac{w}{2}} 2\partial_x(u,v)\partial_y(u,v)$$

(**Equation 3.3**)

$$v_y(i,j) = \sum_{u=i-\frac{w}{2}}^{i+\frac{w}{2}} \sum_{v=j-\frac{w}{2}}^{j+\frac{w}{2}} (2\partial_x^2(u,v) - \partial_y^2(u,v))$$

(**Equation 3.4**)

$$\theta(i,j) = \frac{1}{2}\tan^{-1}\left(\frac{v_y(i,j)}{v_x(i,j)}\right)$$

(**Equation 3.5**)

where $\theta(i,j)$ is the least square estimate of the local ridge orientation at the block centered at pixel (i,j). Mathematically, it represents the direction that is orthogonal to the dominant direction of Fourier spectrum of the *w x w* window.

4. Due to the presence of noise, corrupted ridges, valley structures, minutiae, etc in the input image, the estimated local ridge orientation, $\theta(i,j)$ may not always be correct. Since local ridge orientation varies slowly in a local neighbourhood where no singular points appear, a low-pass filter can be used to modify the incorrect local ridge orientation. In order to perform the low-pass filtering, the orientation image needs to be converted into continuous vector field, which is defined as follows:

$$\phi_x(i,j) = \cos(2\theta(i,j))$$

(**Equation 3.6**)

$$\phi_y(i,j) = \sin(2\theta(i,j))$$

(**Equation 3.7**)

where δx and δy are the x and y components of the vector fields, respectively. With the resulting vector fields, the low-pass filtering can then be performed as follows:

$$\Phi'_x(i,j) = \sum_{u=-w_\Phi/2}^{w_\Phi/2} \sum_{v=-w_\Phi/2}^{w_\Phi/2} h(u,v)\Phi_x(i-uw, j-vw)$$

(**Equation 3.8**)

$$\Phi'_y(i,j) = \sum_{u=-w_\Phi/2}^{w_\Phi/2} \sum_{v=-w_\Phi/2}^{w_\Phi/2} h(u,v)\Phi_y(i-uw, j-vw)$$

(**Equation 3.9**)

where h is a 2-dimensional low-pass filter with unit integral and $w_{\phi x} w_\phi$ specifies the size of the filter. Note that smoothing operation is performed at the block level. The default size of the filter is 5x5.

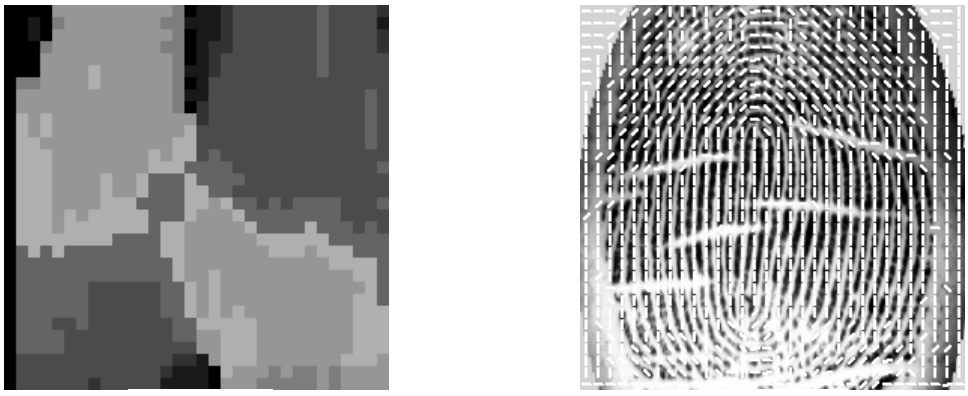
5. Computing the local ridge orientation at (i,j) using equation as follows:

$$o(i,j) = \frac{1}{2}\tan^{-1}\frac{\Phi'_y(i,j)}{\Phi'_x(i,j)}$$

(**Equation 3.10**)

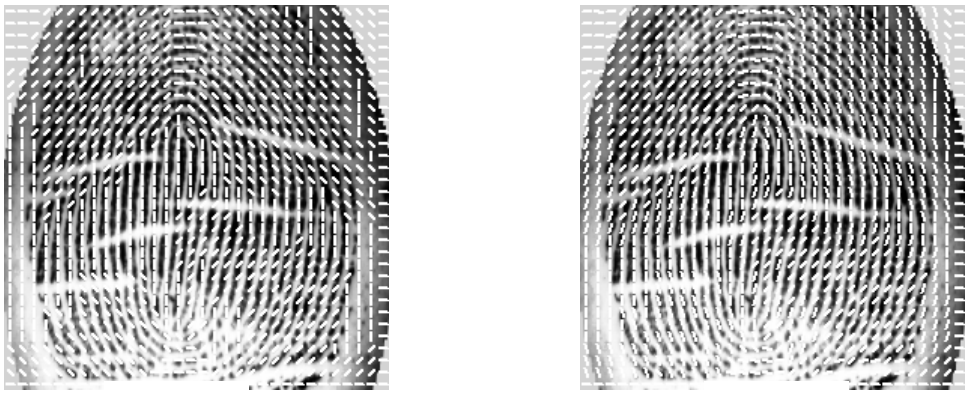
With this algorithm, a fairly smooth orientation field estimate can be obtained.

In this technique, two kind of directions were utilized, i) local ridge orientation using 4 direction was computed, and ii) 8 direction to compute local ridge orientation was used, and the image was divided into 8x8 regions or $w=8$.



(a) (b)

Figure 3.27 (a) block directional image, (b) local ridge orientation in 4 direction without smoothing.



(a) (b)

Figure 3.28 Local ridge orientation in with smoothing, (a) 4 direction, (b) 8 direction

3.4 Fingerprint Reconstruction

Due to the noise and damage in fingerprint images such as scars, sweat holes, etc, a fingerprint image needs to be reconstructed to a new similar image. In the fingerprint enhancement stage, only noise such as spikes and low contrast can be removed and enhanced. The scars or sweat pores in the fingerprint image cannot be removed. To mitigate these problems, the fingerprint needs to be reconstructed. To produce a new fingerprint image, the Directional Fourier Filtering technique was used.

3.4.1 Directional Fourier Filtering

There are three steps involved in creating a new fingerprint image. These steps are Frequency Transformation, Frequency Filtering and Reconstructing Fingerprint Image.

3.4.1.1 Frequency Transformation

Enhancement in process of frequency domain techniques is explained by Gonzales (1993) in his book. First, the Fourier transformation of image is computed, and second, the results of transformation by a filter function is multiplied, and third, the inverse transformation to produce the enhanced image is taken.

In fingerprint images, Fourier spectrum is used to improve fingerprint patterns because fingerprints contain parallel ridges and furrows which represent frequency-like components.

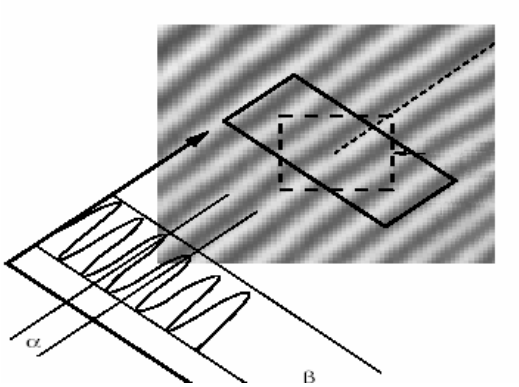
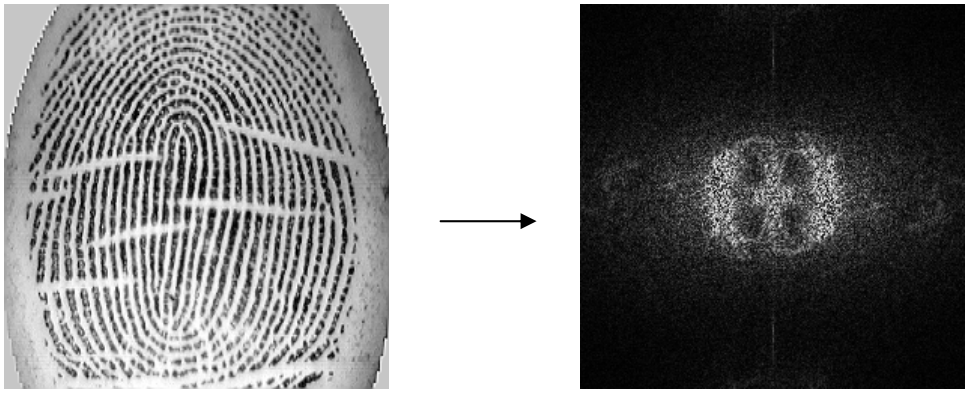


Figure 3.29 Ridges and furrows represents frequency component

In transforming fingerprint image, Fast Fourier Transformation (FFT) is used. Figure 3.30 below shows a fingerprint image in frequency domain.



Original image Image in frequency domain

Figure 3.30 Image transformation in spatial domain into frequency domain using FFT.

3.4.2 Frequency Filter

In implementing filtering in frequency domains the Directional Filter technique is chosen to filter out the noise inside the fingerprint image.

3.4.2.1 Directional Filter

A directional band pass filter passes frequency components unmodified within oriented area while the rest of the component is set to zero. Ikonomopolous and Unser (1984), Ikonomopolous and Kunt (1985) and Kunt, Ikonomopolous and Kocher (1985) introduced a directional filtering approach for texture discrimination. In their works, they isolated the edge information belonging to a limited number of directions using a directional filter so that its frequency response covers a set of frequencies which are within a directional range. They used the following directional filter:

$$G = H * W$$

where H is filter function, W is the Fourier transform of an image and G is the i^{th} of directional filter. H , therefore can be expressed as:

$$\mathrm{H_i} = \begin{cases} 1 & \text{if } \theta_{\mathrm{i}} < \tan^{-1}(v/u) < \theta_{\mathrm{i+1}} \\ & \text{and } u^2 + v^2 > r_c^2 \\ 0 & \text{elsewhere} \end{cases}$$

where u and v denote the spatial frequency co-coordinate, and n the number of directions, with $\theta_i = (i-1)\pi / 2n, \theta_{i+1} = (i+1)\pi / 2n$. Figure 3.31 shows the ideal frequency response defined directional filter by Ikonomopolous.

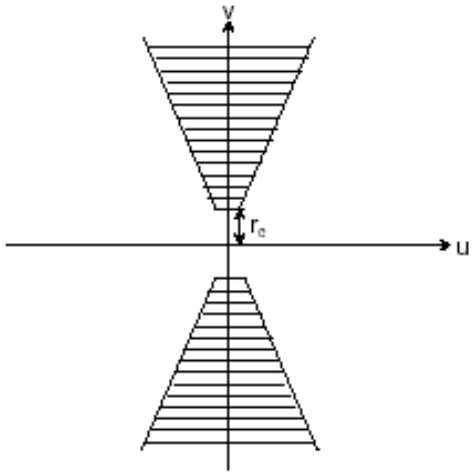


Figure 3.31 A directional filter developed by Ikonomopolous.

3.4.3 Reconstructing Fingerprint Images

At this stage, the frequency image has been filtered using H in each direction G_i . These directions are obtained from the directional image during directional image production stage. Once the frequency image is filtered in each direction G_i , an Inverse Fast Fourier Transform is applied to reconstruct new image with a better quality. In other words, the combination of filter function with the Inverse Fast Fourier Transform applied on directional image and frequency image reconstructs a new fingerprint image. Figure 3.32 shows all the steps involved in Inverse Fast Fourier Transformation.

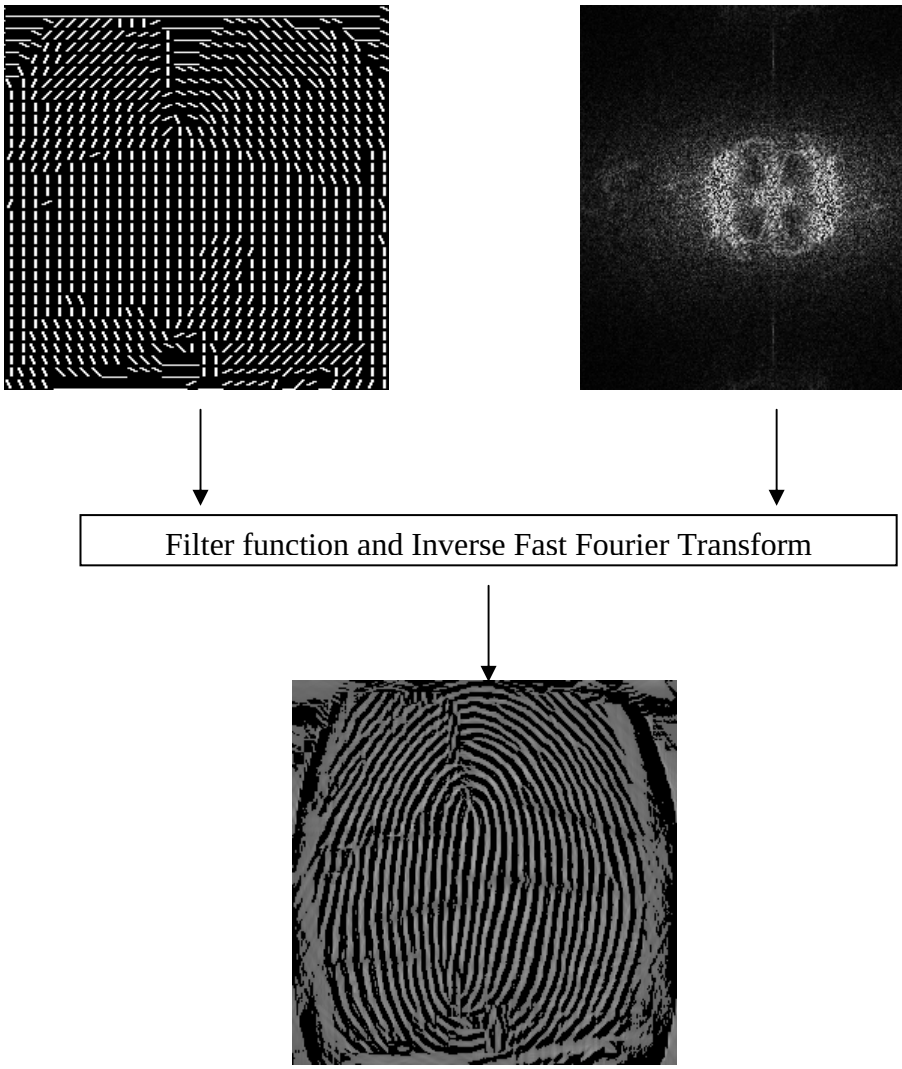


Figure 3.32 Fingerprint image reconstruction process

To reconstruct a new fingerprint image two types of directional images were used. The first directional image is from Mehtre and the second is from Least Square Orientation Estimation Algorithm.

In this study, the Directional Fourier Filtering with Ideal Low-Pass Filter function was used in reconstructing fingerprint images. In the experiment, fingerprint image was filtered in 8 directions.

3.5 Fingerprint Segmentation

Fingerprint segmentation is a process to separate foreground image (ridges) from background image (furrows) and to produce a binary image from greyscale image. Segmentation also removes noise while producing a binary image. Segmentation is defined as a process to identify and group pixels with same attributes in a region or block. There are a few methods to produce segmentation, such as statistical classification, thresholding, edge detection and region detection. The simplest and famous method is thresholding.

Thresholding involves looking at each pixel and deciding whether it should be converted into black or white pixel. This decision is made by comparing each pixel value with a fixed number called a threshold level or threshold value. If the value is less than the threshold value, the pixel is set to black; otherwise it is set to white. The general form of thresholding scheme can be expressed as follows:

$$G(i,j) = \begin{cases} 255 & \text{if } F(i,j) > T \\ 0 & \text{if } F(i,j) \le T \end{cases} \quad (i=0,1...,N,\ j=0,1...,M)$$

(**Equation 3.11**)

where *F(i, j)* indicates the original image, *T* is the threshold level, *G(i, j)* indicates the output binary image and *N* and *M* are the number of rows and columns in the image, respectively. In this fingerprint segmentation, fingerprint image in Figure 3.33 was used as an input image.

Figure 3.33 The new fingerprint image after fingerprint reconstruction

In doing fingerprint segmentation, four techniques were studied. These techniques are Global Thresholding, Regional Average Thresholding, Histogram Based Thresholding and Niblack Binarization.

3.5.1 Global Thresholding

Global thresholding is also called Simple Thresholding. This technique is the easiest compared to other thresholding methods. In this technique, one single threshold value is selected and applied to all pixels in image. This technique gets the average grey scale values from 0 to 255, and usually 128 is chosen as the threshold value. The disadvantage of using this technique is it can damage the image and ultimately will lose some information. Figure 3.34 below shows the results from the Global Thresholding process.

Figure 3.34 Fingerprint image after Global Thresholding

3.5.2 Regional Average Thresholding

Applying a single threshold value for the whole image may cause some feature loss. The average grey level is not the same at different parts of the original image. This is one of the critical problems of fingerprint image. To overcome this problem, the original image divided into small square size regions or blocks such as 32x32, 16x16, 8x8, and 4x4 region-sized windows. In this block, we computed average grey level value using Equation 3.12. Using the average value as threshold value, the block is thresholded.

$$T = \frac{1}{N^2} \sum_{i=0}^{N} \sum_{j=0}^{N} F(i, j)$$

(**Equation 3.12**)

In Equation 3.12, *T* is the average grey level of the windows and it is used as a single threshold value, *N* is the size of each square or rectangular region, and *F(i, j)* is the pixel grey level.

Emiroglu and Akhan (1997a) proposed a Regional Average Thresholding (RAT) scheme for fingerprints. Fingerprint images are first divided into any one of 16x16, 16x8, 8x8 or 8x4 regions. The thresholding is done on each region by using the grey level average of the related region as a single threshold value. This is called regional average thresholding (RAT). In the RAT scheme on a 256 level grey scale image, a window of 4, 8 or 16 pixel squares scans the image starting from the left hand corner of the image at the bottom. An average threshold level is calculated within the current window but only applied to the first half of the current window. The window then moves by half of the used window size pixels to the adjacent square. This time, the left-most of the image is thresholded, although the average threshold level computation is for whole window. The process continues until the entire image is thresholded. Since the average threshold level is calculated regionally, many more of the features are preserved in comparison with the global thresholding. This stage also eliminates the fields that contain no information on the edges of the fingerprint.

As an example, let us follow the progress of RAT for 8x4 windows. The algorithm operates in the following stages as shown in Figure 3.35 below as follows:

1. Division of the images into 8x8 regions.
2. Calculation of the average of grey level in the first 8x8 region.
3. Threshold the leftmost region (8x4) by using average grey level calculated in 2.
4. Movement of the 8x8 operation window by 4 pixels to the right. If right edge of the image is reached, then move the window 8 pixels up and return to the left edge.
5. Repetition of 2 to 4 until the entire image is processed by RAT.

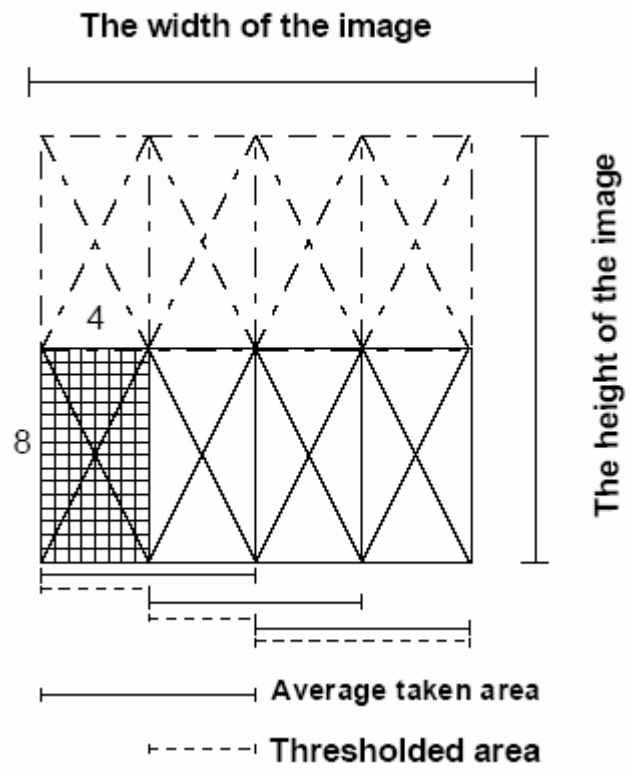


Figure 3.35 The operation of thresholding for 8x4 regions (Emiroglu and Akhan, 1997a)

Figure 3.36 displays the fingerprint images after applying Regional Average Thresholding in 16x16, 16x8, 8x8, 8x4 regions.

(a) (b)

(c) (d)

Figure 3.36 Fingerprint images after Regional Average Thresholding: (a) 16x16, (b) 16x8, (c) 8x8 and (d) 8x4 regions

3.5.3 Histogram Based Thresholding

There are different steps involved in the Histogram Based Thresholding. The first step is to carry out histogram equalization (Equation 3.1), followed by histogram smoothing. Histogram equalization attempts to alter the entire fingerprint image so its histogram is flat and spreads out over the entire range of grey levels. The result is a fingerprint image with better contrast. In examining histogram, too many tall, thin peaks and deep valleys will cause problems. Histogram smoothing, an easy operation removes these spikes, and fills in empty canyons while retaining the same basic shape of histogram. It replaces each point with the average and its two neighbours. In general, an image contains more background image than the foreground image. There are three segmentation techniques: histogram peak technique, histogram valley technique and adaptive technique.

3.5.3.1 Histogram Peak Technique

Histogram Peak technique examines the histogram and selects the threshold value automatically by the histogram peaks. This technique finds the two peaks in the histogram corresponding to background and object or foreground of the fingerprint image. It sets the threshold halfway between the two peaks. Figure 3.37 shows the background peak is at the second two grey levels and the foreground is at the seventh grey level. The midpoint is four, so the low threshold value is four and the highest is nine.

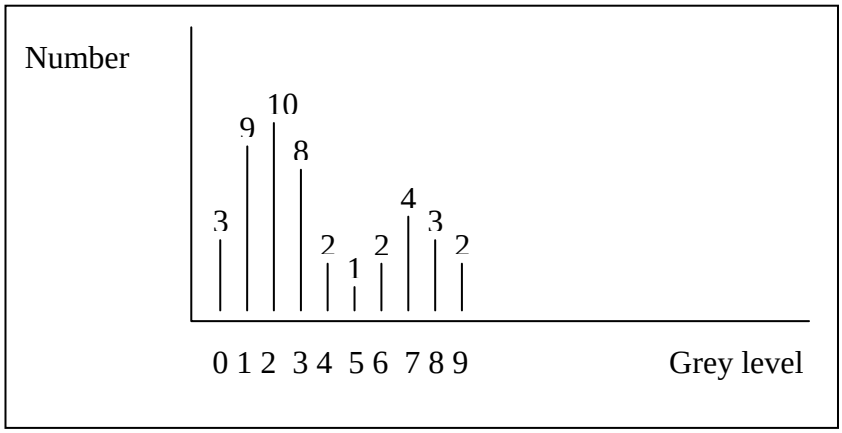


Figure 3.37 Histogram of nine grey level images

The peak technique is a straight forward technique, except for two items. In the histogram shown in Figure 3.37, note that the peak at grey level seven is the fourth highest peak. The peaks at grey level one and three are higher, but they are part of the background mountain of the histogram and do not correspond to the object or foreground. To search the histogram for the peaks, you must use the peak spacing to ensure the highest peaks are separated. If you do not, then you would choose grey level two as the background peak and grey level one as the object peak.

The second item to watch out is to determine which peak corresponds to the background and which correspond to the foreground. Supposedly an image had the

same histogram as shown in Figure 3.38. Which peak do you think corresponds to the background? The peak for grey level eight is the highest but it corresponds to the object, not the background. The reason is that the mountain surrounding the peak at grey level two has a much greater area than the peak next to grey level eight. Therefore, grey level from zero to six occupies the vast majority of the image, and they are the background.

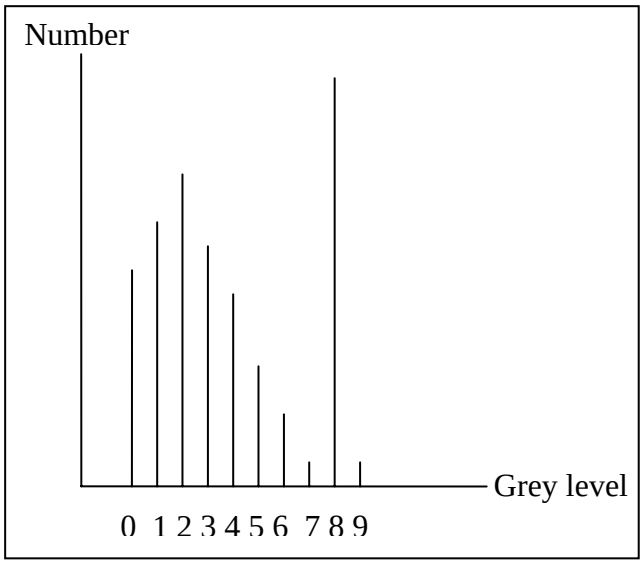


Figure 3.38 : A histogram in which the highest peak does not correspond to the background

Figure 3.39 shows the fingerprint image after the Histogram Peak Technique.

Figure 3.39 Fingerprint image after Histogram Peak Technique

3.5.3.2 Histogram Valley Technique

The second technique, the Histogram Valley technique concentrates on the valleys between them. Instead of setting the midpoint arbitrarily halfway between two peaks, the valley technique searches between the two peaks to find the lowest valley. Looking back at Figure 3.38, the peaks are at grey levels two and eight and the peaks technique would set the midpoint at five. In contrast, the valley technique searches from two through eight to find the lowest valley. In this case, the "valley-point" is at grey level seven.

Figure 3.40 Fingerprint image after Histogram Valley Technique

3.5.3.3 Histogram Adaptive Technique

The final technique, the Histogram Adaptive technique uses the peak of the histogram in a first pass and adapts itself to the objects found in the image in a second pass (Kenneth R. Castleman, 1979). In the first pass, the adaptive technique calculates the histogram for the entire image. It smoothes the histogram and uses the peak technique to find the high and low threshold values.

In the second pass, the technique works on each *ROWSxCOLS* area of the image individually. In each area, the technique segments using high and low values found during the first pass. Then, the mean value for all the pixels segmented is calculated into background and object. These values are then used for that *ROWSxCOLS* area. Then, the area is further segmented using the new values.

Figure 3.41 Fingerprint image after Histogram Adaptive Technique

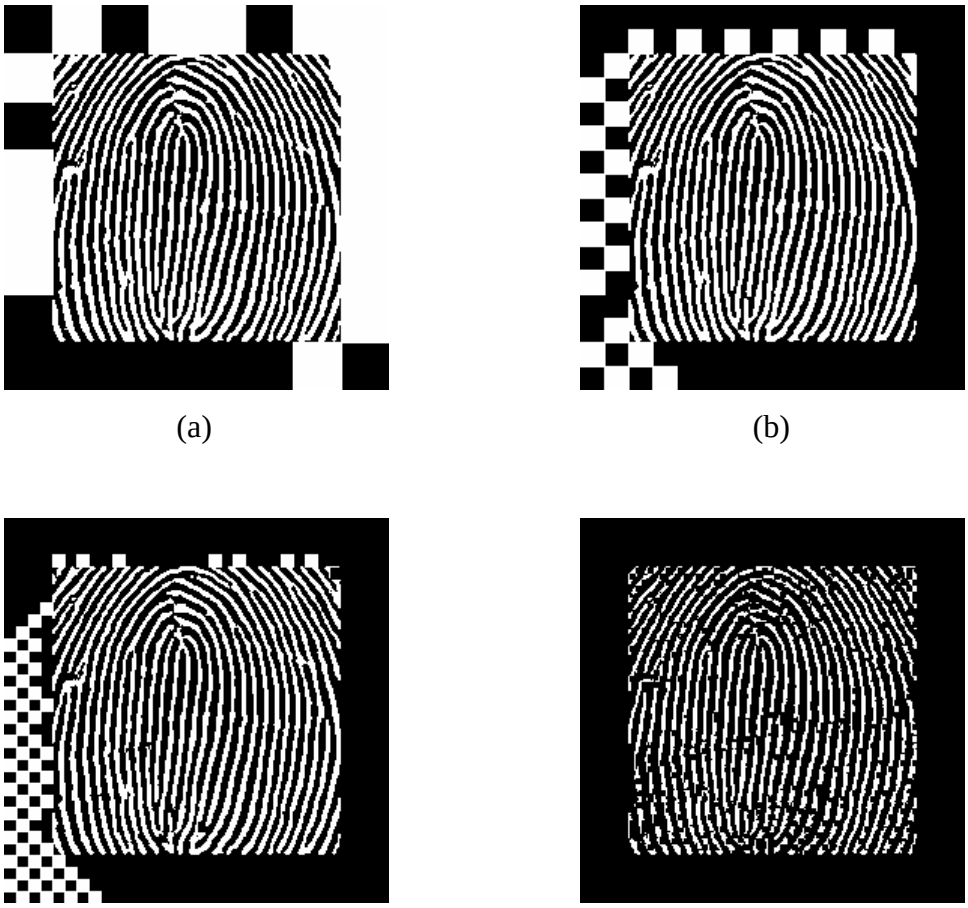
3.5.4 Niblack Binarization

The idea of using the Niblack Binarization method is to vary the threshold over the image based on local mean and local standard deviation. The Niblack Binarization algorithm was selected as it provided robust thresholding and in the presence of shadows and other image defects (D. Trier, 1997). The algorithm calculates a local binarization threshold by calculating the local mean and local standard deviation and then adding to the product of predefined weight constant and the standard deviation using the Equation 3.13 below:

$$T(x, y) = w \times \sigma(x, y) + \mu(x, y)$$

(**Equation 3.13**)

In the equation, T is the threshold value at pixel (x, y) and w is the weight $\sigma(x, y)$ and $\mu(x, y)$ are the standard deviation and mean of local neighbourhood of a pixel (x, y) respectively. This will make the thresholding process to intelligently adapt to its pixel neighbourhood. In this research, 4x4, 8x8, 16x16 and 32x32 pixels and w value of -0.5 are used.



(a) (b)

(c) (d)

Figure 3.42 Fingerprint images after Niblack Binarization, (a) 32x32, (b) 16x16, (c) 8x8, (d) 4x4 regions.

3.6 Fingerprint Thinning

An effective edge-thinning algorithm is very important in image segmentation and objects identification since it increases the possibility of success in detecting objects in the image and saves the processing time in the next steps such as labelling and image transformation. The aim of the thinning algorithm is to make the fingerprint image much simpler for further processing such as feature extraction. The thinning algorithm produces an image where all ridges are one pixel wide known as skeletons. This process acquires a binary image as input. One major advantage of thinning process is the reduction of memory space required for storing the essential structural information presented in a pattern. Moreover, it simplifies the data structure required in a pattern analysis.

Connectivity is an important property that must be preserved in the thinned object. Therefore, border pixels are deleted in such a way that object connectivity is maintained. Thinning algorithms satisfy the following constrains:

1. They maintain connectivity at iteration. They do not remove border pixels that may cause discontinuities.
2. They do not shorten the end of thinned shape limbs.

In fingerprint thinning, the binary fingerprint image is used as in Figure 3.43as input.

Figure 3.43 Binary fingerprint image as input image

There are three methods of fingerprint thinning namely are Two-Way Pass, Fast Thinning Algorithm and Ridge Line Following Thinning.

3.6.1 Two-way Pass

The two-way pass technique requires two successive iterative passes which is described in Zha (1984) and Gon (1987). In step 1, a logical rule P_1 is applied locally in 3 x 3 neighbourhoods to flag border pixels that can be deleted. These pixels are only flagged (not deleted) until the entire image is scanned. Deletion of all flagged pixels is performed afterwards. In step 2, another logical rule P_2 is applied locally in 3 x 3 windows to flag border pixels for deletion. When the entire image has been scanned, the flagged pixels are deleted. This procedure is applied iteratively, until no more thinning can be performed. In the following Equation 3.14, let;

$$N(p_0) = \sum_{i=1}^{8} p_i$$

(**Equation 3.14**)

denotes the number of object pixels $(p_i = 0,1, i = 1,....,8)$ in the 3 x 3 window (excluding the central pixel) and $T(p_0)$ denotes the number of $0 \rightarrow 1$ transition in the pixel sequence $p_1 p_2 p_3 ... p_8 p_1$. Therefore, the logical rules p_1, p_2 used in the two steps of the algorithm can be written in the following equations 3.15 and 3.16:

$$P_1 = (2 \le N(p_0) \le 6) \&\& (T(p_0) = 1) \&\& (p_1 . p_3 . p_5 = 0) \&\& (p_3 . p_5 . p_7 = 0)$$

(**Equation 3.15**)

$$P_2 = (2 \le N(p_0) \le 6) \,\&\&\, (T(p_0) = 1) \,\&\&\, (p_1.p_3.p_7 = 0) \,\&\&\, (p_1.p_5.p_7 = 0)$$

(**Equation 3.16**)

The products of the form $p_i.p_j.p_k$ denote logical conjunctions of the corresponding pixels. The first condition of both logical predicates states that the central pixels can be deleted if it possesses at least one and at most six 8-connected neighbours in 3 x 3 windows. If the central pixel has only one neighbour, it cannot be deleted because the skeleton limb will be shortened. If the central pixel has more than six neighbours, central pixel deletion is not permitted because it will cause object erosion. The second predicate $T(p_0) = 1$ ensures if the pixels of the perimeter of the 3 x 3 neighbourhoods form only one connected component. The third and fourth predicates $(p_1.p_3.p_5 = 0)$ and $(p_3.p_5.p_7 = 0)$ in Equation 3.16 are satisfied if $p_3 = 0$ or $p_5 = 0$, or if $(p_1 = 0$ and $p_7 = 0)$. Examples of these three cases are shown in Figure 3.44. The central pixel belongs to the East boundary $(p_3 = 0)$, to the South boundary $(p_5 = 0)$, or to the North-West object corner $(p_1 = 0, p_7 = 0)$. In the first pass, the algorithm removes pixels belonging to one of these three cases. In the second pass, the thinning algorithm removes pixels having $p_1 = 0$ or $p_7 = 0$ or $(p_3 = 0$ and $p_5 = 0)$, as can be seen by inspecting Equations 3.16. In the second pass, pixels lying at the North boundaries, or the West boundaries, or the South–East corner points are removed, as can be seen in Figure 3.45

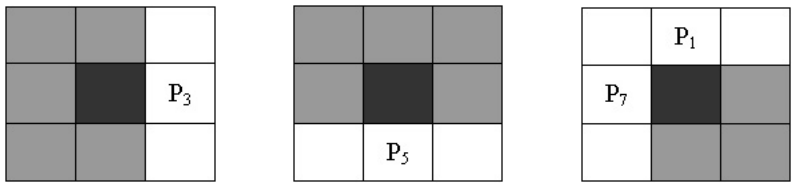
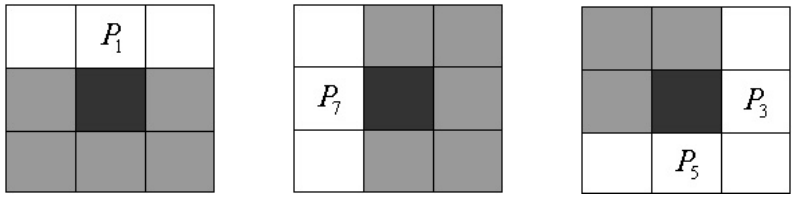


Figure 3.44 Central window pixels belonging to: (a) East boundary; (b) South boundary; (c) North-West corner point.



(a) (b) (c)

Figure 3.45 Central window pixels belonging to: (a) North boundary; (b) West boundary; (c) South-East corner.

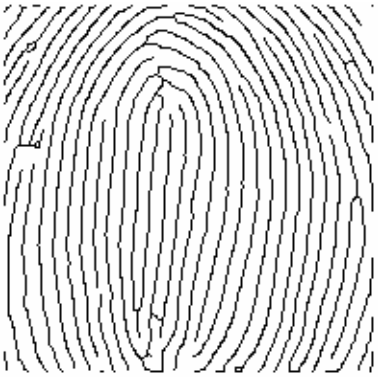


Figure 3.46 Fingerprint image after Two-Way Thinning Algorithm

3.6.2 Fast Thinning Algorithm

There are two main steps in the Fast Thinning Algorithm (FTA) that are repeated until the obtained image approaches the medium axis of the original image. In the first step, the contour of the image is marked, while in the second step, the marked contour is analyzed to verify which pixels-n belonging to this contour that should be deleted. The contour of an image is formed by a pixels-on that is found in the innermost and most distant position of this image. There are four main characteristics of the FTA:

1. Maintains the connectivity and preserves the end points.
2. Resulting skeleton approaches the medium axis of the original image.
3. Practically immune to noise.

4. Execution time is very fast.

The process of delimitation and deletion of the contour of the image carried out by the FTA algorithm is divided into five stages as follows:

1. Delimitation of the contour of the image.
2. Deletion of the contour of the image.
3. Verification of noise.
4. Verification of connectivity.
5. Verification of deviation.

3.6.2.1 Delimitation of the contour of the image

The objective of the first stage is to carry out the delimitation of the contour of the image. The FTA algorithm scans the original-matrix from left to right and from top to bottom. Figure 3.47 illustrates the chain code of 8 directions. For each pixel-on that is scanned, the following steps are carried out:

1. If the pixel-on that is being scanned possesses a pixel-on neighbour that is found in the 2 direction, and does not posses a pixel-on neighbour that is found in the 6 direction or vice versa, the pixel-on that is being scanned is marked with the value 2 to be considered as belonging to the contour of the image.

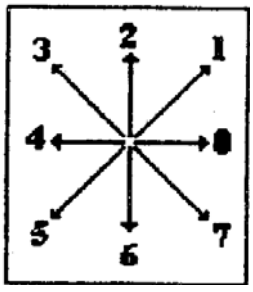


Figure 3.47 Chain codes of 8 directions.

2. If the pixel-on that is being scanned possesses a pixel-on neighbour that is found in the 4 direction and does not possesses a pixel-on neighbour that is found in the 0 direction or vice versa, the pixel-on that is being scanned is marked with the value 2 to be considered as belonging to the contour of the image.
3. If the pixel-on that is being scanned does not possess any pixel-on neighbour that is found in the 0, 2, 4, or 6 direction, the algorithm verifies whether the pixel-on neighbour in any of the 8 directions of the chain code. In the affirmative case, the pixel-on that is being scanned is deleted from the image, to be considered by the algorithm as noise. The algorithm will consider the set of pixels-on formed by up to 4 pixels-on as noise.
4. The process of delimitation of the contour of the image is carried out until the lower extreme right of the image is found.

After terminating the process of delimitation of the contour of the image, the algorithm executes the second stage '*Deletion of the contour of the image*', whose objective is to delete the contour of the image that is delimited.

3. 6.2.2 Deletion of the contour of the image.

In this stage, the image scans the original-matrix from left to right and from top to bottom, analyzing each pixel-on marked, with the value 2 whether or not this pixel-on should be deleted from the image. In this analysis, it is verified whether the pixel-on marked with value 2 and its pixels-on neighbour are disposed as illustrated on Figure 3.48 and Figure 3.49.

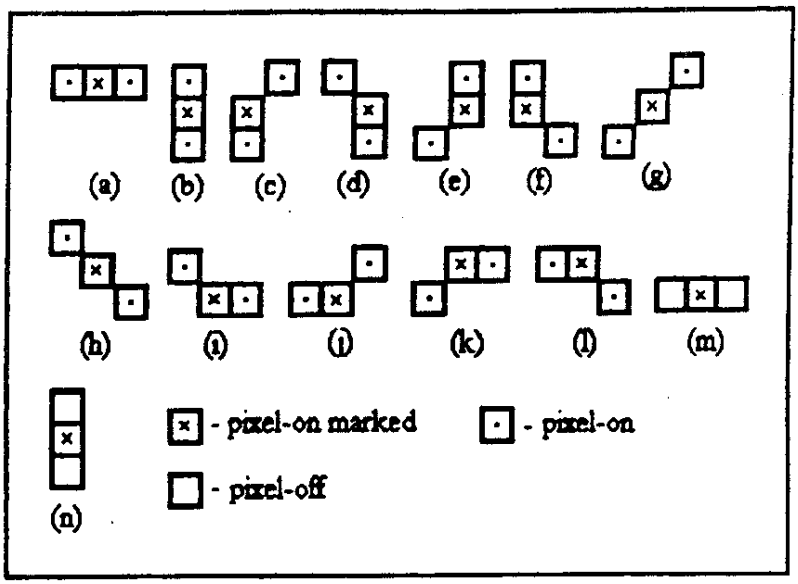


Figure 3.48 Templates utilized in the verification of the disposition of the pixel-on marked with the value 2 and their pixel-on neighbours.

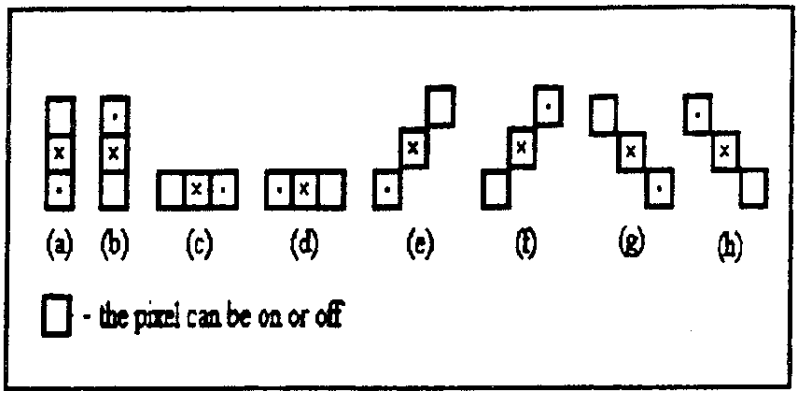


Figure 3.49 Templates utilized in the verification of the disposition of the pixel-on marked with the value 2 and their pixels-on neighbours.

If the disposition of the pixel-on marked with the value 2 and its pixel-on neighbours coincide with any of the templates (a) to (l) in Figure 3.47, the *sign0* variable is equal to l. If the *sign0* variable equal to 1 and the disposition of a pixel-on marked with the value 2 and its pixels-on neighbours coincide with the template (m) or (n) of Figure 3.47, the *sign2* variable is equal to 1.

If the disposition of the pixel-on marked with value 2 and its pixels-off neighbours do not coincide with the templates (a) to (l) of Figure 3.47, the algorithm verifies whether the disposition of the pixel-on marked with the value 2 and its pixels-on neighbours coincide with: i) the template (a) or (b) of Figure 3.48, and the template (m) of Figure 3.47 or ii) the template (c) or (d) of Figure 3.48 and the template (n) of Figure 3.47 or iii) the template (e) or (f) of Figure 3.48 and the templates (m) and (n) of Figure 3.47 or iv) the template (g) or (h) of Figure 3.48) and the templates (m) and (n) of Figure 3.47.

In the affirmative case, if the number of pixel-on neighbours to the pixel-on marked with the value 2 is less than 2, the algorithm proceeds to the third stage, '*verification of noise*'. Otherwise, if the number of pixels-on neighbours to the pixel-on marked with the value 2 is greater or equal to 2 and different from 3, the *sign1* variable is equalled to 1 and if the number of pixels-on neighbours of the pixel-on marked with the value 2 is equal to 2, the coordinate of the pixel-on marked with the value 2 is stored for later verification as to whether this pixel-on should stay in the image or be deleted as noise by the algorithm.

If the values of the *sign0* and *sign1*, or *sign1* variable are equal to 1, the algorithm verifies whether the pixel-on neighbours of the pixel-on marked, with the value 2 are found in the 1, 3, 5, or 7 direction and the pixel-on marked with the value 2 possesses only 2 pixels-on neighbours. In the affirmative case, the algorithm executes the fifth stage, which is the verification of deviation..

If the values of the *sign0* and *sign2* and *sign1* variables are different from 1, the algorithm verifies whether:

i) the pixel-on marked with the value 2 possesses a pixel-on neighbour in 1 or 3 direction and does not posses a pixel-on neighbour in the 2 direction, or

ii) ii) the pixel-on marked with the value 2 does not posses pixels-on neighbours found in the 1 and 3 direction and possesses a pixel-on neighbour found in the direction 2, or

iii) the pixel-on marked with the value 2 possesses a pixel-on neighbour found in the 5 or 7 direction and does not posses a pixel-on neighbour found in the 6 direction, or

iv) the pixel-on marked with the value 2 does not posses pixels-on neighbour found in the 5 and 7 directions and possesses a pixel-on neighbour found in the 6 direction and the pixel-on marked with the value 2 possesses a fewer than 2 pixels-on neighbour found in the 0 and 4 direction and the number of pixels-on neighbours to the pixel-on marked with the value 2 is different from 1.

In the affirmative case, the algorithm proceeds to the fourth stage, the verification of connectivity. Otherwise, the pixel-on marked with the value 2 is deleted from the image, considered by the algorithm to be noise.

After terminating the scanning of the original-matrix from left to right and from top to bottom and analyzing whether the pixels-on marked with the value 2 belong to the contour of the image or not, the algorithm carries out the analysis of all the pixels-on whose coordinates were stored in STEP 1. In this analysis, if the pixel-on whose coordinate was stored in STEP 1 possesses 2 pixel-on neighbours, the algorithm executes the third stage. Otherwise, if the pixel-on whose coordinate was stored in STEP 1 possesses more than 2 pixels-on neighbours, the pixel-on whose coordinate was stored in STEP 1 is deleted from the image, considered by the algorithm as noise.

3.6.2.3 Verification of Noise

The objective the verification of noise stage is to analyse whether the pixel-on marked with value 2 represents a noise in the image or not. The algorithm verifies whether the pixel-on marked with value 2 possesses 2 or more pixels-on neighbours. In the affirmative case, the algorithm verifies whether among the pixels-on neighbour of the pixel-on marked with the value 2, there are 2 adjacent pixels-on. If this occurs, the *sign1* variable is equal to 0 and the pixel-on marked with value 2 is removed from the image, considered as noise. The pixel-on marked with value 2, used in this stage is the one analyzed in the second stage that satisfied the condition necessary for this stage. Figure 3.50 below illustrates a case in which the pixel-on marked with the value 2 possesses 2 adjacent pixels-on neighbour found in the 2 and 3 directions, according to the chain code of 8 directions.

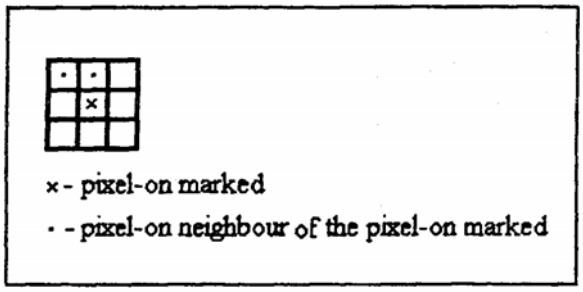


Figure 3.50 Pixel-on marked with value 2 and its 2 adjacent pixels-on neighbours.

If the pixel-on marked with the value 2 possesses a pixel-on neighbour, the algorithm denominates this pixel-on marked *pn* and verifies whether i) the pixel-on *pn* found in the 0 or 4 direction in relation to the pixel-on marked with the value 2 and possesses a pixels-on neighbours found in 2 and 6 directions, or ii) the pixel-on *pn* found in a direction different from 0 and 4 directions, in relation to the pixel-on marked with value 2 and the pixel-on *pn* possesses 2 or more pixels-on neighbours. In the affirmative case, the pixel-on marked with the value 2 is deleted from the image, considered as noise. Otherwise, the *sign1* variable is equal to 1.

3.6.2.4 Verification of Connectivity

In this stage, the objective is to verify whether pixel-on being deleted from the image will not provoke the disconnection of other parts that will constitute the final skeleton if this image.

The algorithm verifies the direction the pixel-on marked with the value 2 possesses a pixel-on neighbours. The algorithm denominates as *pn* each one of the pixel-on marked with the value 2. In this stage, the pixel-on marked with the value 2 is the pixel-on that was analyzed in the second stage and satisfies the conditions to proceed in this stage. For each pixel-on *pn,* the following actions are described:

1. The algorithm verifies whether:
   i) the pixel-on *pn* is found in the 5 direction in relation to the pixel-on marked with the value 2 and does not posses pixels-on neighbours that are found in the 0 and 2 directions, or
   ii) the pixel-on *pn* is found in the 7 directions in relation to the pixel-on marked with the value 2 and does not posses pixels-on neighbours that are found in 2 and 4 directions, or
   iii) the pixel-on *pn* is found in the 6 direction in relation to the pixel-on marked with the value 2 and does not posses pixels-on neighbours that are found in 1, 2, and 3 directions, or
   iv) the pixel-on *pn* possesses only one pixel-on neighbour that is found in the 2 direction. In the affirmative case, the pixel-on marked with the value 2 stays in the image. Due to this, this pixel-on is unmarked.
2. The algorithm verifies whether:
   i) the *pn* pixel-on is found in the 1 direction in relation to the pixel-on marked with the value 2 and does not posses pixels-on neighbour that are found in the 4 and 6 direction, or
   ii) the *pn* pixel-on is found in 3 direction in relation to the pixel-on marked with the value 2 and does not possess pixels-on neighbours that are found in the 0 and 6 directions, or

iii) the *pn* pixel-on is found in the 2 direction in relation to the pixel-on marked with the value 2 and does not possess pixels-on neighbours that are found in the 5, 6 and 7 directions. In the affirmative case, the pixels-on marked with the value 2 should stay in the image. Due to this, this pixel-on is unmarked.

3. The algorithm verifies whether
   i) the pixel-on marked with the value 2 possesses a pixel-on neighbour that is found in the 3, 4 or 5 direction and the *pn* pixel-on is found in the 0 directions in relation to the pixel-on marked with the value 2 and does not possess pixels-on neighbours that are found in 3, 4 and 5 directions, or
   ii) the pixel-on marked with the value 2 possesses a pixel-on neighbour that is found in the 0, 1 or 7 direction and the *pn* pixel-on is found in the 4 direction in the relation to the pixel-on marked with the value 2 and does not possesses pixels-on neighbours that are found in the 0, 1, and 7 directions and the pixel-on marked with the value 2 does not possess pixels-on neighbours that are found in the 2 and 6 directions. In the affirmative case, the pixel-on marked with the value 2 must stay in the image. Due to this, this pixel-on is unmarked.
4. If none of the previous conditions is satisfied, the pixel-on marked with the value 2 is deleted from the image

After verifying that all pixel-on *pn* neighbours to the pixel-on marked with the value 2 are analyzed, the algorithm terminates the process of the fourth stage. Otherwise, the algorithm executes the proceedings described above on the pixels-on *pn* that were not analyzed.

3.3.2.5 Verification of Deviation

The objectives of the verification of deviation stage are verify and correct the sequence in which the pixel-on appears along the skeleton of that image, making sure that the variation of the direction between adjacent pixels-on is kept to the minimum

level. If the pixel-on marked with the value 2 possesses pixel-on neighbours that are found in the 1 and 7 directions as illustrated in Figure 3.51(a), the value of the abscissa of the pixel-on marked with the value 2 is increased to one unit. The algorithm denominates as *nva1* the value of the abscissa of pixel-on marked with the value 2 increased to one unit. If the number of pixel-on neighbours to be positioned *nva1* is less than 4, the pixel-on marked with the value 2 is moved to the position *nva1*which is illustrated in Figure 3.51 (b). If the pixel-on marked with the value 2 possesses pixels-on neighbours that are found in the 1 and 3 directions as illustrated in Figure 3.51 (c), the value of the ordinate of the pixel-on marked with the value 2 is decreased one unit. The algorithm denominates as *nvo1*, the value of the ordinate of the pixel-on marked with the value 2 decreased one unit. If the number of pixel-on neighbours to the position *nvo1* is less than 4, the pixel-on marked with the value 2 is moved to the position *nvo1* as illustrated in Figure 3.51 (d). If the pixel-on marked with the value 2 does not posses a pixel-on neighbour that is found in 1 or 3 direction, the algorithm verifies the pixel-on marked with the value 2 possesses pixel-on neighbours that are found in 3 and 5 directions as is illustrated in Figure 3.51 (e), and the value of abscissa of the pixel-on marked with the value 2 is decreased one unit. The algorithm denominates as *nva2*, which is the value of abscissa of the pixel-on marked with the value 2 decreased one unit. If the number of pixel-on neighbours to the position *nva2* is less than 4, the pixel-on marked with the value 2 is moved to the position *nva2* as illustrated in Figure 3.51 (f). If the pixel-on marked with the value 2 does not possess a pixel-on neighbour that is found in the 3 or 5 direction, the algorithm verifies the pixel-on marked with the value 2 possesses pixel-on neighbours that are found in the 5 and 7 directions as illustrated in Figure 3.51 (g), and the value of the ordinate of the pixel-on marked with the value 2 is increased one unit. The algorithm denominates as *nvo2* the value of the ordinate of the pixel-on marked with the value 2 increased one unit. If the number of pixels-on neighbours to the position *nvo2* is less than 4, the pixel-on marked with the value 2 is moved to the position *nvo2* as is illustrated in Figure 3.51 (h). Figure 3.51 illustrates the analysis carried out in this stage.

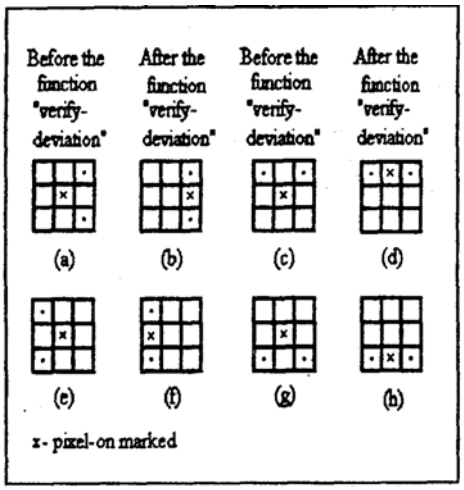


Figure 3.51 Analysis carried out during the deviation verification stage

Figure 3.52 illustrates a case in which the pixel-on marked with the value 2 is moved to the position found.

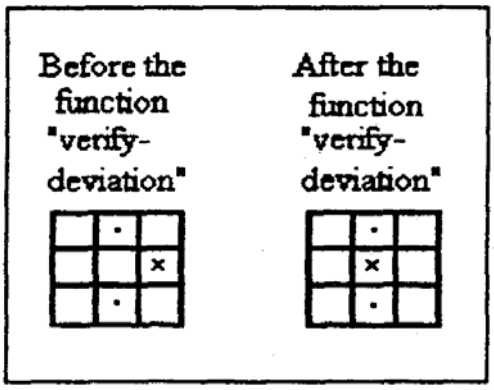


Figure 3.52 Case example in which the pixel-on marked with the value 2 is moved to the position found.

Figure 3.53 Thinned fingerprint image using Fast Thinning Algorithm

3.6.3 Ridge Line Following Algorithm

Emiroglu (1998) proposed the thinning algorithm based on ridge line following and is used only on threshold fingerprint images. The algorithm uses black pixels as the ridges for fingerprint. The thinning algorithm presented here has been designed particularly for fingerprint images. Before the thinning operation starts, it is necessary to obtain a block directional image which produces the direction of each pixel value in a fingerprint image. The aim of the thinning algorithm is to remove redundant black pixels in the image and to produce a thinned image. Black pixels to be removed from the threshold fingerprint image depend on the direction of the Ridge Meeting Point (RMP) and the Ridge Continuity Point (RCP) which are explained below.

Ridge Meeting Point (RMP) is a point in the image where the algorithm meets a ridge. After obtaining a block directional image, the thinning algorithm then starts to scan the image from the bottom left line to the right side line by line, and the algorithm tries to find a black pixel at the RMP location. A part of zoomed fingerprint pattern is shown in Figure 3.54. The pattern contains 3 ridges which have been marked in the diagram.

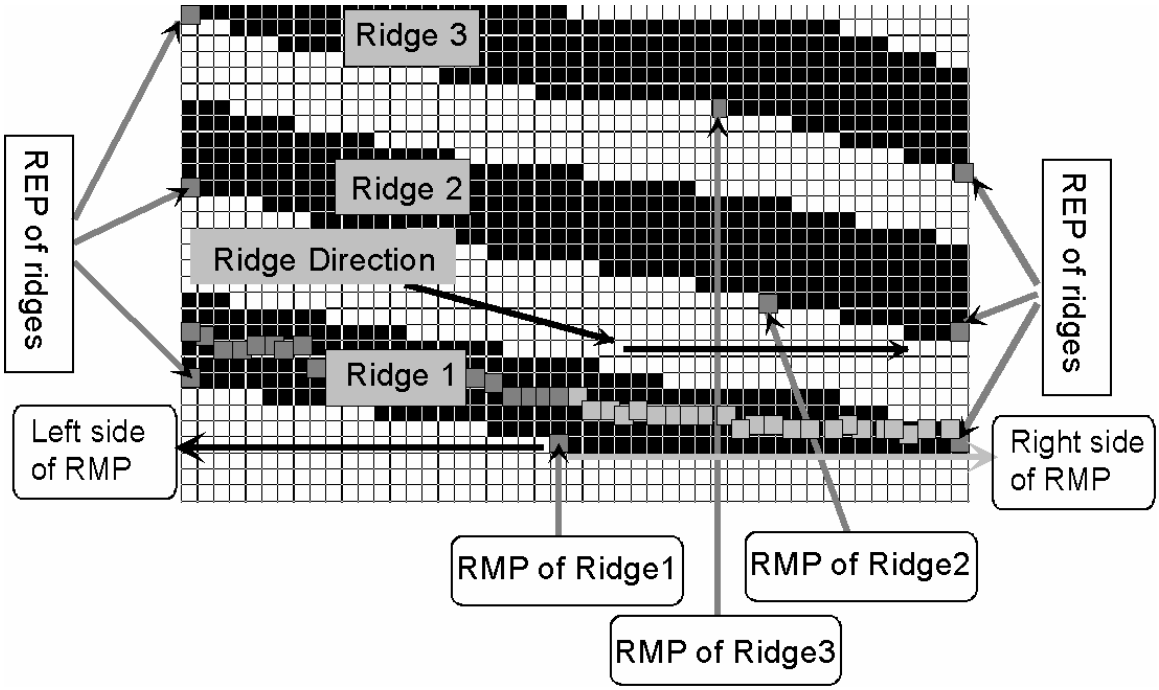


Figure 3.54 The RMP and REP ridges and the thinned points (Emiroglu, 1998)

Once the RMP has been found by the algorithm, there are two possible paths that the algorithm may follow, which are the left side and the right side of the RMP as seen in Figure 3.54. The algorithm always gives priority to the left side of the ridge.

After selecting the path to be followed, the algorithm then uses the ridge direction of the RMP to remove black pixels found on the same line with RMP either horizontally or vertically. If the direction of the RMP is one of $0^o, 22.5^o$ or157.5^o, the algorithm counts the black pixels until the first white pixel in the vertical line, and processed black pixels are removed from the thresholded image. The pixel value in the middle of vertical line is selected as a thinned point. If the direction of RMP is one of $45^o, 67.5^o, 90^o, 112.5^o, 135^o$ angle values, the algorithm counts the black pixels until the first white pixel in the horizontal line and processed black pixels are removed from the thresholded image. The pixel value in the middle of the horizontal line is selected as a thinned point. Then a window size of 3x3 pixels is centered at the RMP. Within this window the RCP is searched. The black pixels within the window

are candidates for RCP. The RCP candidates for both the left side and the right side following are shown in Figure 3.55 and Figure 3.56 respectively.

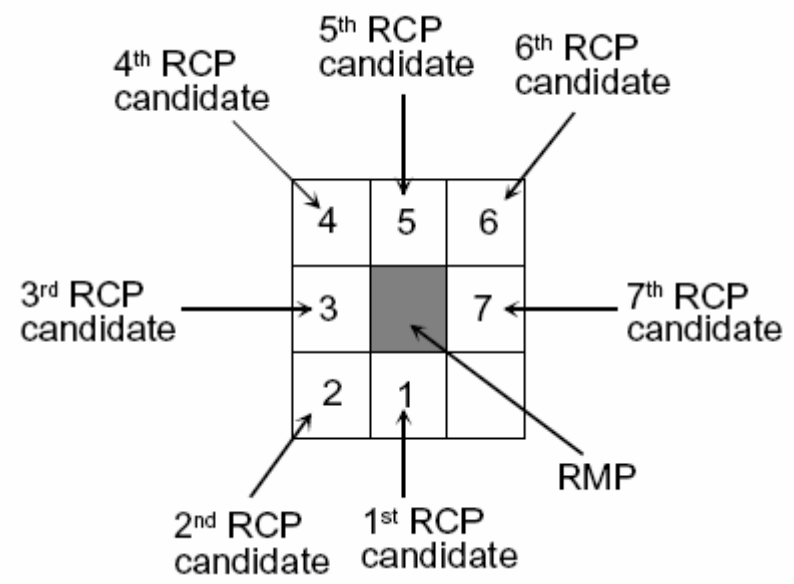


Figure 3.55 RCP candidates for left side following (Emiroglu, 1998)

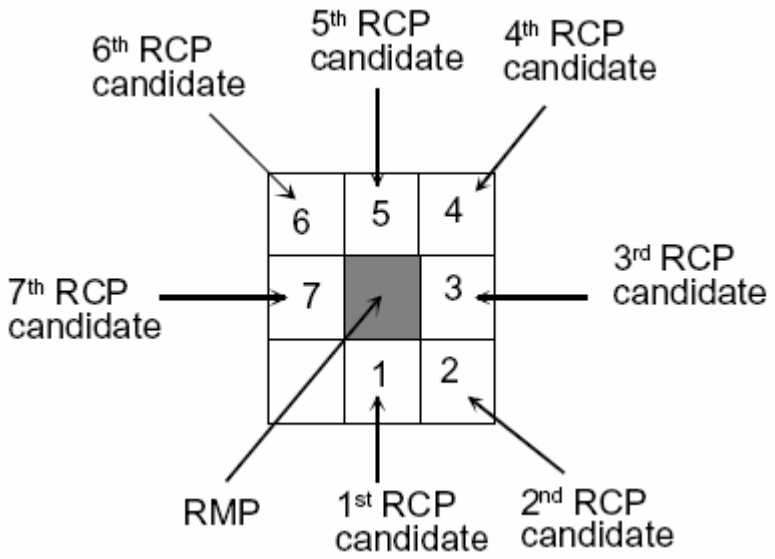


Figure 3.56 RCP candidates for right side following (Emiroglu, 1998)

The algorithm scans pixels within the window in the same order given in Figure 3.55 and finds the first pixel that satisfies the following conditions:

1. The candidate must be a black pixel,
2. The candidate must have a white adjacent pixel.

The first black pixel that satisfies the conditions given above is set as the new RCP. In the following example, Figure 3.57, pixels number *3* and number *4* both satisfy the two conditions given above. Since number 3 takes priority, it is assigned as RCP point.

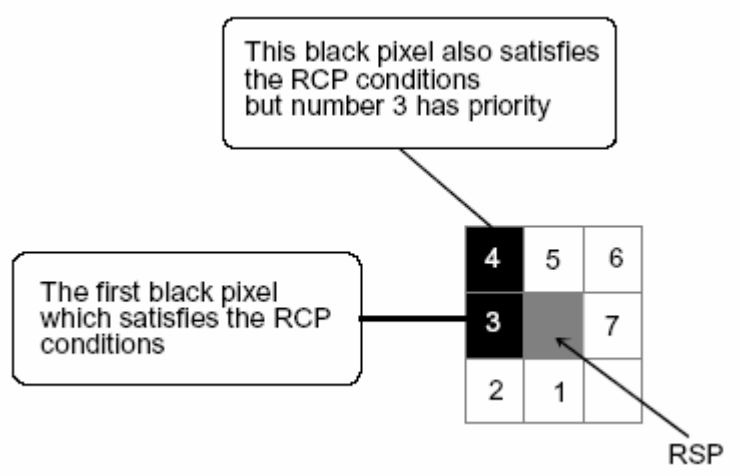


Figure 3.57 An example for Ridge Continuity Point (RCP) (Emiroglu, 1998)

In the next step, the algorithm tries to find a new RCP. This time, a window of size 3x3 pixels is centered at the previous RCP instead of the RMP. This operation is repeated until the algorithm reaches a **Ridge Ending Point (REP),** which is on the left side of the RMP. An example of REP with three ridges is shown in Figure 3.54. The algorithm then follows the right side of the RMP the same way it follows the left side. However, this time, the RCP candidates are searched from within the window shown in Figure 3.56. This operation continues until all ridges are processed by the algorithm.

Bifurcation is a point where a ridge forks into two lines. This means, at a bifurcation point, the ridge spacing changes rapidly in a positive or negative way, as shown in Figure 3.58. In the diagram, the value of ridge spacing changes from 6 to 12 at the bifurcation point. If the ridge spacing of an RCP point changes rapidly, the algorithm searches for the second ridge around that region of the image. If there is

another ridge, the ridge point is set to a bifurcation point. Figure 3.58 shows this operation pictorially.

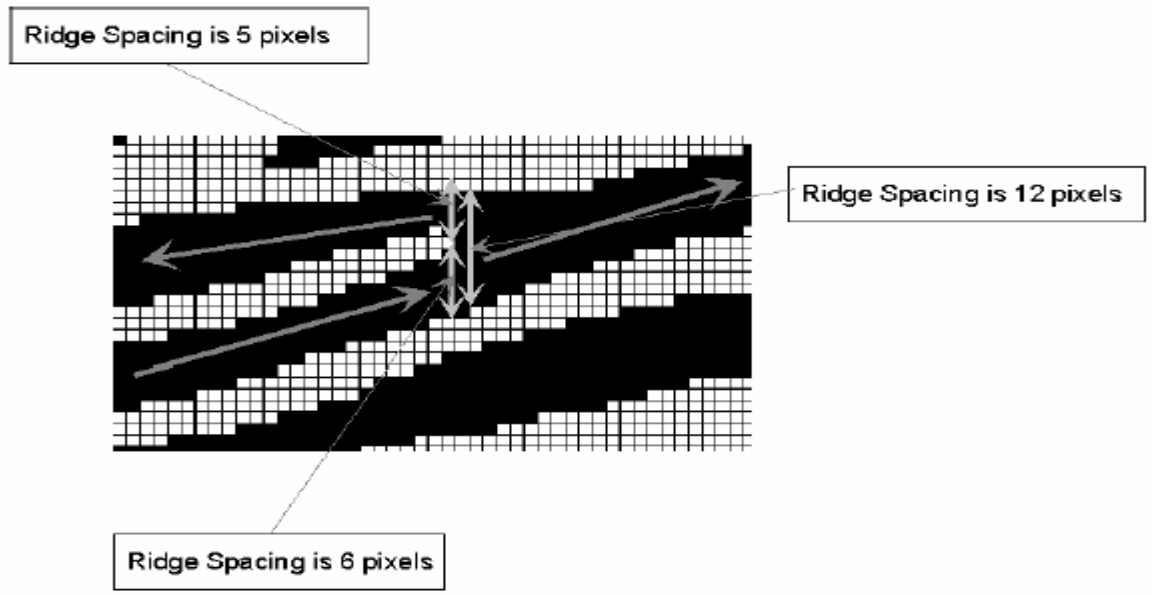


Figure 3.58 The changes of ridge spacing at a bifurcation (Emiroglu, 1998)

The following Figure 3.59 shows a pseudo code for the thinning algorithm presented here.

```
void Ridge_Thinning(){
        Compute the block directional image
        For (i=0;i<n;i++)// n is the number of rows in the image
        {
                For (j=0;j<m;j++) //m is the number of columns in the image
                {
                        Find any black pixel in the image and assign this pixel as
                        Ridge Meeting Point (RMP) which might be a point on
                        the ridge
                        While (RMP=true)
                        {
                                Find Ridge Continuity Points (RCP) on the ridge,
                        the thinned points of the ridge. Check ridge spacing at
                        each step, if ridge spacing changes rapidly, look for a
                        bifurcation point. During this operation, remove the black
                        pixels from original image, which is already processed by
                        the algorithm, if all pixels are processed on the ridge,
                        assign RMP=false else assign RMP=true;
                        }
                }
        }
}
```

Figure 3.59 Pseudo code for thinning algorithm (Emiroglu, 1998)

Figure 3.60 Thinned fingerprint image using Ridge Line Following

3.7 Fingerprint Feature Extraction

Fingerprint feature extraction is also known as minutiae extraction. In this stage, the minutiae are extracted from thinned fingerprint images. In this minutiae extraction stage, only the bifurcations and ridge endings known as basic or primitives from thinned images. Other minutiae such as lakes, pores and hooks are disregarded since they are essentially a combination of these basic minutiae. Each extracted minutia has four attributes: the x-coordinate, the y-coordinate, the minutiae direction and the minutiae type. In this research, two methods are used to extract minutiae: 1) Crossing Number, and 2) Template Based, using a proposed technique, Block Template Extraction. All these techniques use thinned fingerprint images as input image. The fingerprint image in Figure 3.61 is used as input for minutiae extraction.

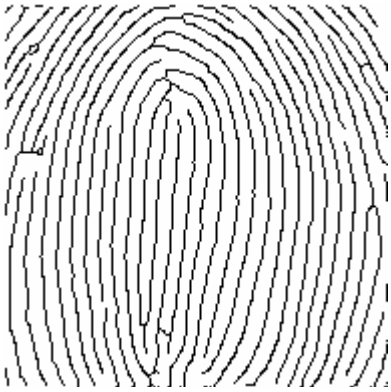


Figure 3.61 Thinned fingerprint image as input

3.7.1 Crossing Number

The minutiae are then extracted using Crossing Number (CN) at a point *P* (B.M. Mehtre, 1993), which is expressed in the following Equation 3.17 as:

$$CN = 0.5\sum_{i=1}^{8}\left|p_i - p_{i+1}\right|, \quad p_9 = p_1$$

(**Equation 3.17**)

where P_i is the pixel value in 3x3 neighbourhood of P. Figure 3.62 shows the 3x3-neighbourhood structure.

P_4	P_3	P_2
P_5	P	P_1
P_6	P_7	P_8

Figure 3.62 A 3x3 neighbourhood of P.

Table 3.4 provides the characteristics of CN:

Table 3.4 The characteristics of Crossing Number.

CN	***Characteristics***
0	Isolated Point
1	End Point
2	Continuing Point
3	Bifurcation Point
4	Crossing Point

In Figure 3.62, P is the reference point to identify ridge bifurcation or ridge ending. Only CN=1 and CN=3 are used in minutiae extraction. Figure 3.63, show a thinned fingerprint image.

0	1	0
1	1	1
0	0	0

Figure 3.63 A thinned fingerprint image.

Value 1 represents the foreground image and value 0 represents the background image. The reading of data begins from P_i and follows to next point by counter clockwise movement until the last value. Figure 3.64 shows the process of reading the data.

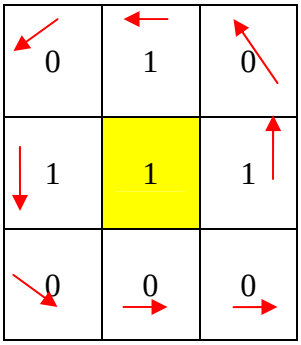


Figure 3.64 Crossing Number counters clockwise movement.

Here is an example of how to calculate the CN value in order to get the minutiae. To identify Figure 3.64 whether contains minutiae or not, CN value is obtained using Equation 3.17.

$$CN = 0.5\sum_{i=1}^{8}\left|P_i - P_{i+1}\right|, P_9 = P_1$$

$CN = 0.5*(|P_1- P_2|-|P_2- P_3|-|P_3- P_4|-|P_4- P_5|-|P_5- P_6|-|P_6- P_7|-|P_7- P_8|-|P_8- P_9|)$

$CN = 0.5*(1+1+1+1+1+0+0+0+1)$

$CN = 0.5*6$

$CN = 3$

CN=3 represents ridge bifurcation. The same process is used in finding the ridge ending.

3.7.2 Template

Emiroglu (1998), Hong (1998) used other template in the minutiae extraction. The process use a mask window of size 3x3 pixels centered at the black pixel. The algorithm finds the number of pixels, *N*, within the window. To be ridge endings, bifurcations and ridge continuity, there are three possibilities:

1. if *N* is 2, the point is regarded as 'ridge endings'
2. if *N* is 3, the point is regarded as 'ridge continuity'
3. if *N* is greater than 3, the point is regarded as 'ridge bifurcations'

A pseudo code for minutiae extraction is shown in Figure 3.65.

```
void main(){
        for (i=0; i<n;i++) // n is the number of rows,
        {
                for (j=0;j<m;j++) // m is the number of columns
                {
                        Start scanning the thinned fingerprint image from the
                        origin point line by line and find any black pixel in the
                        thinned image, a windows size of 3x3 is centered at the
                        point; Count the number of black pixels within the 3x3
                        window (N)

                        if (N==2) assign this point as 'ridge endings'
                        if (N==3) assign this point as 'ridge continuity'
                        if (N>3) assign this point as 'bifurcations'


                }
        }
}
```

Figure 3.65 A pseudo code for minutiae extraction (Emiroglu, 1998)

The three possibilities given above are shown in Figure 3.66 pictorially.

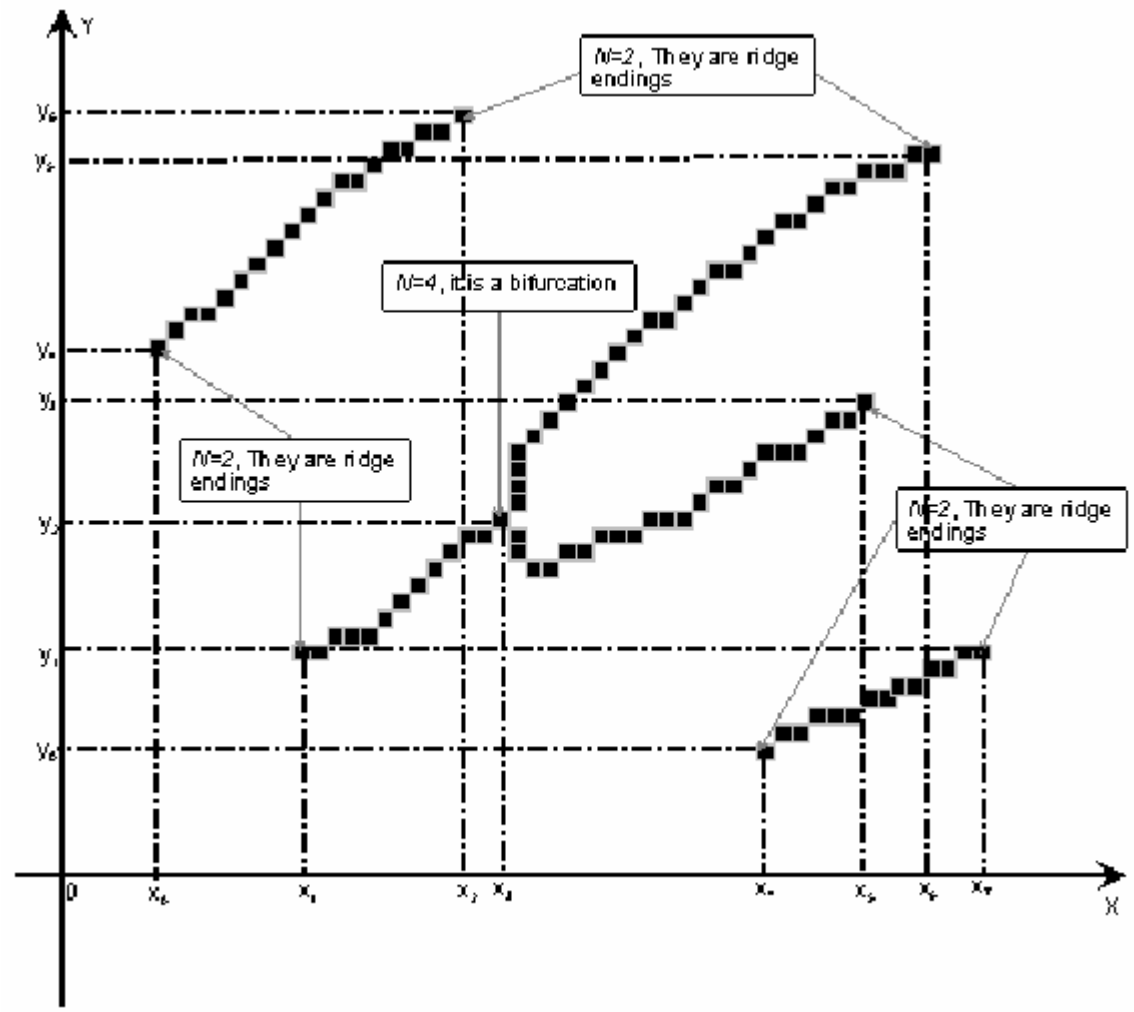


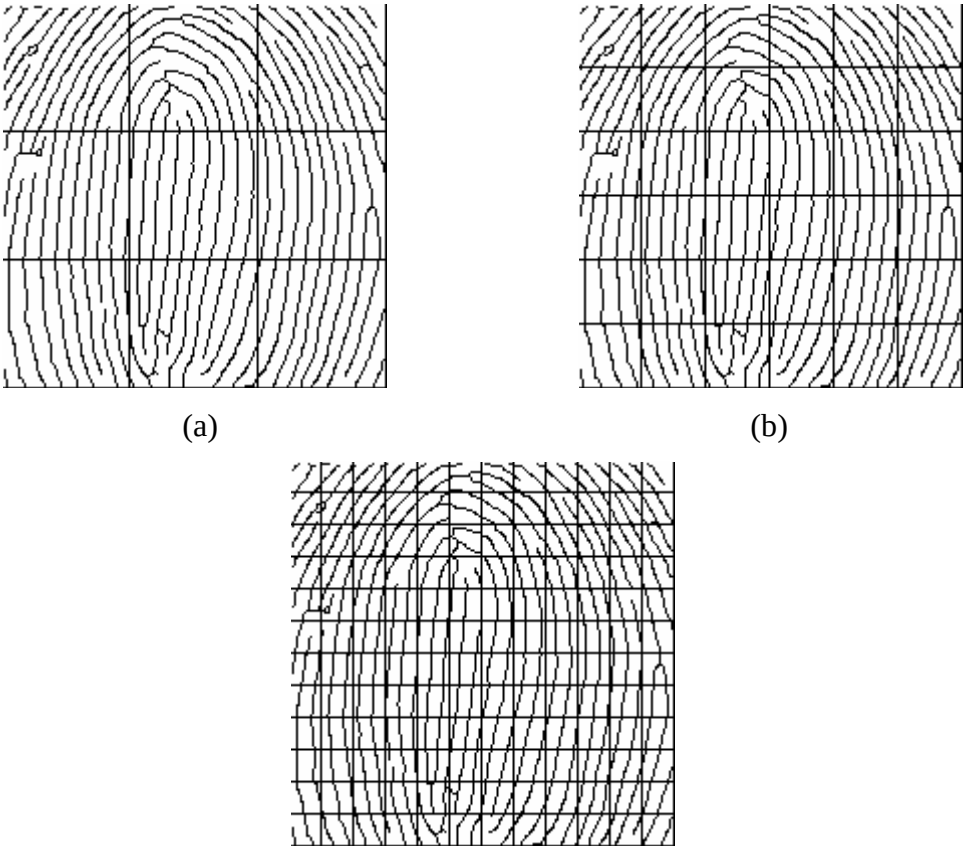
Figure 3.66 The three possibilities in minutiae extraction process (Emiroglu, 1998)

3.7.3 Proposed Block Template Extraction

In this research, the accuracy and the efficiency of the minutiae extraction were enhanced. For minutiae extraction, the Crossing Number algorithm in thinned fingerprint image was used. Using this technique, the thinned fingerprint image using

64x64, 32x32, 16x16 blocks were divided, and in each block, the Crossing Number to extract the minutiae was applied. In each block, we get the minutiae count, type, location and distant between neighbour and reference minutiae.

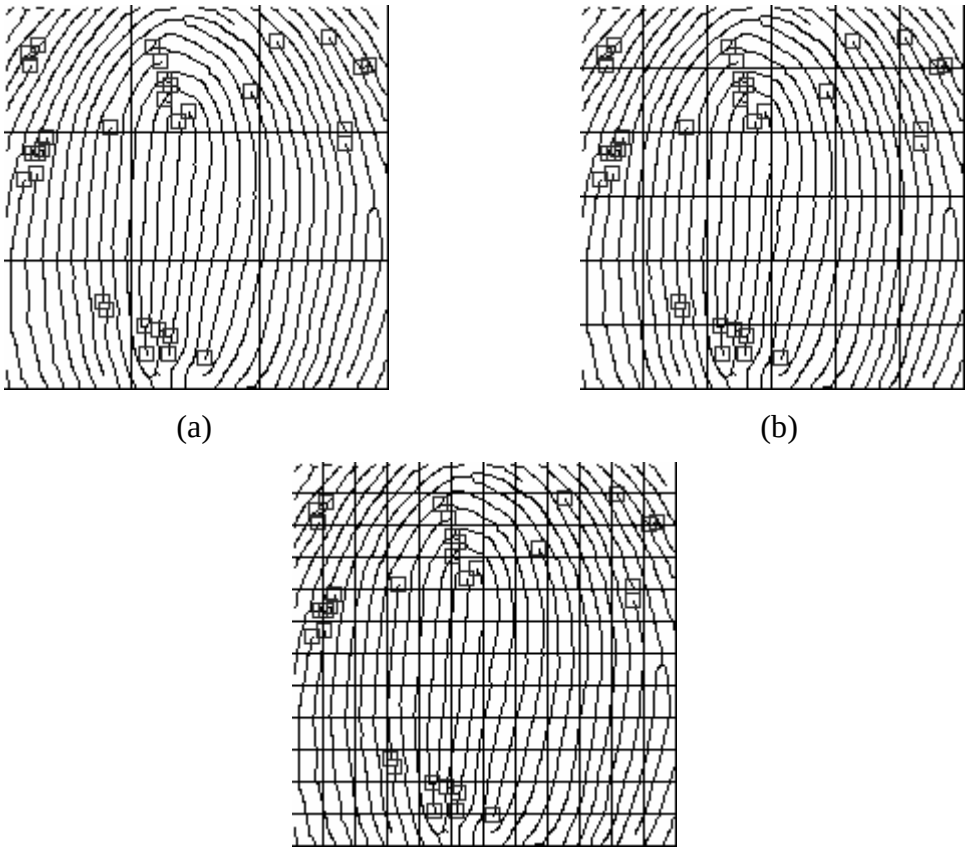
In using this technique, only the block that contains more than 2 minutiae was considered and stored. The block is rejected if it contains less that 2 minutiae. Figure 3.67 shows the thinned fingerprint images divided into various blocks, while Figure 3.68 displays the extracted minutiae for each block.



(a) (b)

(c)

Figure 3.67 Divided thinned fingerprint image using (a) 64x64 (b) 32x32 (c) 16x16 window blocks



(a) (b)

(c)

Figure 3.68 Extracted minutiae in various blocks: (a) 64x64 (b) 32x32 (c) 16x16 window blocks

In this research, the fingerprint image was divided into 64x64 windows which produce 9 regions.

3.8 Fingerprint Matching

Given two minutiae patterns (an input and a template), the minutiae matching algorithm determines whether they are from the impressions of the same finger.

A minutiae matching is essentially a point pattern matching problem. The similarity of two minutiae patterns is determined by the total number of corresponding minutiae and the decision is made by comparing the value of similarity with a pre-specified *threshold*. Formally, it can be stated as follows: Let $P = ((x_1^P, y_1^P, \theta_1^P), ..., (x_M^P, y_M^P, \theta_M^P))$ and $Q = ((x_1^Q, y_1^Q, \theta_1^Q), ..., (x_N^Q, y_N^Q, \theta_N^Q))$ denote the M minutiae in the template and the N minutiae in the input image, respectively. Find the number, M_{pair}, of the corresponding pairs between P and Q and compare it against a threshold value $T_{minutiae}$.

In the ideal case, if (i) the correspondence between the template and input is known, (ii) there are no deformation such as translation, rotation and deformations between them, and (iii) each minutiae present in a fingerprint image is exactly localized, then minutiae matching is only a trivial task of counting the number of spatially matching pairs between the two fingerprints and comparing it against a pre-specified threshold value.

In practice, determining whether two minutiae patterns extracted from two fingerprint impressions, possibly separated by a long duration of time, are indeed from the same finger, is an extremely difficult problem. The difficulty can be attributed to two primary reasons. First, even though the test and template minutiae patterns are indeed mated pairs, the correspondence between the test and template minutiae patterns is generally not known. Secondly, the imaging system presents a number of peculiar and challenging situations, of which some are unique to fingerprint image capture scenario as follows:

(i) Inconsistent contact: The act of sensing distorts the finger. Based on the pressure and contact of the finger on the glass platen, the three dimensional shape of the finger gets mapped onto the two dimensional surface of the glass platen. Typically, this mapping function is uncontrolled and results in different fingerprint across the impressions.

(ii) Non-uniform contact: The ridge structure of a finger would be completely captured if ridges of the part finger being imaged are in complete optical

contact with the glass platen. However, dryness of the skin, skin disease, sweat, dirt, humidity in the air all confound the situation, resulting in a non-ideal contact situation; some parts of the ridges may not come in complete contact with the platen and regions representing some furrows may contact with glass platen. This results in noisy low contrast images, leading to either spurious minutiae or missing minutiae.

(iii) Irreproducible contact: Manual works, accidents, etc. inflict injuries to the finger, thereby, changing the ridge structures of finger either permanently or semi-permanently. This may introduce additional spurious minutiae.

(iv) Feature extraction artefacts: The features extraction algorithm is imperfect and introduces measurements errors. Various image processing operations might introduce in consist biases to perturb the locations and orientation estimates of the reported minutiae from their grey scale counterparts.

(v) The acts of sensing itself add noise to the image. For example, residues are leftover from the previous fingerprints capture. A typical imaging system distorts the image of the object being sensed due to imperfect imaging conditions.

In light of the operational environment mentioned above, the design of the minutiae matching algorithms needs to establish and characterize a realistic model of the variations among representations of mated pairs. This model should include the properties of interest listed below:

1. The finger may be placed at different locations on the glass platen resulting in a (global) translation of the minutiae of the test representation from those in template representation.
2. The finger may be placed in different orientations on the glass platen resulting in a (global) rotation of the minutiae of the test representation from those in template representation.
3. The finger may exert a different (average) downward normal pressure on the glass platen resulting in a (global) spatial scaling of the minutiae of the test representation from those in template representation.
4. The finger may exert a different (average) shear force normal pressure on the glass platen resulting in a (global) shear transformation

(characterize by a shear direction and magnitude) of the minutiae of the test representation from those in template representation.

5. Spurious minutiae may be present in both the template as well as the test representation.
6. Genuine minutiae may be absent in the template or test representations.
7. Minutiae may be locally perturbed from their true location and perturbation may be different for each individual minutiae. (The magnitude of such perturbations. However, it is assumed to be small and within a fixed number of pixels.)
8. The individual perturbations among the corresponding minutiae could be relatively large (with respect to ridge spacing) but the perturbations among pairs of the minutiae are spatially linear.
9. The individual perturbations among corresponding minutiae could be relative (with respect to ridge spacing) but the perturbations among pairs of the minutiae are spatially non-linear.
10. Only a (ridge) connectivity preserving transformation could characterize the relationship between the test and template presentation.

There are two ways in doing fingerprint matching: firstly, the Template Matching and secondly, a new proposed technique, Block Template Matching.

3.8.1 Template Matching

The template matching algorithm attempts to match a set of minutiae obtained from a scanned fingerprint of a previously stored template. In this algorithm, classic ridge counting is not performed as this could increase the possibility of a false rejection without profoundly affecting the false acceptance rate. The matching of fingerprint is based on the minutiae features. Each minutiae feature of a fingerprint image is described as:

(TYPE, X, Y, DIRECTION)

Here TYPE represents the minutiae type, with “1” for endpoint and “2” for bifurcation. X and Y indicate the position of the minutiae in the image. DIRECTION is the direction of that minutia.

For example, two minutiae set, S1 is taken from database or stored fingerprint and S2 is from test fingerprint. S1 and S2 are said to be paired if their minutiae type are the same, their position and direction are close since the fingerprint images are taken in affixed windows and it is assumed that there are small translation and rotation. If $f^1 \in S1, f^2 \in S2$ and

$$TYPE(f^1) = TYPE(f^2)$$
$$DIST(f^1, f^2) \le D_f$$
$$ANGLE(f^1, f^2) \le A_f$$

then (f^1, f^2) is a pair of matched minutiae features. Here D_f and A_f are maximum tolerance for translation and rotation respectively. Here it is assumed there is so little rotation because the window on fingerprint scanner is just the size of one fingerprint and one presses his/her finger almost in the same direction: Set A_f to $\pi/6$ and D_f to 60.

Let S_m be a set of matched pairs. Each element in S_m has the form (f_i^1, f_i^2) where f_i^1 is from S1 and f_i^2 is from S2. There are two constraints to S_m. All f_i^1 and f_i^2 in S_m should be different. These mean that each minutia in S1 or S2 should not be matched more than once. The following condition must also be satisfied if (f_1^1, f_1^2) and (f_2^1, f_2^2) are two elements in S_m.

$$| DIST(f_1^1, f_1^1) - DIST(f_2^2, f_2^2) | < \varepsilon$$

The value of ε is small, which takes the value of 15 in our system. In order to match two fingerprints, a set $S_{\max}$ with the maximum number of paired minutiae features is to be found. The procedure to do this is as follows:

1. Let S_m be empty.
2. Select f^1 from S1, f^2 from S2. If f^1 and f^2 can be matched and (f^1, f^2) can be added to S_m, add it.
3. Repeat step 2 until no pair could be added to S_m. Check whether the current number of elements in S_m is the maximum. If it is, save S_m as the current $S_{\max}$. Backtrack to search the other combinations.
4. Finally, $S_{\max}$ is found.

Let N_1, N_2 be the number of elements in S1 and S2 respectively, N_m be the number of elements in $S_{\max}$. The similarity measure M between two fingerprint images is written as

$$M = \sqrt{\frac{N_m x N_m}{N_1 x N_2}}$$

The similarity measure M for two images from the same fingerprint is close to 1. In practice, if the calculated M is bigger than a predefined reasonable threshold, then it can be said that two images originate from the same fingerprint.

3.8.2 Proposed Block Template Matching

The accuracy of the Template Matching algorithm using block technique was enhanced in this research. The technique used was the master template obtained from Block Template Extraction for matching process.

In this experiment, 64x64 blocks for minutiae matching were chosen. Using this algorithm, the minutiae properties in each block were checked. The algorithm follows the steps below:

1. Get the total block of minutiae collection from master and live template.
2. In each block,
    a. Get a total minutiae count
    b. Get the minutiae type
    c. Get the minutiae distance to the reference minutiae
3. Match the block details from live template with the master template.
4. If they match, proceed to next block live template.
5. If not, repeat step 2 until all blocks in live template have been processed. The block in live template can only be used once. If one block in master template has been matched with live template, that block cannot be used for further matching processes.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Overview

This chapter discusses the experiment procedures, dataset, and findings of the study. This is followed by the discussion of results and conclusion. A total of 1000 fingerprint images were collected for the experiment using SecureTouch®2000 optical fingerprint scanner.

4.2 Dataset

Using these fingerprint images, a fingerprint database was developed to be our database and therefore, we do not have to rely on NIST Database. Each fingerprint image is an 8-bit grey scale with a size of $320x320$. The dataset consist of 5 fingerprint classes which are whorl, arch, left loop, right loop and tented. The images in the dataset are of inconsistent quality, with some of them containing noise. These inconsistencies lead to the introduction of false minutiae in the minutiae extraction process. For the fingerprint with high quality image, the ridges and

furrows are very clean, with a very minimum level of noise. From 1000 fingerprint images, only 500 fingerprint images were chosen for experiment. Figure 4.1 shows the fingerprint images for each class, while Figure 4.2 shows fingerprint image qualities of (a) a low quality fingerprint image, and (b) a high quality fingerprint image.

(a)

(b)

(c)

(d)

(e)

Figure 4.1 Fingerprint images for each class: (a) whorl, (b) arch, (c) left loop, (d) right loop and (e) tented.

(a) (b)

Figure 4.2 Fingerprint image qualities: (a) low quality; (b) high quality

4.3 Experiment

The main objective of the experiment is to find the best solution and method for the Automated Fingerprint Identification System (AFIS). Through this experiment, the best method for each stage in AFIS is proposed.

In order to find the best method, a comparative study on each stage in AFIS was carried out. The evaluation of the results of each algorithm in each stage was also conducted.

In the first stage, the fingerprint pre-processing and enhancement stage, three approaches for filtering the noise and to improve the clarity of fingerprint image were used, namely are as follows: i) Smoothing Filtering, ii) Sharpening Filter, and iii) Histogram Modelling. In our search to find the algorithm that produces the image of high quality, the results for each stage were compared. The Histogram

Equalization technique was used to remove the noise and improve the sharpness and clarity of fingerprint images, and the High-Pass Filtering technique to enhance the ridges and furrows details in getting a better directional image.

In the Directional Image computation, a comparative study between the Mehtre based algorithm and the Least Mean Square Orientation Estimation Algorithm was conducted. 8x8 windows to estimate ridge direction on each block were utilised. The directional image is then filtered using smoothing filter to improve the image.

The enhanced fingerprint image was transformed into frequency domain using the Fast Fourier Transform and filtered by the Directional Fourier Filtering to get a new grey scale image. Using the Regional Average Thresholding on the grey scale image, a binary image was produced.

Once the Ridge Line Following was applied on the binary image, a thinned image was obtained. In getting the minutiae, the Crossing Number algorithm was utilized on the thinned image. Finally, a Template Matching was used to compare the extracted minutiae from the live fingerprint image with the database.

4.4 Results

The results from the experiment are discussed separately for each stage, namely the fingerprint pre-processing and enhancement stage, the directional image stage, the reconstruction stage, the segmentation stage, the thinning stage, the feature extraction stage and the matching stage.

4.4.1 Fingerprint Pre-Processing and Enhancement

In this segment, all the figures for each technique that will be discussed are shown. The figures are divided into two groups; i) fingerprint pre-processing and ii) fingerprint enhancement. Figure 4.3 below shows the original fingerprint image w that was chosen to undergo the two processes.

Figure 4.3 Original fingerprint images before pre-processing and enhancement

Figure 4.4 (a – f) below shows the different filtered fingerprint images according to different filters: (a) Averaging filter, (b) Minimum filter, (c) Median filter, (d) Maximum filter, (e) Low-Pass filter, and (f) Hexagonal Grid filter, while Figure 4.4 (g) shows the Histogram Equalization image.

(a)

(b)

(c)

(d)

(e)

(f)

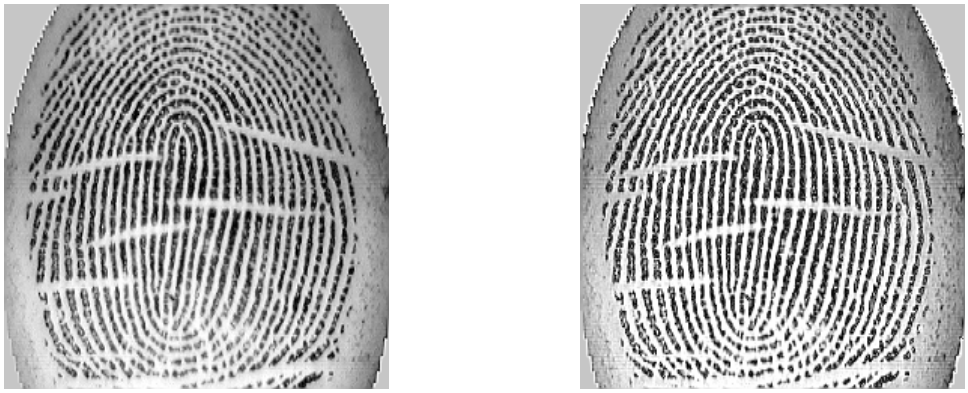
(g)

Figure 4.4 Filtered fingerprint images: (a) Averaging filter, (b) Minimum filter, (c) Median filter, (d) Maximum filter, (e) Low-Pass filter, (f) Hexagonal Grid filter, (g) Histogram Equalization.

The main objective of the fingerprint pre-processing stage is to get a clear fingerprint images without any noise. From the different techniques available, the Histogram

Equalization technique was selected because it can produce clear and sharper fingerprint image with successful noise removal element. Although the Averaging, Low-Pass and Hexagonal Grid techniques do remove the noise level, they also blur the fingerprint images, which decrease the image sharpness. The other filter, the Median filter can produce clear image without blurring. However, the image produced is not sharp enough compared with the Histogram Equalization output image. Meanwhile, the Minimum and Maximum techniques can produce thick and thin fingerprint images respectively.

The objective of fingerprint enhancement is to preserve the fingerprint details while increasing the differential ridges and furrows to get a better directional image in the next stage. From the original fingerprint in Figure 4.3, the fingerprint images below show the filtered fingerprint images after going through the High-Pass filter in Figure 4.5 (a) and the High-Boost filter in Figure 4.5 (b).



(a) (b)

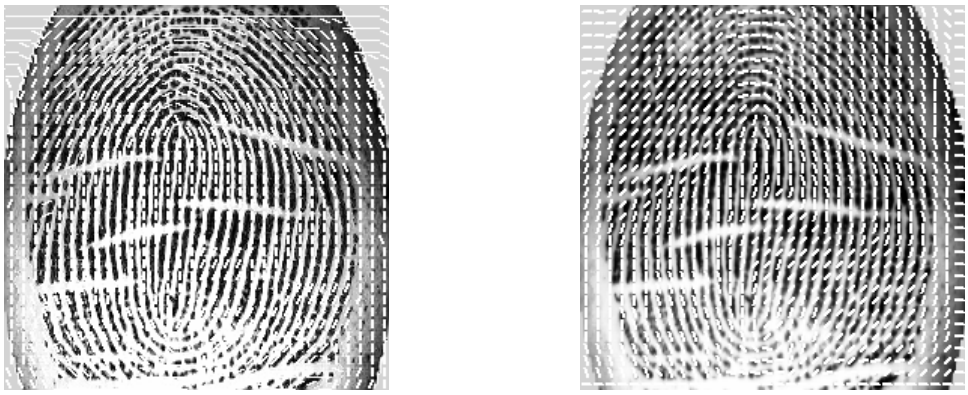
Figure 4.5 Filtered fingerprint images; (a) High-Pass filter, (b) High-Boost filter.

From these two fingerprint images, the High-Pass technique produces fingerprint image that is less sharp than the High-Boost technique. The High-Boost technique produces a very sharp and clear fingerprint image. However, it also increases the unwanted noise. The High-Pass technique is more suitable in this case.

As a conclusion, from the experimental results, the Histogram Equalization technique was chosen to be used in the Fingerprint Pre-processing and Enhancement stage, instead of the High-Pass technique. The quality of output image using the Histogram Equalization is better than the High-Pass technique. With the chosen technique, the clarity of fingerprint ridges and furrows is improved without any distortion and the image produced is very smooth and clear. Although using High-Pass technique increases the ridges and furrows sharpness tremendously, the quality of the image produced is less than the image produced by the Histogram Equalization. Using the Histogram Equalization, wrong direction in directional image can be fixed and improved.

4.4.2 Directional Image

The main objective in this stage is to produce directional image with improved clarity of ridges and furrows and accurate direction of ridges orientation. Two techniques were used: i) the Mehtre based concept, ii) the Least Mean Square Orientation Estimation. In first the technique, (refer to Figure 3.26), the best directional image is produced by using the Median Filtering. In Figure 3.26(b), the directional image is mapped onto the original image, and it is very similar to the original ridges direction. Figure 4.6 below shows the two directional images using i) the Mehtre based concept, and ii) the Least Mean Square Estimation.



(i) (ii)

Figure 4.6 Two directional images using: i) the Mehtre based concept, and ii) the Least Mean Square Estimation

4.4.3 Fingerprint Reconstruction

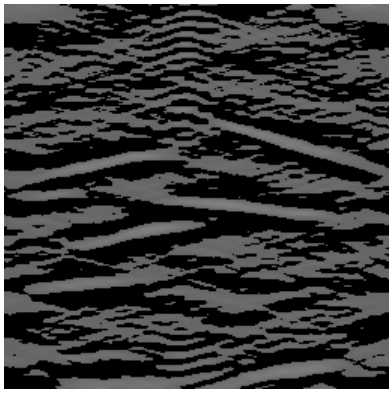
Accurate Directional Image is very important to the next stage, the Fingerprint Reconstruction stage as it produces accurate new fingerprint image. In Figure 4.6 above, two different directional images were mapped onto the same fingerprint using, i) the Mehtre based concept and ii) the Least Mean Square Estimation. As can be seen from these two directional images, Figure 4.6(i) is better than Figure 4.6(ii), in terms of sharpness. As a conclusion, the Mehtre based concept produces better images than the Least Mean Square Estimation.

4.4.3.1 Mehtre Based as Directional Image

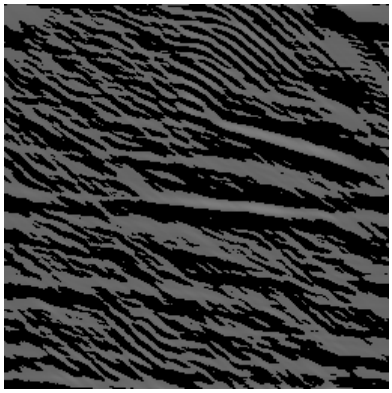
Figure 4.7 shows these 8 pre-filtered fingerprint images of the original fingerprint image (a).

(a)

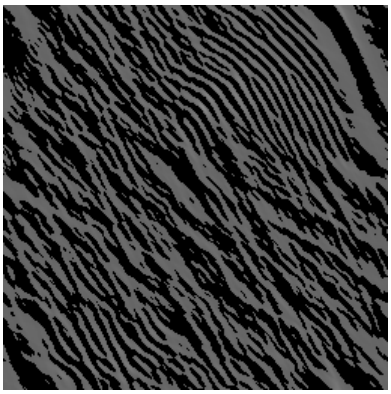
(b)



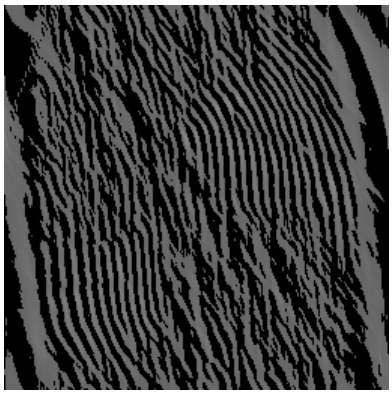
(c)



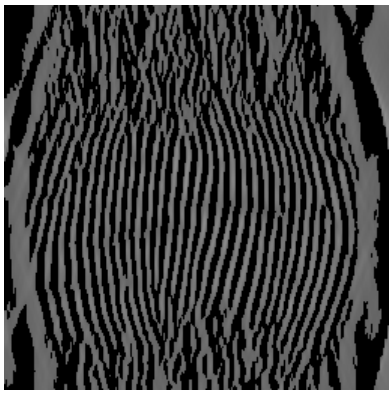
(d)



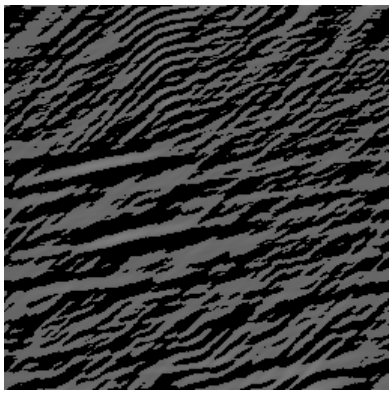
(e)



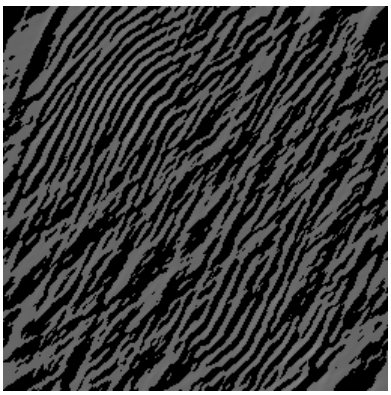
(f)



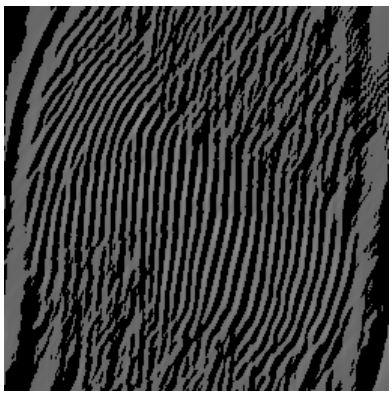
(g)



(h)



(i)



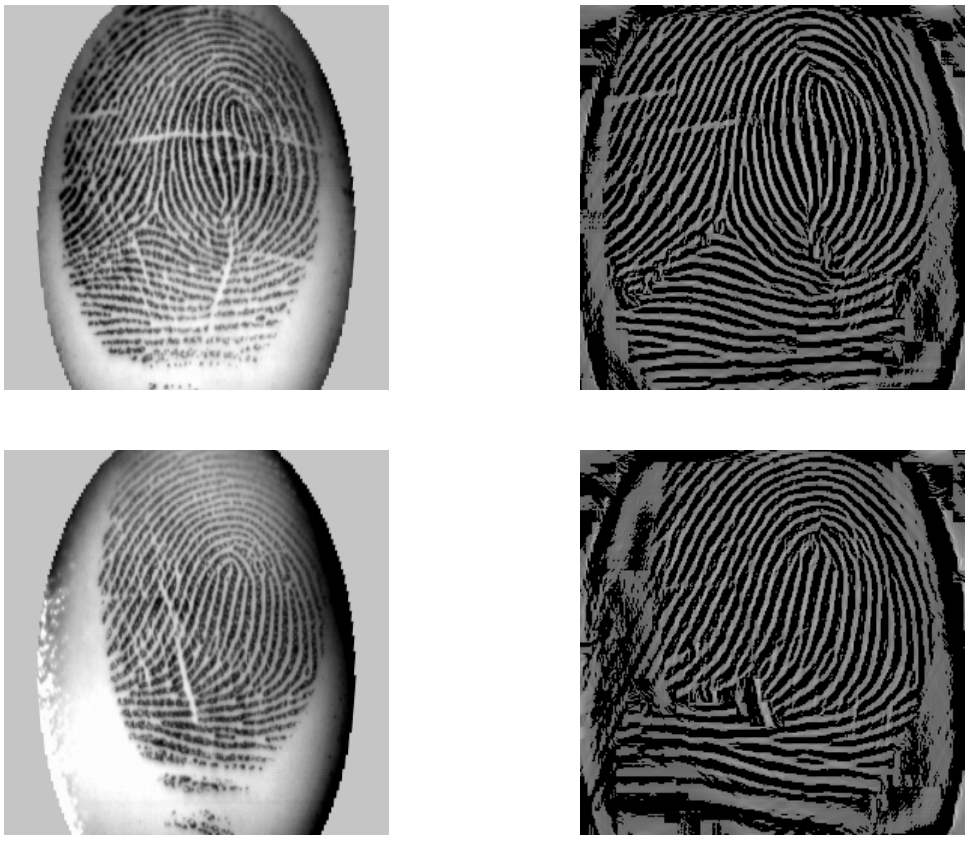
(j)

Figure 4.7 8 set pre-filtered fingerprint images: a) Original image, b) Directional image using Mehtre based, c) 0^0, d) 22.5^0, e) 45^0, f) 67.5^0, g) 90^0, h) 112.5^0, i) 135^0, j) 157.5^0

From these filtered fingerprint images, a new fingerprint image is generated. Figure 4.8 shows a reconstructed fingerprint image.

Figure 4.8 A reconstructed fingerprint image obtained from 8 set pre-filtered fingerprint images above.

Figure 4.9 below shows a successful fingerprint image reconstruction and its original image.



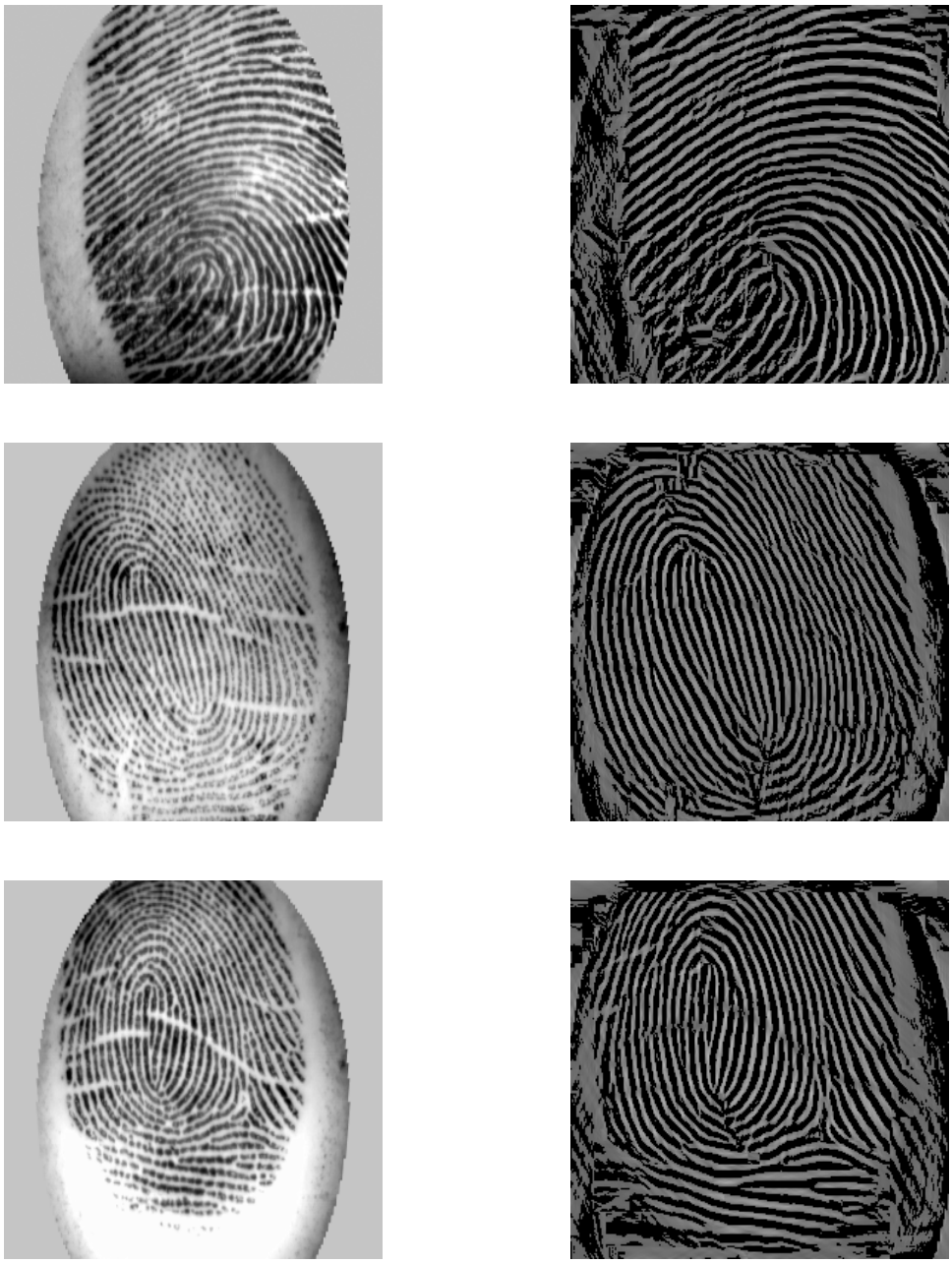


Figure 4.9 A successful fingerprint image reconstruction

Figure 4.10 shows the original fingerprint image which has vertical scars is reconstructed into a new fingerprint image.

Figure 4.10 Fingerprint image containing vertical scars and the newly reconstructed image.

Figure 4.11 shows an original fingerprint image, which has horizontal scars, is reconstructed into a new fingerprint image.

Figure 4.11 Fingerprint image containing horizontal scars with the newly reconstructed image.

Figure 4.12 shows an original fingerprint image, which has a vertical and horizontal scars are reconstructed into a new fingerprint image.

Figure 4.12 Fingerprint image containing both vertical and horizontal scars and the newly reconstructed fingerprint image.

Figure 4.13 shows a clear original fingerprint image is reconstructed into a new fingerprint image.

Figure 4.13 Clear fingerprint image and the new reconstructed image

Figure 4.14 shows a damaged fingerprint image and after the reconstruction process.

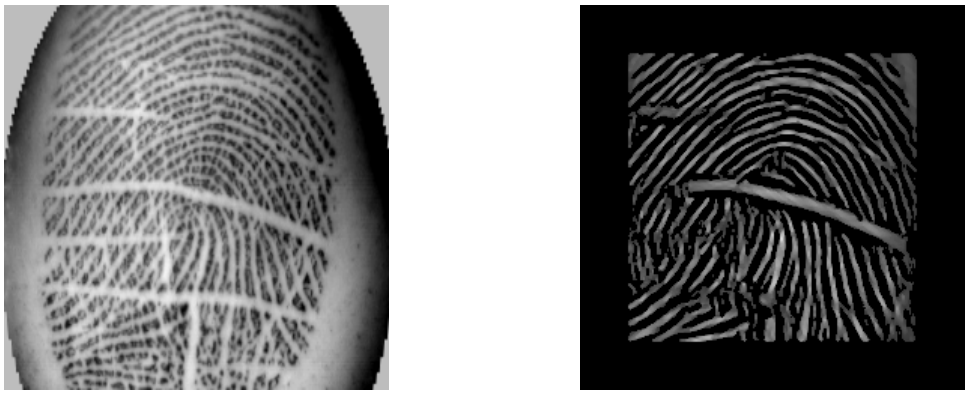
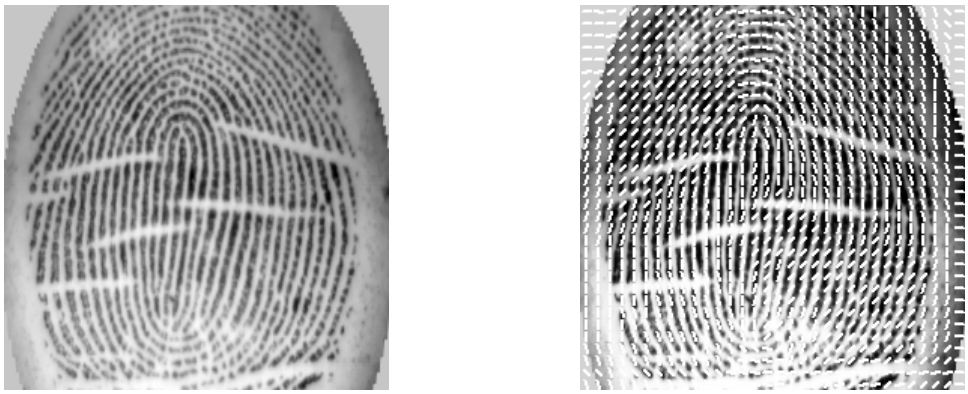


Figure 4.14 Damaged fingerprint image and after reconstruction image

4.4.3.2 Least Mean Square Orientation Estimation as Directional Image



(a) (b)

(c)

Figure 4.15 New reconstructed fingerprint image using the Least Square Orientation Estimation Algorithm: (a) Original fingerprint image, (b) Directional Image, and (c) Fingerprint image reconstruction

From Figure 4.8 and Figure 4.15, a comparison between the Mehtre Based and the Least Square Estimation algorithm can be made. From both figures, we can see that with the Mehtre Based algorithm, the reconstructed fingerprint image is free from horizontal or vertical scars. This will decrease the number of false minutiae created. With the Least Square Estimation algorithm, the reconstructed image is very poor and of low quality, and this will lead to the creation of false minutiae. The failure and successful creation of a new fingerprint image depends on the quality of the directional image. If the quality of the directional image is poor, the new fingerprint image is also poor, and vice versa. Each directional image element plays an important role in the fingerprint reconstruction process. If it is corrupted or wrong, the new image is also corrupted. Figure 4.16 shows the effects of the directional image to fingerprint image reconstruction process.

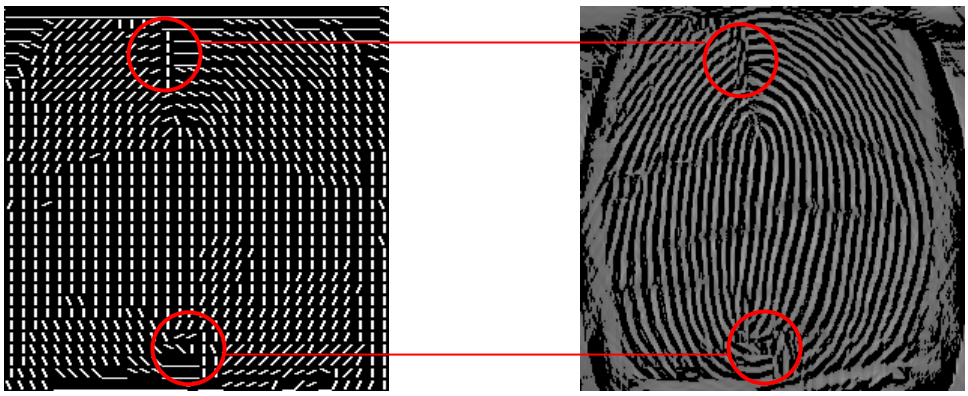


Figure 4.16 Wrong elements in the directional image will result in new fingerprint image with wrong elements.

This technique was tested with 100 samples of fingerprint images. Table 4.1 shows the results of the experiment of the fingerprint reconstruction process.

Table 4.1 The results of fingerprint reconstruction using Mehtre Based and Least Square Estimation as directional image.

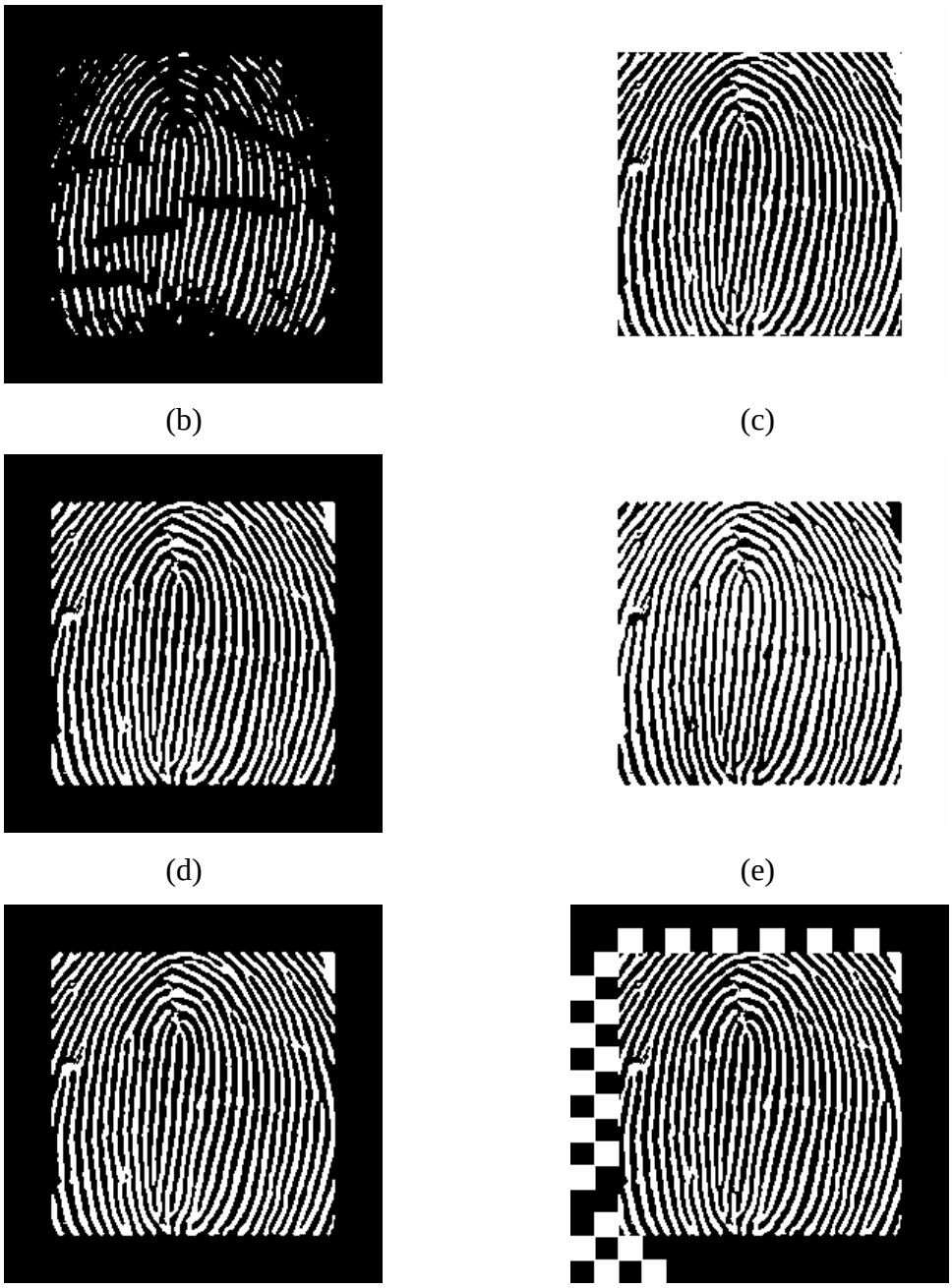
	Mehtre Based		Least Square Estimation	
	Success	Failure	Success	Failure
100 Fingerprint Images	91 %	9 %	67 %	33 %

As a conclusion, fingerprint image reconstruction using the Mehtre Based is better than the Least Square Estimation.

4.4.4 Fingerprint Segmentation

The objective of the fingerprint segmentation stage is to produce a binary image which separates ridges and furrow into black and white image. Here, all the images for each technique are listed for comparison and discussion purposes. The comparisons are made based on the quality of output image after segmentation. The white area represents fingerprint furrows, while the black area represents fingerprint ridges. Figure 4.17 shows the segmentation of a fingerprint image using six different techniques (b) to (g), namely; (a) reconstructed fingerprint image as input image, (b) Global Thresholding, (c) Regional Average thresholding, (d) Histogram Peak, (e) Histogram Valley, (f) Histogram Adaptive, (g) Niblack Binarization thresholding.

(a)



(b) (c)

(d) (e)

(f) (g)

Figure 4.17 The fingerprint segmentation: (a) Original image produced from image reconstruction; (b) Global Thresholding; (c) Regional Average Thresholding; (d) Histogram Peak Thresholding; (e) Histogram Valley Thresholding; (f) Histogram Adaptive Thresholding; and (g) Niblack Binarization.

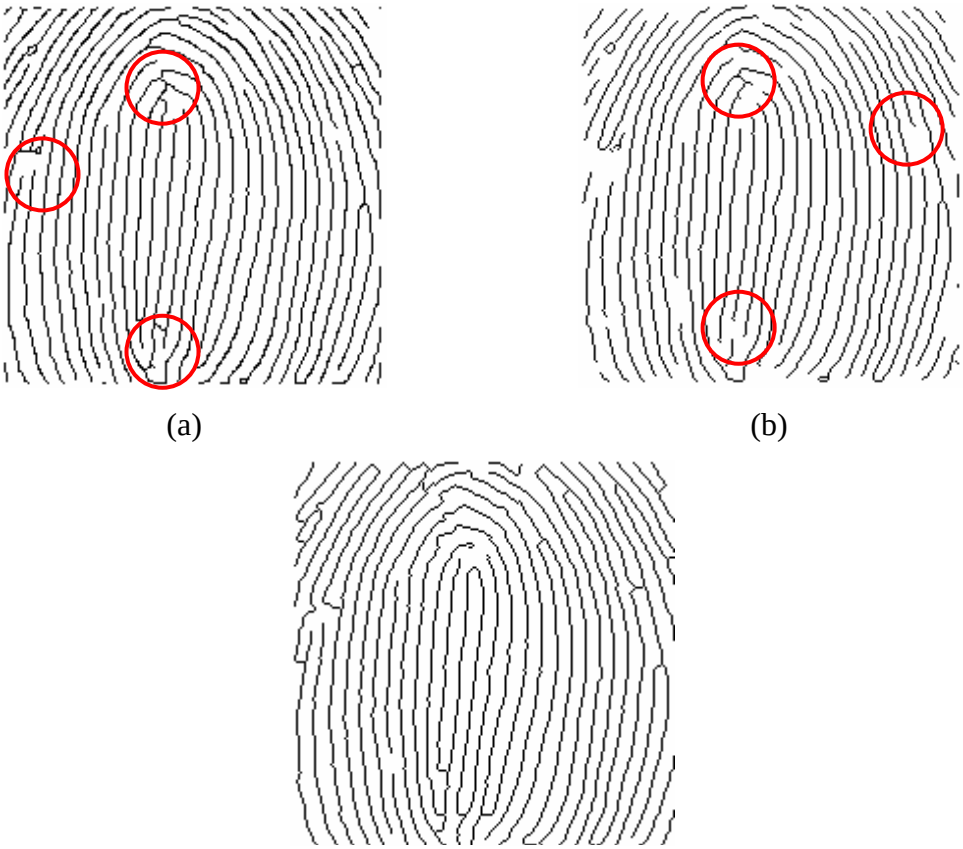
After experimenting with these various techniques of fingerprint segmentation, the Regional Averaging Thresholding is found to be the best method in producing binary

image with the highest quality. Highest quality is determined by looking for fine segmented image which has clear ridges and furrows separation.

4.4.5 Fingerprint Thinning

The main objectives of this stage are to obtain the skeleton image with a 1-pixel in width and to preserve the original connectivity of each ridge. Three different methods to determine the best thinning technique for fingerprint image have been employed. The methods are the Two-Way Pass Thinning, Fast Thinning Algorithm and Ridge Line Following Thinning. With the same segmented fingerprint image as input, and by means of each method, various thinned fingerprint images of different qualities were obtained. Here we list down all thinned fingerprint images from each technique, and comparison is made based on the quality of the output image. Figure 4.18 shows: (a) the original image after segmentation, (b) Two-Ways Pass Thinning, (c) Fast Thinning Algorithm, and (d) Ridge Line Following Thinning.

(a)



(a) (b)

(d)

Figure 4.18 Thinned fingerprint images: (a) Original image after segmentation, (b) Two-Way Pass Thinning, (c) Fast Thinning Algorithm, (d) Ridge Line Following Thinning.

From the comparison of the images, it is found that the thinned fingerprint images from the Two-Way Pass and the Fast Thinning Algorithm techniques, both produce poor quality skeleton image. With the Two-Way Pass technique, the thinned fingerprint image produced is not one pixel wide and fails to discard noise. While with Fast Thinning Algorithm, a clean fingerprint image was produced. However, this technique also removes the original minutiae. It is only through the Ridge Line Following technique, a very clean thinned fingerprint image was obtained, while at the same time preserving the original minutiae.

As a conclusion from the experiment conducted, the Ridge Line Following technique was chosen as the best method in producing thinned fingerprint image.

4.4.6 Fingerprint Feature Extraction

The main objective of fingerprint feature extraction stage is to get accurate minutiae from thinned fingerprint image. In this stage, different techniques, such as the Crossing Number, Proposed Block Template Extraction and Template techniques were used and compared in order to determine the best technique to choose for the process of fingerprint feature extraction. Figure 4.19 below shows a thinned fingerprint image. Ridge ending extraction is shown in Figure 4.22 using Crossing Number and Figure 4. 24 using Template. Ridge bifurcation extraction is shown in Figure 4.23 using Crossing Number, and Figure 4.25 using Template. Meanwhile, Figure 4.26 shows the minutiae extraction using Block Template Extraction of 64x64 block with same image.

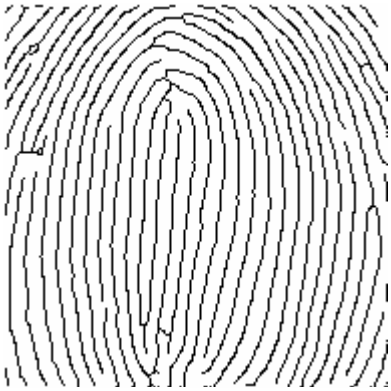


Figure 4.19 Fingerprint thinned image

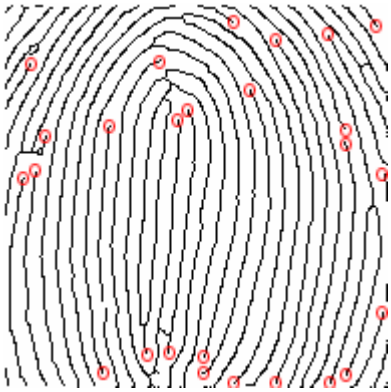


Figure 4.20 Ridge ending extraction using manual technique

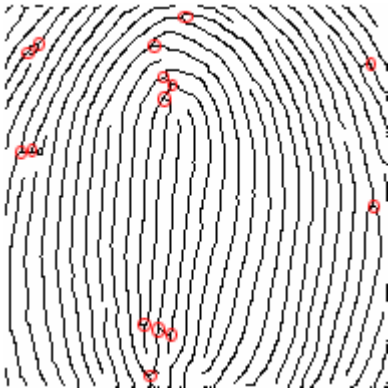


Figure 4.21 Ridge bifurcation extraction using manual technique

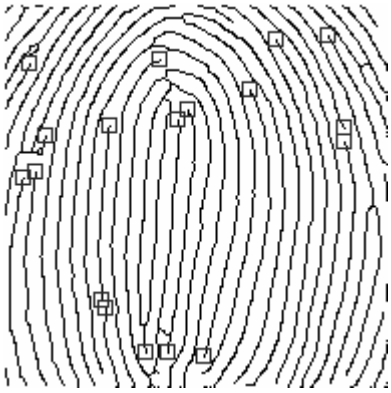


Figure 4.22 Ridge ending extraction using Crossing Number

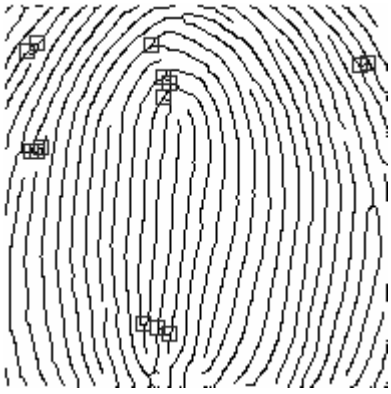


Figure 4.23 Ridge bifurcation extraction using Crossing Number

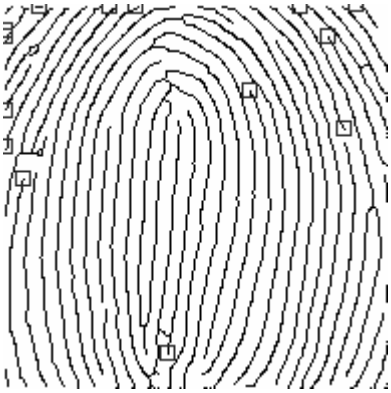


Figure 4.24 Ridge ending extraction using Template

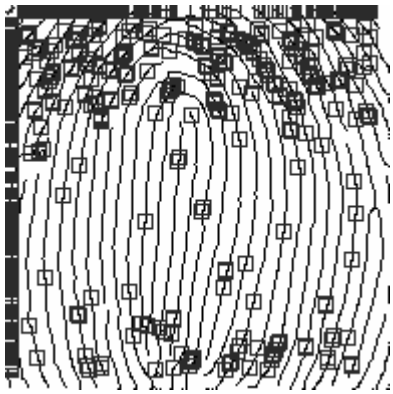


Figure 4.25 Ridge bifurcation extraction using Template

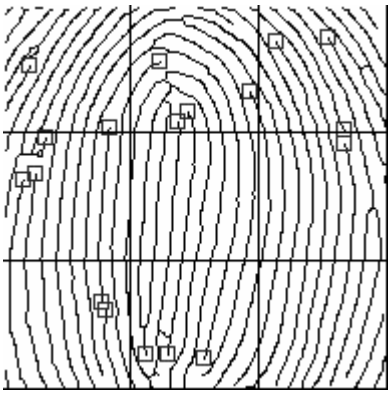


Figure 4.26 Ridge ending extraction using Proposed Block Template Extraction

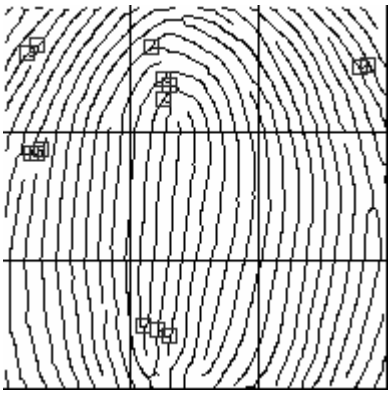


Figure 4.27 Ridge bifurcation extraction using Proposed Block Template Extraction

From the comparisons of the different Figure 4.19 - Figure 4.25 above, we can conclude that by using the Crossing Number technique, the ridge and bifurcation ending or minutiae extraction matches with the manual technique. However, using the Template technique, the minutiae extraction fails. With the Template technique, many false minutiae were produced. This happened because the Template technique needs a one pixel wide thinned fingerprint image to succeed. The Crossing Number technique does not depend heavily on the thinnest fingerprint image. Although the image is not one pixel wide, it still extracts accurate minutiae. Meanwhile, the results from the Proposed Block Template Extraction technique equal the results from the Crossing Number technique. This can be seen by comparing Figure 4.26 with Figure 4.22, and Figure 4.27 with Figure 4.23. As a conclusion, using the Proposed Block Template Extraction will produce accurate minutiae and shorten the processing time.

4.4.7 Fingerprint Matching

The Template Matching and Proposed Block Template Matching technique in the fingerprint matching stage were used. A total number of 274 fingerprint samples were tested in the experiment, consisting of 90 whorl samples, 49 arch samples, 20 tented arch samples, 44 left loop samples, 71 right loop samples and other fingerprint

class. In Template Matching, the maximum tolerance value for translation is D_f =10, and the rotation tolerance is A_f = 30. In Block Template Matching, the translation tolerance is D_f =10.

As a result of the minutiae extraction from the dataset using Template Matching technique, the overall results show a successful matching rate of 84% and false matching rate of 15%, without failed matching. While for the Block Template Matching, the successful matching is 91%, 7.2% for false matching and 4.5% for failed matching. From these results, the Block Template Matching has improved the matching results by 7%. The results for Template Matching and Block Template Matching are shown in Table 4.2 and Table 4.3.

Table 4.2 Average minutiae successfully extracted from dataset using Template Matching Technique

Fingerprint Class	**Template Matching**		
	True Matching	**False Matching**	**Fail Matching**
Arch	81.79 %	18.21%	-
Tented Arch	86.98%	13.02%	-
Left Loop	85.33%	14.67%	-
Right Loop	82.11%	17.89%	-
Whorl	85.20%	14.8%	-

Table 4.3 Average minutiae successfully extracted from dataset using Block Template Technique

Fingerprint Class	**Block Template Matching**		
	True Matching	**False Matching**	**Fail Matching**
Arch	89%	9%	2%
Tented Arch	89%	10%	1%
Left Loop	95%	3%	2%
Right Loop	90%	8%	2%
Whorl	92%	6%	2%

Based on the results of the experiment, the limitations of Template Matching were determined. This matching technique depends heavily on the location and the direction of the extracted minutiae. So when the translation or rotation was made on the fingerprint images, the matching results may go wrong.

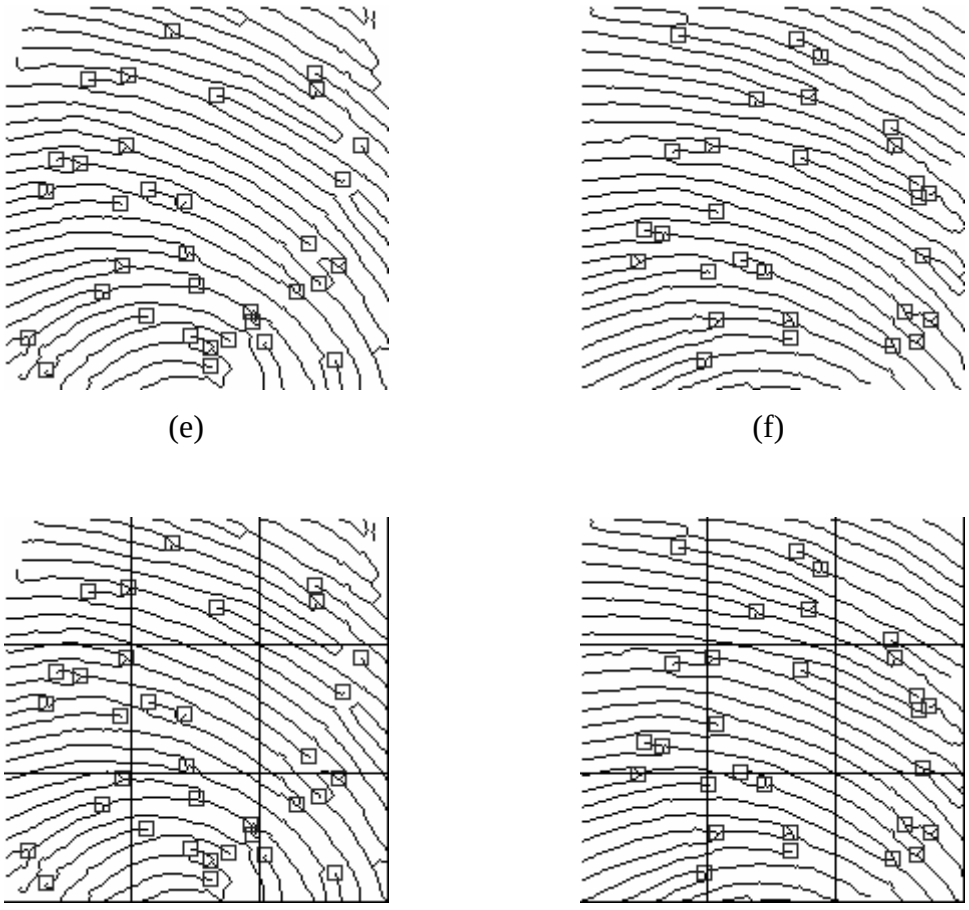
To solve this problem, the Block Template matching was proposed, which is a novel technique in fingerprint matching. This technique does not depend on the direction and location of the extracted minutiae. In Block Template matching, we used the minutiae distance from each other. With that ability, the algorithm is able to identify two identical fingerprint images even after rotation or translation.

(a)

(b)

(c)

(d)



(e) (f)

(g) (h)

Figure 4.28 A matched fingerprint image, which has different translation and rotation: (a) and (b) are from the same fingerprint image; (c) and (d) thinned fingerprint image; (e) and (f) extracted minutiae superimposed in thinned fingerprint image; and (g) and (h) extracted minutiae divided on 3x3 window.

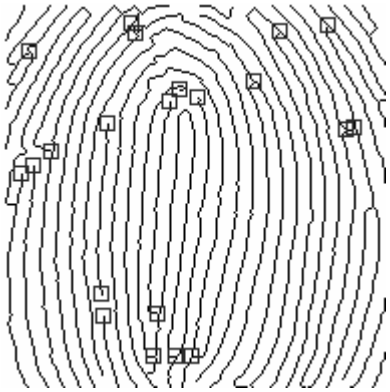
From Figure 4.28, the total block that matched the master template is 5 and the block matching index is 0.5556. Figure 4.29 shows the matching results of the four identical fingerprint images in different orientation.

(a)

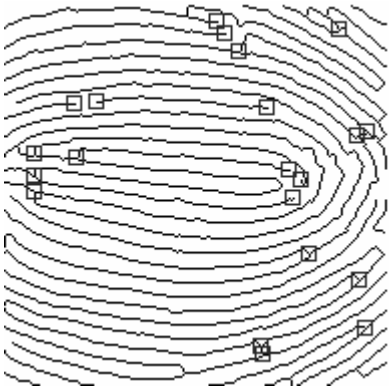
(b)

(c)

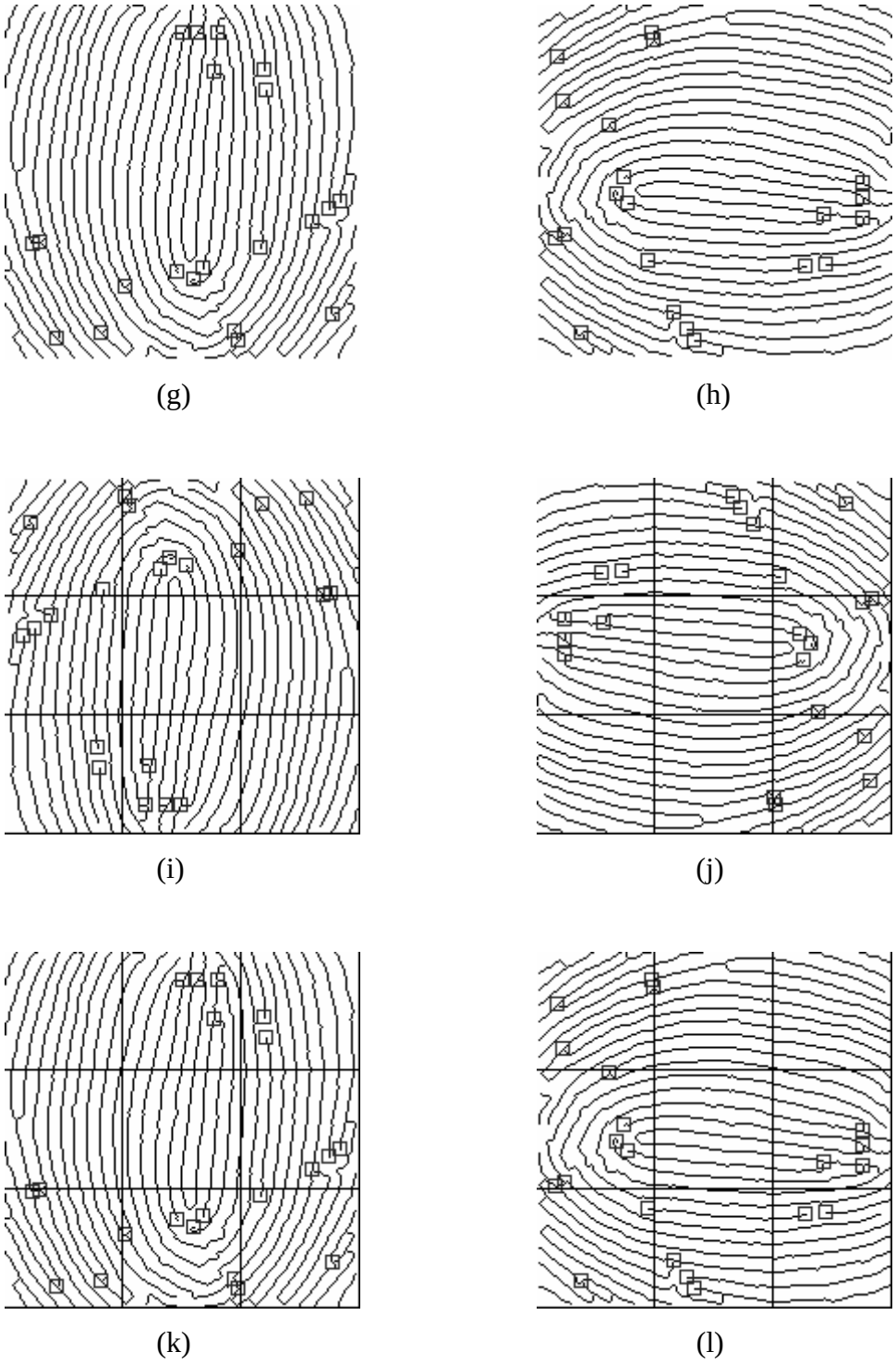
(d)



(e)



(f)



(g) (h)

(i) (j)

(k) (l)

Figure 4.29 Matching results of four fingerprint rotations: 0°; 90°; 180°; 270°; (a)-(d) thinned fingerprint image; (e)-(h)extracted minutiae superimposed on thinned image; and (i)-(l) extracted minutiae on 3x3 windows

The matching process was run on Intel Pentium II 1.0 GHz using flat file as database, and the speed is an average of 3 seconds for both techniques.

4.5 Discussion

Based on the results, we conclude that the proposed methodology makes a tremendous improvement in fingerprint identification and verification. Using this methodology, the minutiae extraction is better, with less number of false minutiae, which the key to successful identification and verification of fingerprint.

There are a few factors for successful fingerprint identification and verification. These factors are:

1. Using Histogram Equalization and High-Pass to improve the directional elements in directional image.
2. Reconstructing a new fingerprint image using Directional Fourier Filtering.
3. Using Ridge Line Following algorithm to produce thinned fingerprint image which is able to eliminate the spurious minutiae.
4. Using a new Block Template minutiae extraction to create master template.
5. Using a new Block Template in matching stage.

All these factors are already discussed in details in Chapter 3. There is also a weakness which leads to the failure or false fingerprint identification and verification. The factor is the quality of fingerprint image, which contains noise and is of low quality.

The quality of fingerprint image is very important to reduce false minutiae extraction. Generally, every fingerprint image has noise and is hard to eliminate. For example, the noise that is generated by human pores. These pores are on the ridges, and generate sweat to moist the fingerprint. With the sweat on fingerprint, people leave their fingerprint image everywhere. If the pores are visible during fingerprint acquisition, it may lead to the creation of false minutiae.

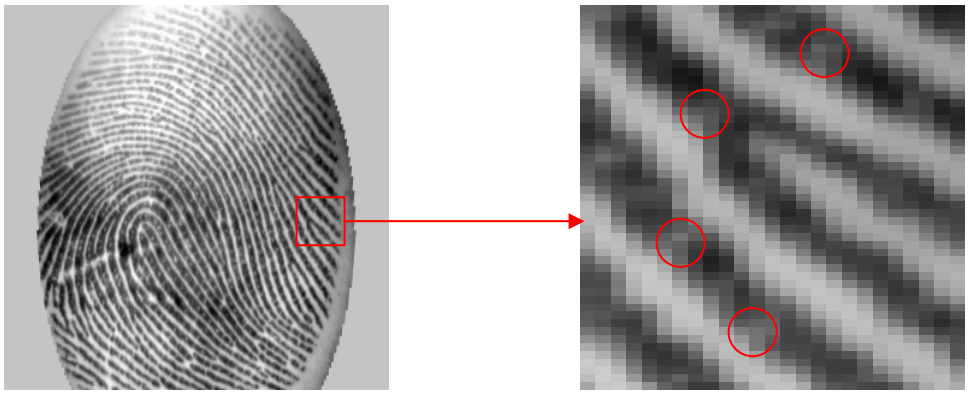


Figure 4.30 Fingerprint containing pores on ridges

The pores will make the directional image calculation to be more difficult. When this occurs, the directional image may contain wrong directional elements and lead to false minutiae extraction. Figure 4.31 shows two low quality fingerprint images.

(a) (b)

Figure 4.31 Low quality fingerprint images: (a) very light, (b) very dark

Noise or low quality fingerprint image exists during fingerprint acquisition. The light fingerprint image happens when user presses the thumb or finger softly, or the fingerprint may be very dry. The dark fingerprint image happens when user presses hardly on the fingerprint scanner, or the fingerprint may be very wet.

From all the results of the experiment in each stage, we determine the best methods to be used in the fingerprint identification system. These methods are summarized in Figure 4.32 below.

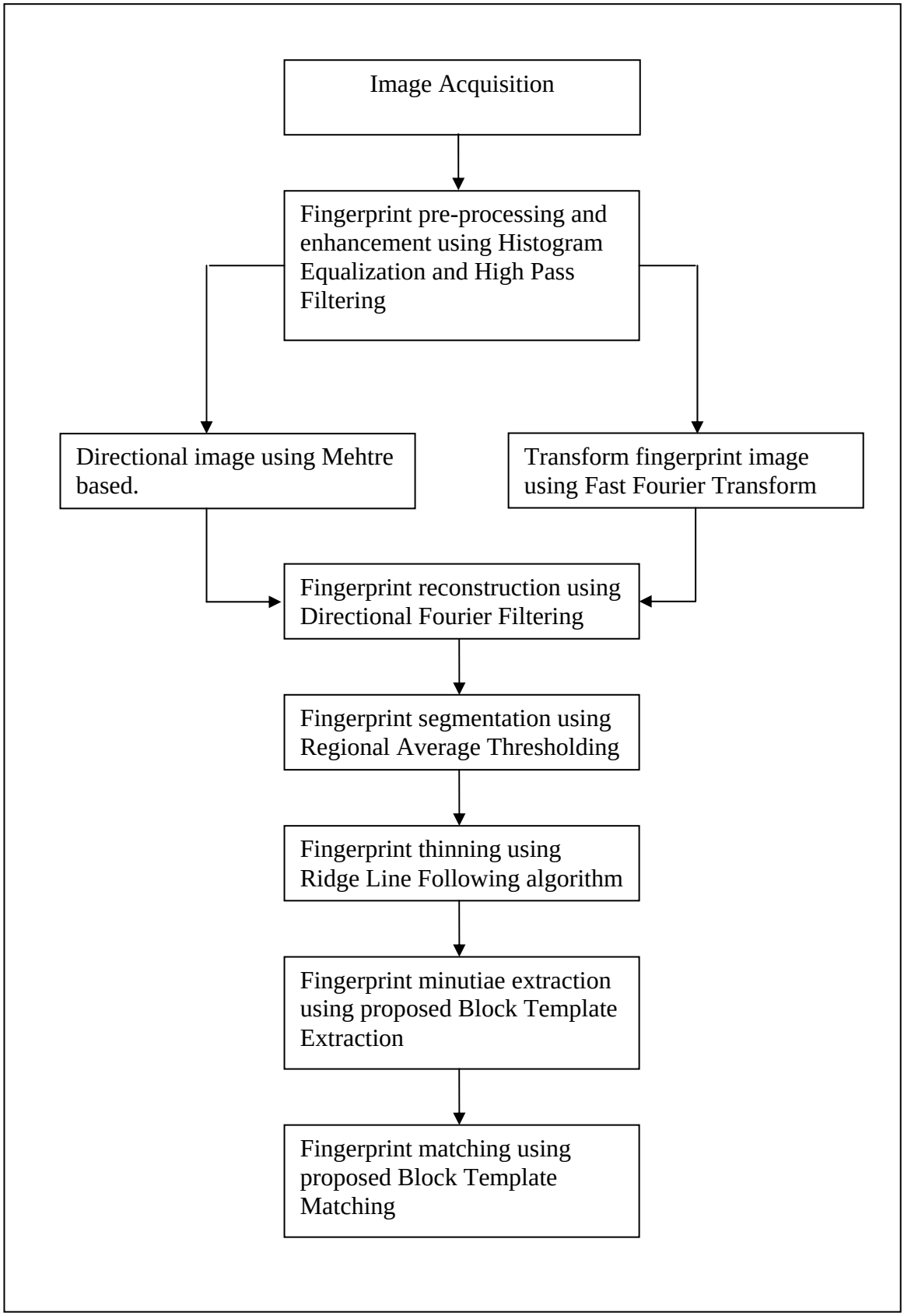


Figure 4.32 The proposed method in the fingerprint identification system

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 Overview

This chapter concludes all the results from the experiment and lists of contributions in this research. The objectives of the research cover various existing methodology or algorithm in each stage of the fingerprint identification system, which includes the pre-processing, enhancement, directional image, segmentation, thinning, feature extraction and matching.

5.2 Summary of the Study

Based on the results from all the experiments in this research, we found:

1. The usage of Histogram Equalization in pre-processing and enhancement stage has successfully improved the quality of fingerprint image. With the high quality image, the directional image is better and more accurate.

2. Filtering fingerprint image in frequency domain produces a new fingerprint image of higher quality.
3. Fingerprint reconstruction produces a new fingerprint image free from noise.
4. The Ridge Line Following was used to thin out the binary image obtained through the Regional Average Thresholding to one pixel. The skeleton image maintains the genuine features to reduce the percentage of false minutiae.
5. Using successful one-to-many technique in fingerprint matching.

As a conclusion, the genuine minutiae extraction percentage was increased up to 90% and thus decreases the false minutiae. With the high percentage of genuine minutiae, matching accuracy also increases by up to 90%.

5.3 Contributions of the Study

1. Introducing a new framework for fingerprint identification as shown in Figure 1.2. This new framework adds three more steps before the segmentation stage. Those three steps are transforming image into frequency domain, reconstructing new image and median filtering. In this framework, we manage to have new fingerprint image with high quality.
2. Enhancing the quality of directional image using Histogram Equalization and High Pass Filtering.
3. Introducing the fingerprint reconstruction using frequency domain approach to obtain high quality fingerprint image. Using this approach, the scars were successfully removed.
4. Comparative study between Mehtre techniques with Least Mean Square Orientation Estimation in getting the directional image.
5. Comparative study between Mehtre based and Least Mean Square Orientation Estimation as directional elements in producing a new fingerprint image.

6. Comparative study in fingerprint segmentation. Here, we examine Regional Average Thresholding, Global Thresholding, Histogram Based Thresholding, and Niblack Binarization in choosing the best algorithm for fingerprint segmentation.
7. Comparative study in fingerprint thinning. Here, we examine Fast Thinning Algorithm, Two-Way Pass Thinning Algorithm, and Ridge Line Following Thinning Algorithm in choosing the best algorithm for fingerprint thinning.
8. Comparative study between Crossing Number and Template in fingerprint features extraction.
9. Proposing a new enhanced feature extraction method based on Crossing Number to using block technique known as Proposed Block Template Extraction.
10. Proposing a new enhanced fingerprint matching based on Template Matching known as Proposed Block Template Extraction.

5.4 Future Work

Despite the fact that the fingerprint identification system can achieve good performance, we believe that there are more research works needed in making the system to be more effective in practice.

5.4.1 Fingerprint Matching

In order to determine whether a pair of fingerprint images is from the same finger, two conditions must be fulfilled: (i) the two fingerprints must be of the same

pattern configuration, and (ii) they must share a substantial number of identical minutiae details. Currently our minutiae algorithm depends only on the assessment of the second condition to make decision. Obviously, this is not enough. Fingerprint image contains noise that could introduce false minutiae, and leads to false identification. A fingerprint classification scheme assigns a fingerprint into one of the pre-specified categories based on its global pattern configuration. If two fingerprints are from the same finger, they must be in the same category.

REFERENCES

Amengual, J.C., Juan, A. Perez, J.C., Prat, F., Saez, S., and Vilar, J.M. (1997). Real Time Minutiae Extraction in Fingerprint Images. *Conference Publication*. 443. 871-875.

Bazen, A. M. dan Gerez, S. H. (2000). Computational Intelligence In Fingerprint Identification. *Proc. 2nd IEEE Benelux Signal Processing Symposium (SPS-2000).* S00-1 – S00-4.

Baruch, O. (1988). Line Thinning by Line Following. *Pattern Recognition Letters.* 8**.** 271-276.

Castleman, K.R. (1996). *Digital Image Processing*. Prentice-Hall, Englewood Cliffs, New Jersey.

Chapel, C. (1971). *Fingerprinting – A Manual of Identification*. Coward McCann, New York.

Candela, G.T., Grother, P.J., Watson, C.I., Wilkinson, R.A., and Wilson, C.L. (1995). *PCASYS – A Pattern-Level Classification Automation System for Fingerprints*. Technical Report. NISTIR 5647.

Cappelli, C., Lumini, A., Maio, D. and Maltoni, D. (1999). Fingerprint Classfication by Directional Image Partitioning. *IEEE Transaction on Pattern Analysis and Machine Intelligence*. 21(5): 402–421.

Cherill, F. R. (1954). *The Fingerprint System at Scotland Yard.* HMSO London

Chin R. dan Jang B. (1992). One-Pass Thinning: Analysis, Properties and Quantitative Evaluation. *IEEE Transaction on Pattern Analysis and Machine Intelligence.* 11. 14**.** 1129-1140.

Cummins, H., and Mildo, C. (1961). *Finger Prints, Palms and Soles*. Dover Publication Inc. New York.

DeLaRue Printrak Inc. (1985). *Automated Classification System Reader Project*

(ACS). Technical Report, February.

Emiroglu, I. (1997). *Fingerprint Image Enhancement and Recognition*. Department of Electrical and Electronic Engineering, University of Hertfordshire, Thesis Ph.D.

Emiroglu, I. (1997). *Fingerprint Image Enhancement and Recognition*. Department of Electrical and Electronic Engineering, University of Hertfordshire, Thesis Ph.D.

Federal Bureau of Investigation. (1984). *The Science of Fingerprints: Classification and Uses*. U.S. Government Printing Office, Washington, D. C.

Galton, F. (1961). *Finger Prints*. Da Capo Press, New York.

Gonzalez, R.C. and Woods, R.E. (2002). *Digital Image Processing. 2nd Edition*. Upper Saddle River: Prentice Hall.

Gonzalez, R.C. and Woods, R.E. (1992). *Digital Image Processing*. Addison-Wesley Publishing Company.

Mehtre, B. M. (1993). Fingerprint Image Analysis for Automatic Identification. *Machine Vision and Applications.* vol. 6. No. 2-3, pp. 124-139.

Mehtre, B. M., Murthy, N. N., Kapoor, S. (1987). Segmentation of Fingerprint Images using the Directional image. *Pattern Recognition.* vol. 20. No 4. pp. 429-435.

Moenssens, A. (1971). *Fingerprint techniques*. Chilton Book Company. London.

Mohamad Kharulli Othman. (2002). Fingerprint Enhancement And Reconsturction Using Inverse Fast Fourier Transform With Filter and Non Filter Image. *Seminar Kumpulan Fokus-2002*, Universiti Teknologi Malaysia, 2002.

Mohamad Kharulli Othman. (2002). Digital Image Processing: Biometric Systems and Fingerprint Application. *KUTPM Journal of Technology & Management (2002) 25-26.* Kolej Universiti Teknologi dan Pengurusan Malaysia.

Mohamad Kharulli Othman. (2003). Fingerprint Pre-Processing and Enhancement: Scars Removal. *International Arab Conference on Information Technology (ACIT'2003).* Iskandariah, Egypt.

Maio, D. dan Maltoni, D. (1997). Direct Gray-Scale Minutiae Detection In Fingerprints. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* 19. 1. 27-40.

Mehtre, B.M. dan Chatterjee, B. (1989). Segmentation of Fingerprint Images – A

Composite Method. *Pattern Recognition*. 22.4. 381-385.

Mehtre B., Chartterje B., Kapoor S., Murthy N. (1987). Segmentation of fingerprints images using directional images, *Pattern Recognition*, 20**.** 429-435.

Newham, E. (1995). *The Biometric Report*. SJB Services, New York

Henry, E.R. (1905). *Classification and Uses of Finger Prints.* Wyman and Sons Ltd

Hong, L. Wan, Y. dan Jain, A. (1997a). Fingerprints Image Enhancement : Algorithm and Performance Evaluation. *Pattern Recognition and Image Processing Laboratory*, Department of Computer Science.

Hong. L, (1998). *Automatic Personal Identification Using Fingerprints*. Ph.D Dissertation, Michigan State University, June 25.

Ikonomopoulos, A., Unser, M. (1984). A Directional Filtering Approach to Texture Discrimination. *Proceedings of the Seventh International Conference on Pattern Recognition.* Montreal. Canada. 30 July-2August. pp. 87-89.

Ikonomopolus, A., Kunt, M. (1985). High Compression Image Coding via Directional Filtering. *Signal Processing 8 North-Holland.* pp. 179-203.

Ikonomopolus, A., Kunt, M. (1985). High Compression Image Coding via Directional Filtering. Signal Processing 8 North-Holland. pp. 179-203.

Jain, A., Hong, L. dan Blooe, R. (1997a). On-Line Fingerprint Verification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 19. 4**.** 302-314.

Jain, A.K., Lin, Hong, Pankanti, S., and Bolle, R. (1997). An Identity-Authentication System Using Fingerprints. *Proceedings of the IEEE*. 85(9): 1365–1388.

Jain, A.K., Lin, Hong and Bolle, R. (1997). On-line Fingerprint Verification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 19(4): 302–314.

Karu, K. and Jain, A.K. (1996). Fingerprint Classification. *Pattern Recognition*. 29(3): 389–404.

Kasaei, S., Deriche, M., dan Boashash, B. (1997). Fingerprint Feature Enhancement Using Block-Direction On Reconstructed Images. *International Conference on Information, Communications and Signal Processing.* 721-725.

Kunt, M., Ikonomopoulos, A., Kocher, M. (1985). Second-Generation Image Coding Techniques. *Proceedings of IEEE* vol. 73, No 4, pp. 549-574, April 1985.

Kass, M. and Wtkin, A. (1987). Analyzing oriented patterns. *Computer Vision Graphics Image Process.* pp. 362-385.

Kawagoe, M. and Tojo, A. (1984). Fingerprint pattern classification. *Pattern*

Recognition. pp. 295-303.

Lee, H. C. and Gainsslen, R. E. (1991). *Advances in Fingerprint Technology*. Elsevier, New York.

Leong, Chung Ern. (2003). *Fingerprint Classification: A Bi-Resolution Approach to Singular Point Extraction.* Universiti Teknologi Malaysia: Master Thesis.

Lin, Hong, Jain, A.K., Pankanti, S., and Bolle, R. (1996). Fingerprints Enhancement. *Proceedings 3rd IEEE Workshop on Applications of Computer Visions*. 202–207.

Lin, Hong and Jain, A.K. (1998). *Classification of Fingerprint Images*. Technical Report. MSUCPS:TR98-18.

Lin, Hong, Wan, Y., Jain, A.K. (1997). Fingerprints Image Enhancement: Algorithm and Performance Evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 20(8): 777–789.

Parker, James R. (1997*). Algorithms for Image Processing and Computer Vision*. John Wiley and Sons, New York.

Ratha, N.K., Chen, Shaoyun dan Jain, A.K. (1995). Adaptive Flow Orientation-Based Feature Extraction In Fingerprint Images. *Pattern Recognition.* 28. 11. 1657-1672.

Stock, R.M. and Swonger, C.W. (1969). Development and evaluation of a reader of fingerprint minutiae. *Cornell Aeronautical Laboratory. Technical Report* CAL No. XM-2478-X-1:13-17.

Sherlock, B.G., Monro, D.M. dan Millard, K. (1994). Fingerprint Enhancement by Directional Fourier Filtering. *IEEE Proc – Visual Image Signal Processing*. 141.2**.** 87-94.

Siti Masrina Sulong (2000). *Pengesanan Minutiae Dalam Imej Cap Jari Berskala Kelabu*. Universiti Teknologi Malaysia: Master Thesis.

Stock, R.M. and Swonger, C.W. (1969). Development and evaluation of a reader of fingerprint minutiae. *Cornell Aeronautical Laboratory. Technical Report* CAL No. XM-2478-X-1:13-17.

Xiao, Sun dan Zhuming, Ai (1996). Automatic Feature Extraction and Recognition of Fingerprint Images. *Proceedings of ICSP '96*. 1086-1089.

Xiao, Qinghan dan Raafat, Hazem (1991). Fingerprint Image Postprocessing: A Combined Statistical and Structural Approach. *Pattern Recognition.* **24. 10.** 985-992.

Young, I.T., Gerbrands, J.J. dan van Vliet, L.J. (1995). Fundamentals of Image Processing. Delft University of Technology.

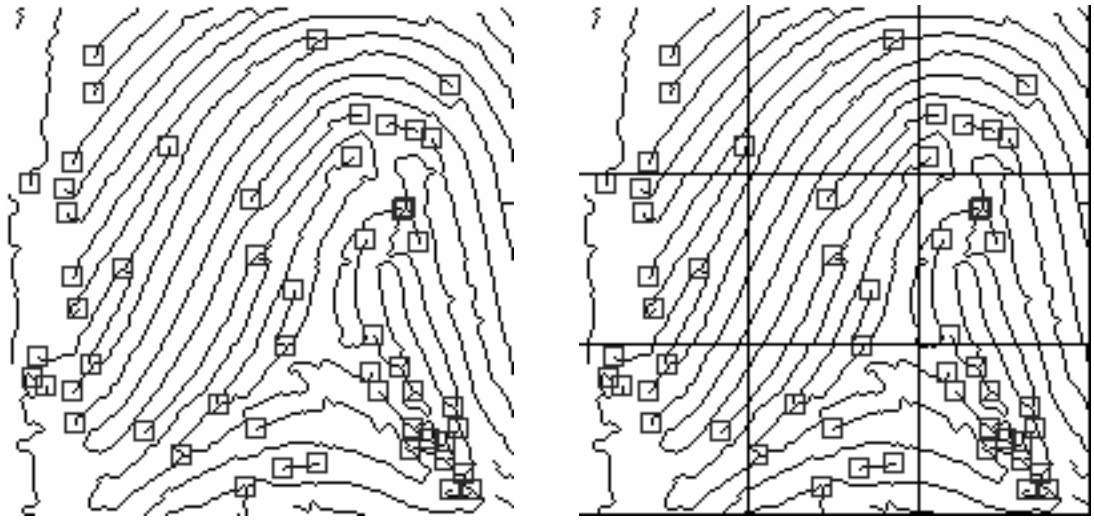
APPENDIX A

Results from Arch Class

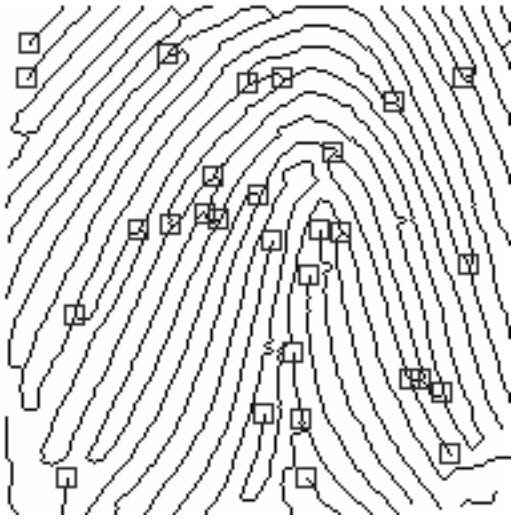
Results from fingerprint identification and verification in arch class are presented here. Here, we list ten fingerprint images structured according to the captions in the boxes below.

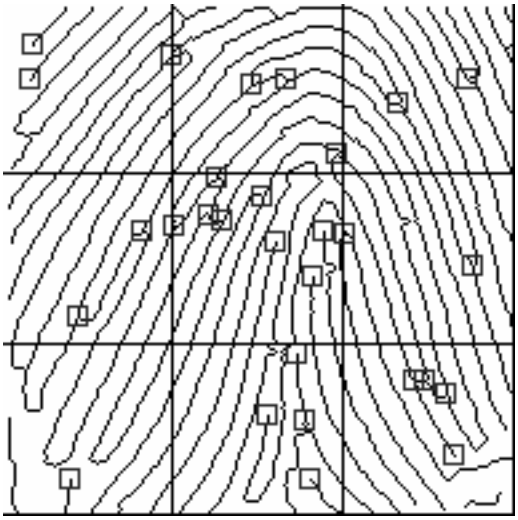
Original fingerprint image	Fingerprint image from pre-processing and enhancement
Fingerprint directional image	New reconstructed fingerprint image
Fingerprint image after segmentation	Fingerprint image after thinning
Minutiae extraction	Block minutiae template extraction

Fingerprint 1

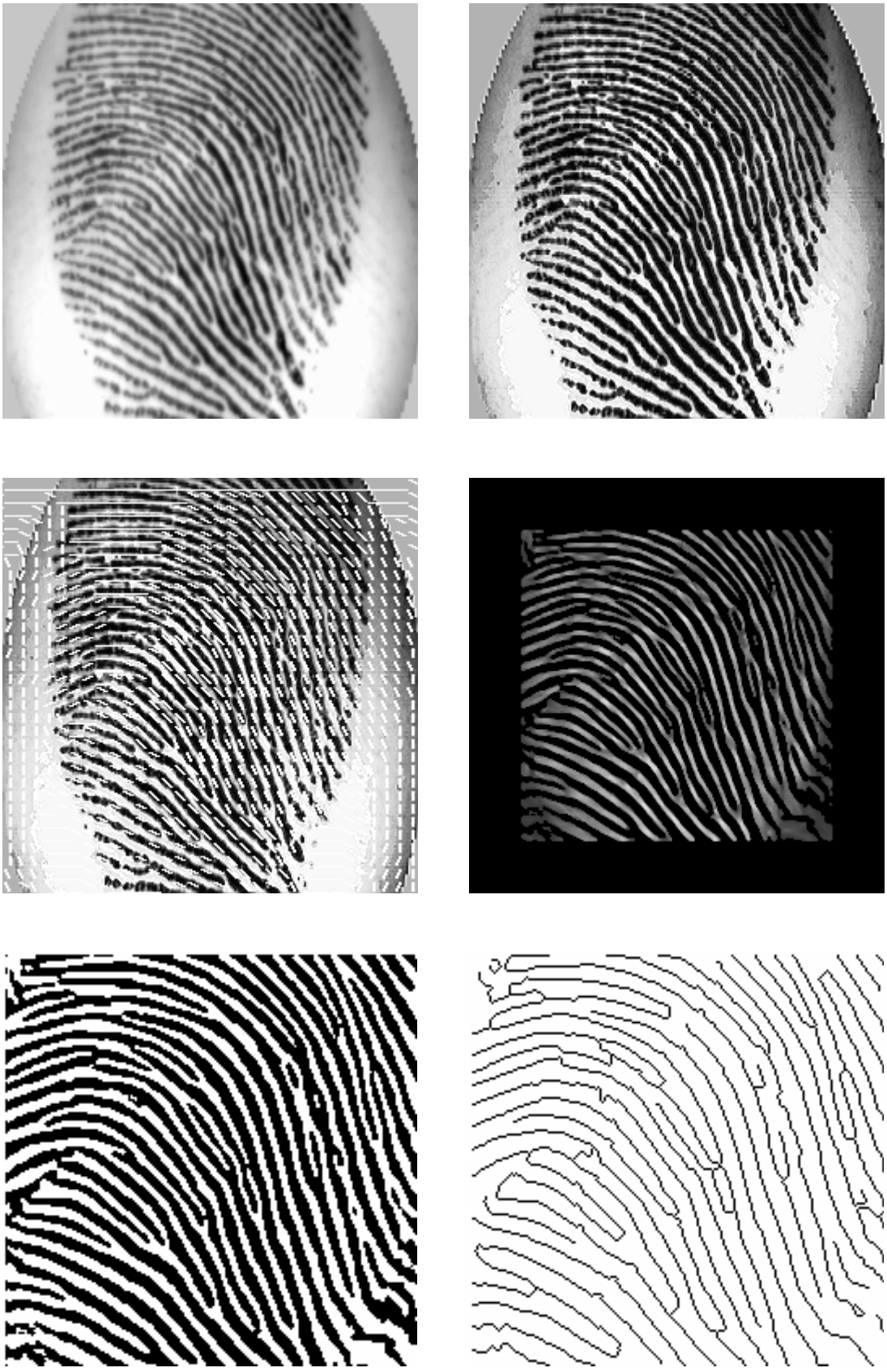


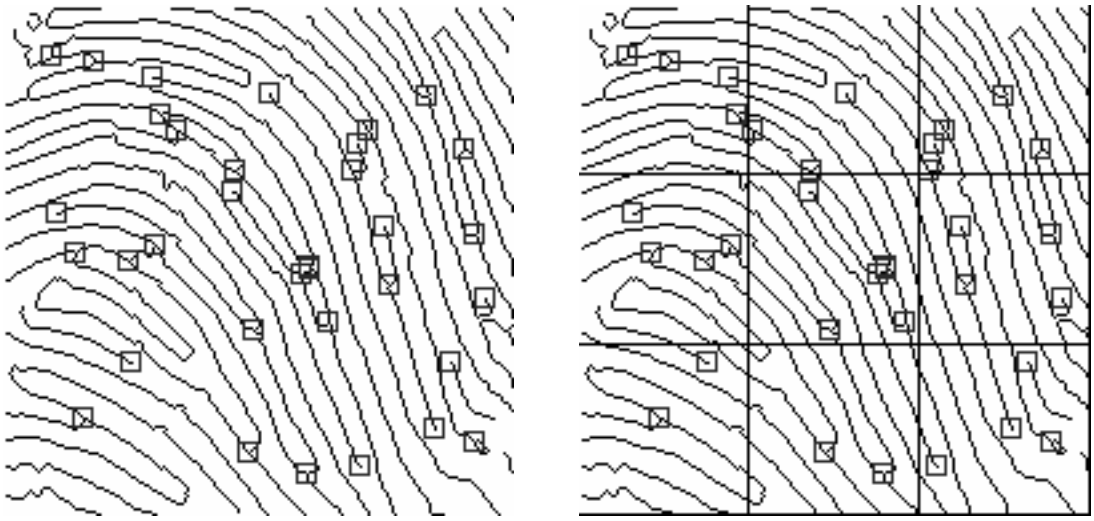
Fingerprint 2



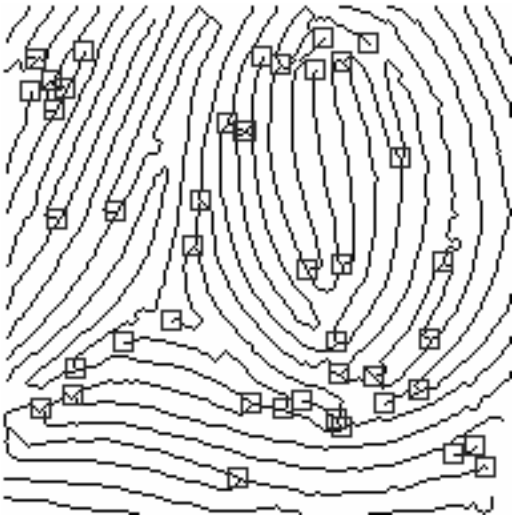


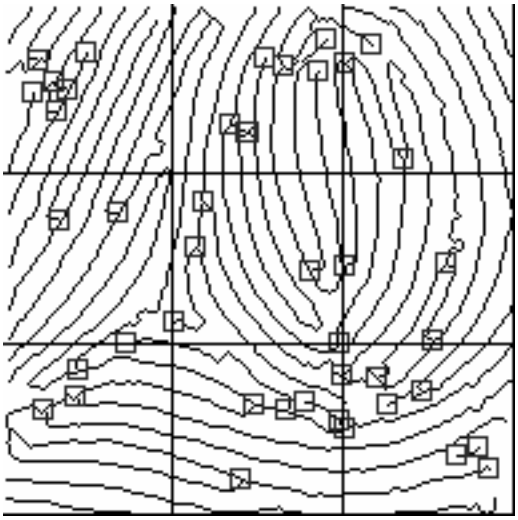
Fingerprint 3



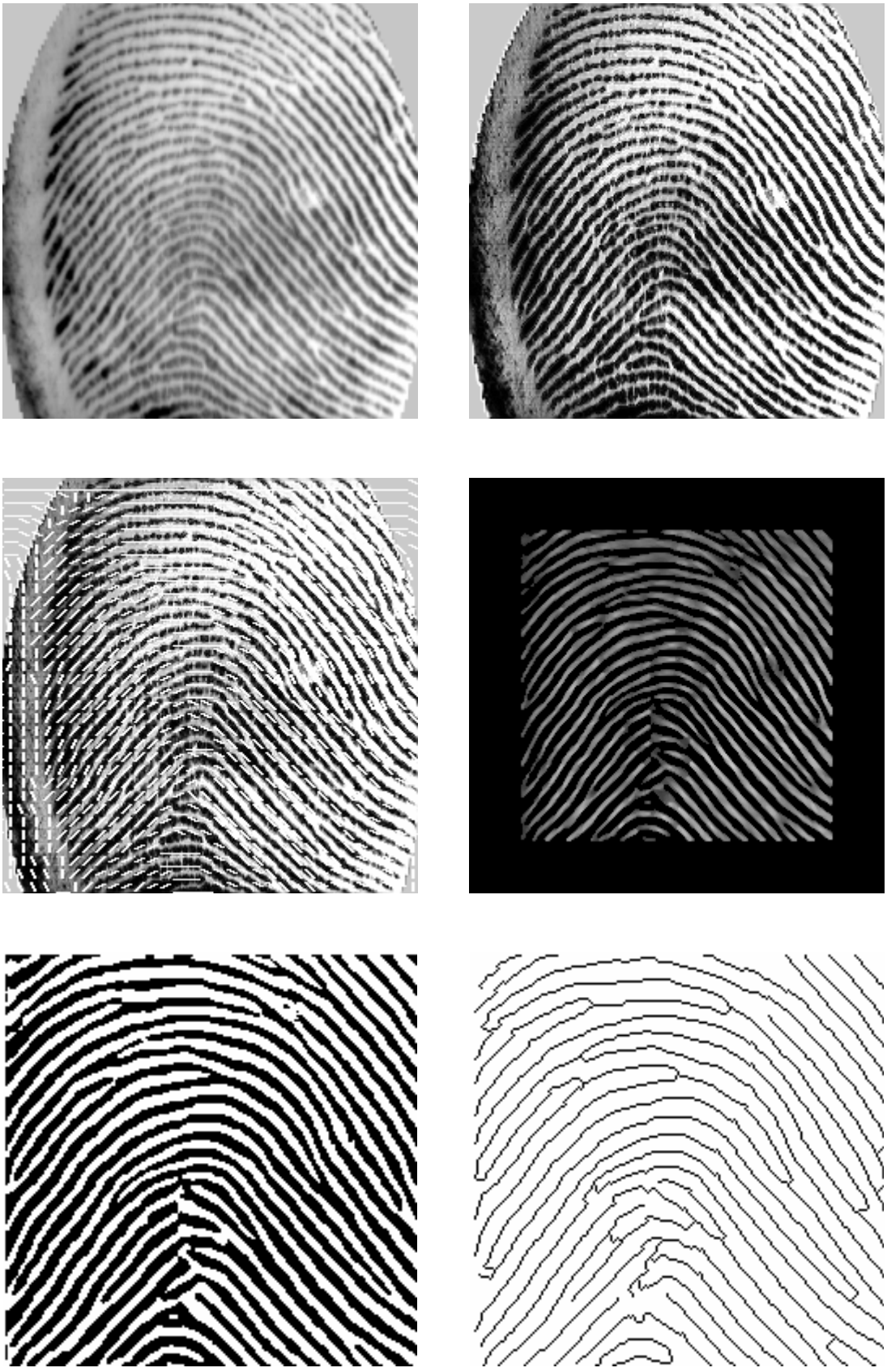


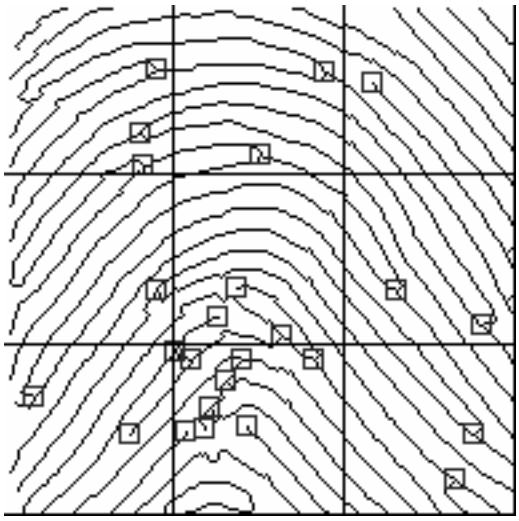
Fingerprint 4





Fingerprint 5





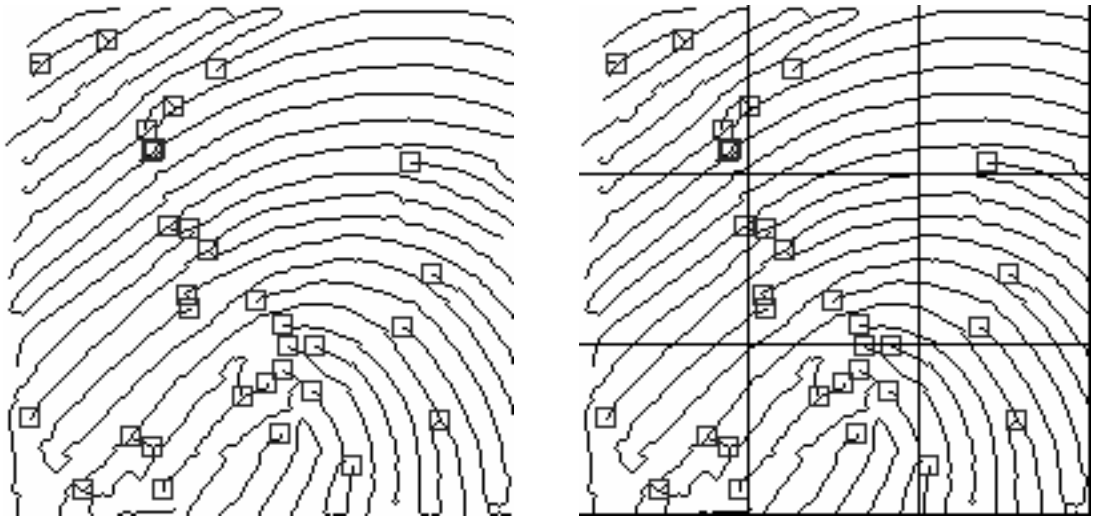
APPENDIX B

Results from Left Loop Class

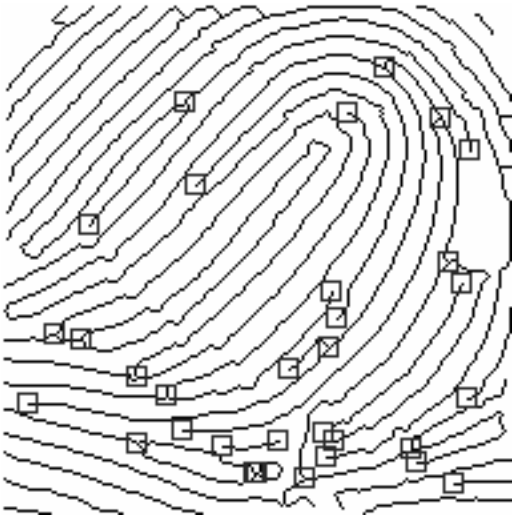
Result from fingerprint identification and verification in left loop class is presented here. Here, we list ten fingerprint images structured according to the captions in the boxes below.

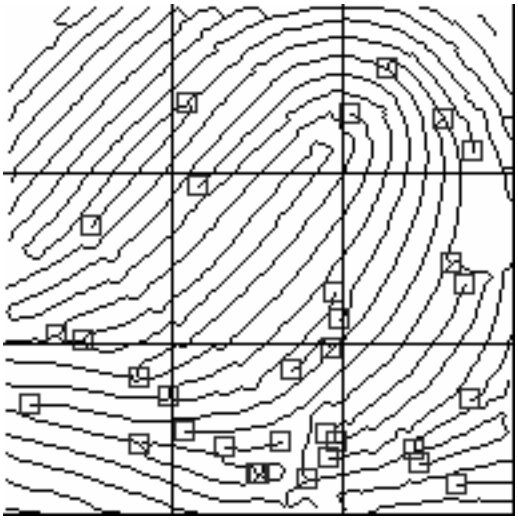
Original fingerprint image	Fingerprint image from pre-processing and enhancement
Fingerprint directional image	New reconstructed fingerprint image
Fingerprint image after segmentation	Fingerprint image after thinning
Minutiae extraction	Block minutiae template extraction

Fingerprint 1

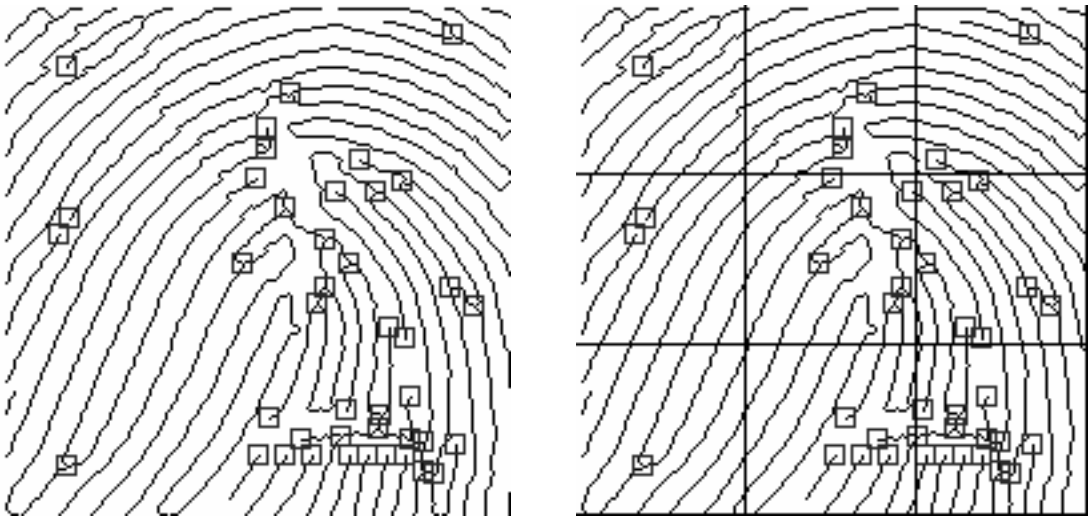


Fingerprint 2

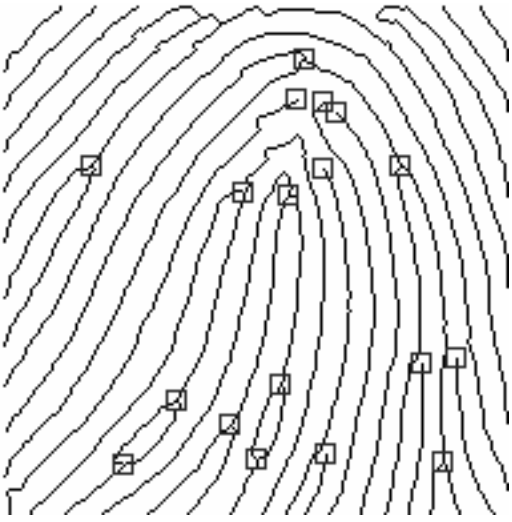


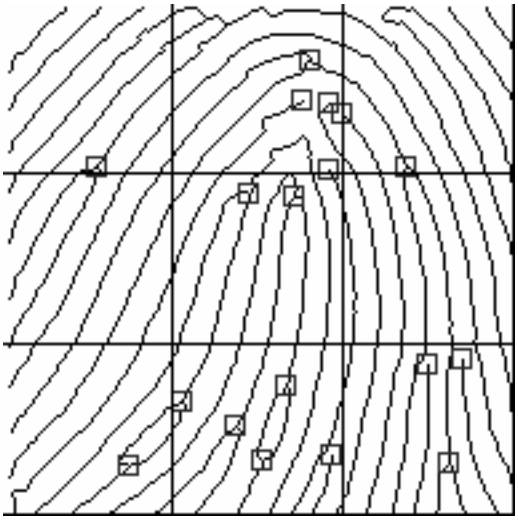


Fingerprint 3

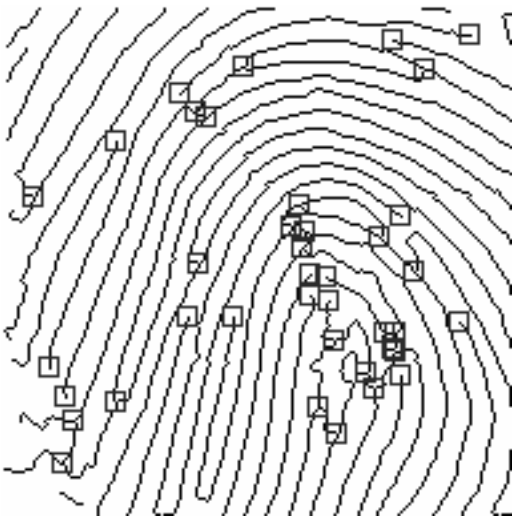


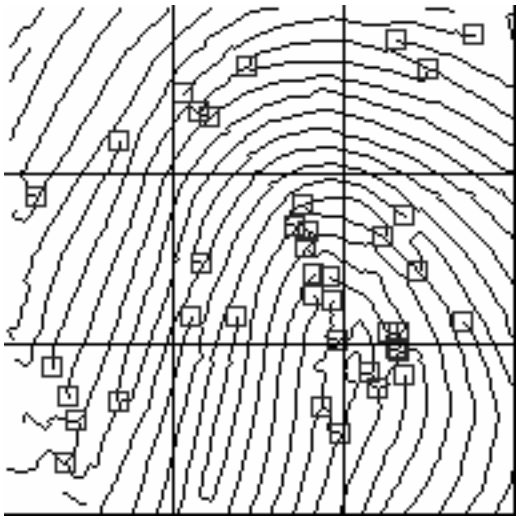
Fingerprint 4





Fingerprint 5





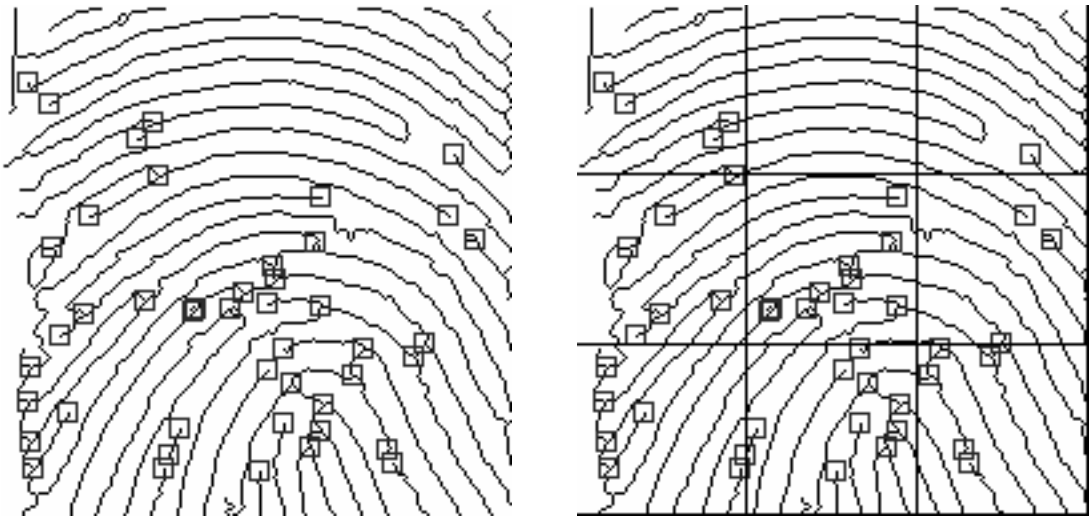
APPENDIX C

Results from Right Loop Class

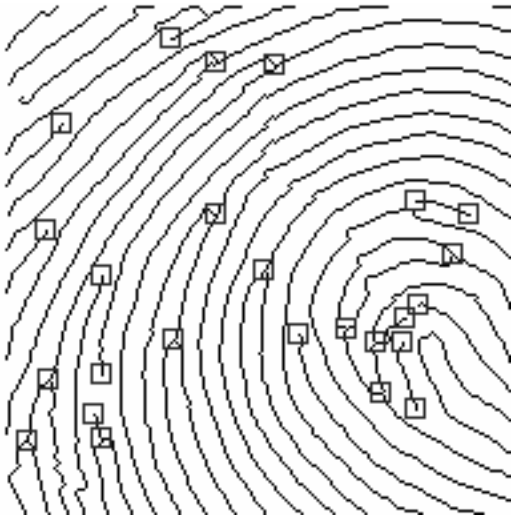
Results from fingerprint identification and verification in right loop class are presented here. The figures are structured like below. Here we list ten fingerprint images

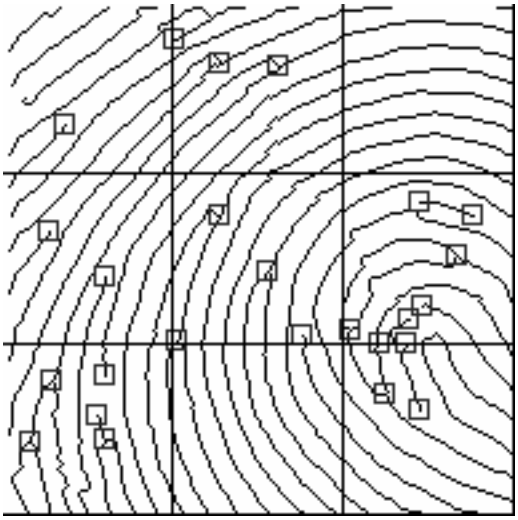
Original fingerprint image	Fingerprint image from pre-processing and enhancement
Fingerprint directional image	New reconstructed fingerprint image
Fingerprint image after segmentation	Fingerprint image after thinning
Minutiae extraction	Block minutiae template extraction

Fingerprint 1

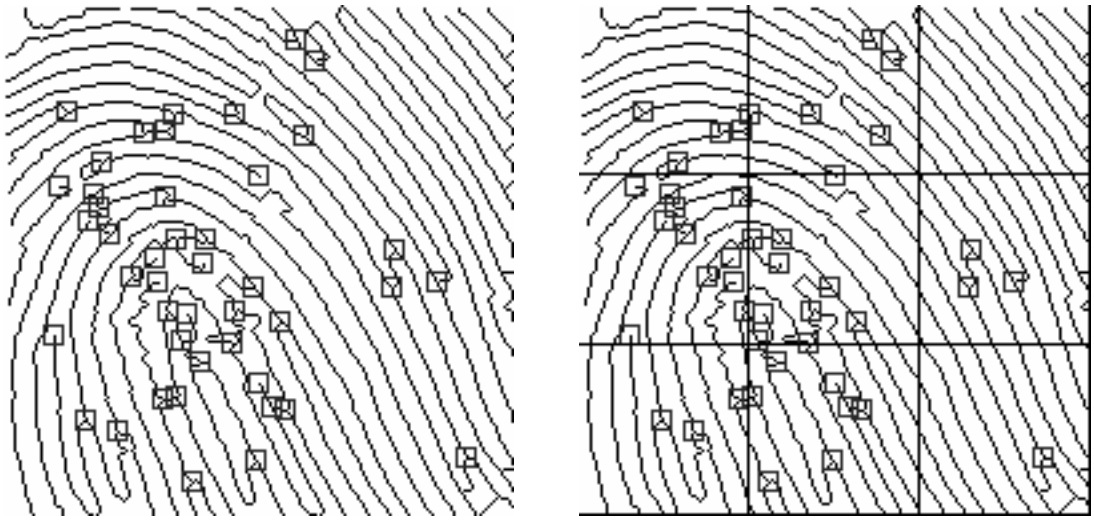


Fingerprint 2

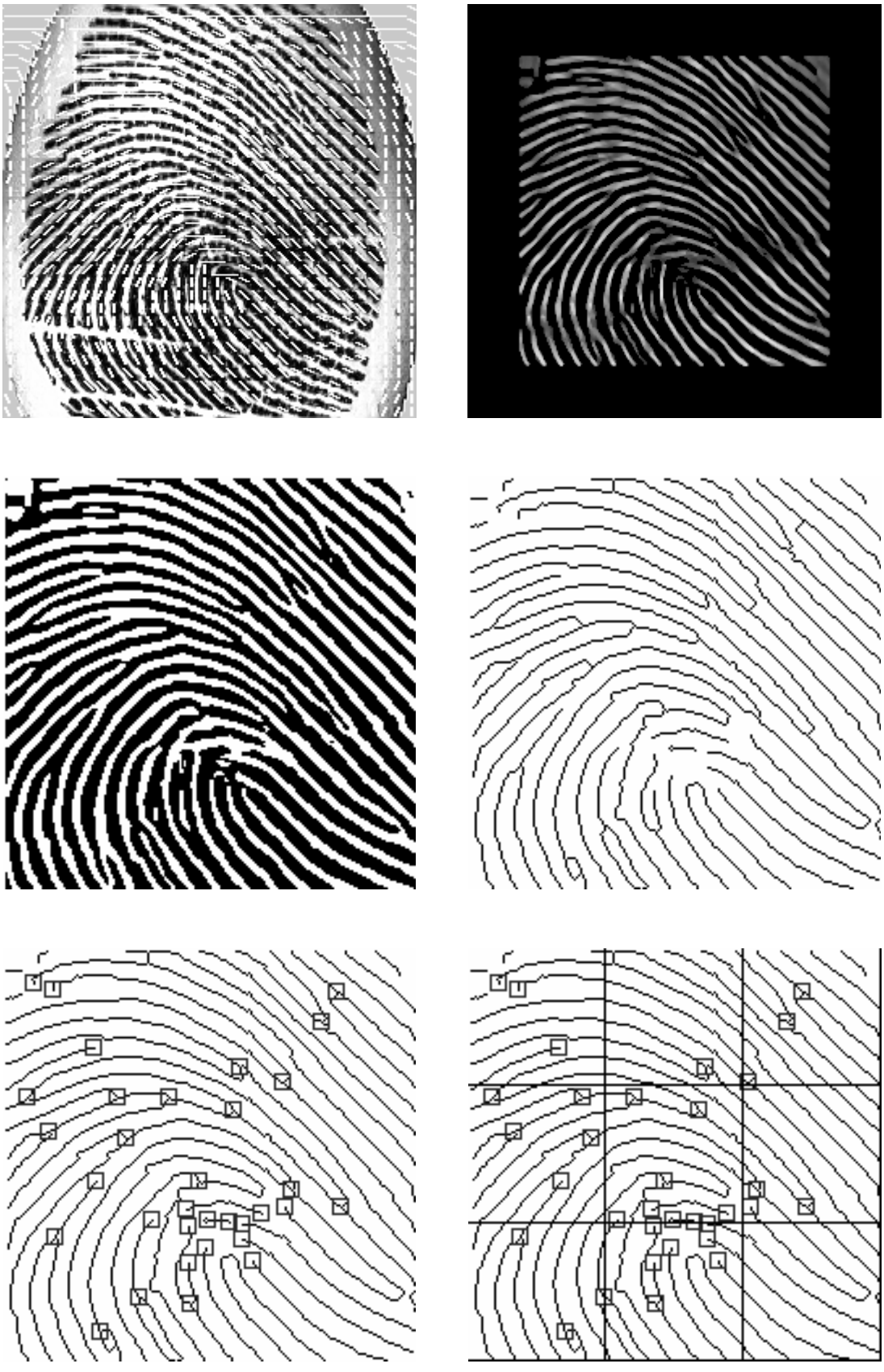




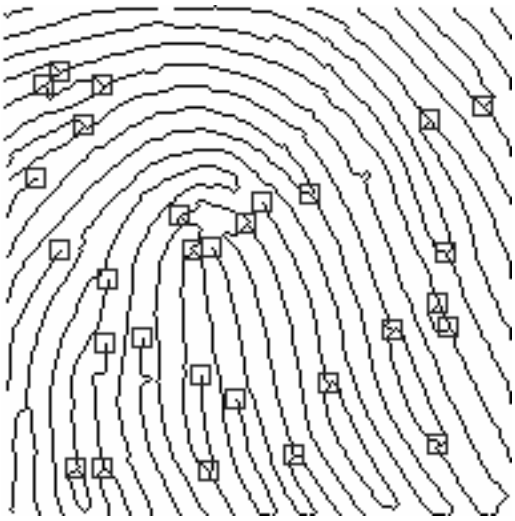
Fingerprint 3

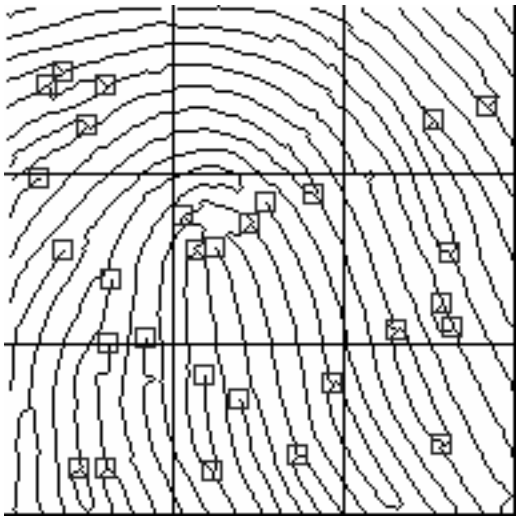


Fingerprint 4



Fingerprint 5





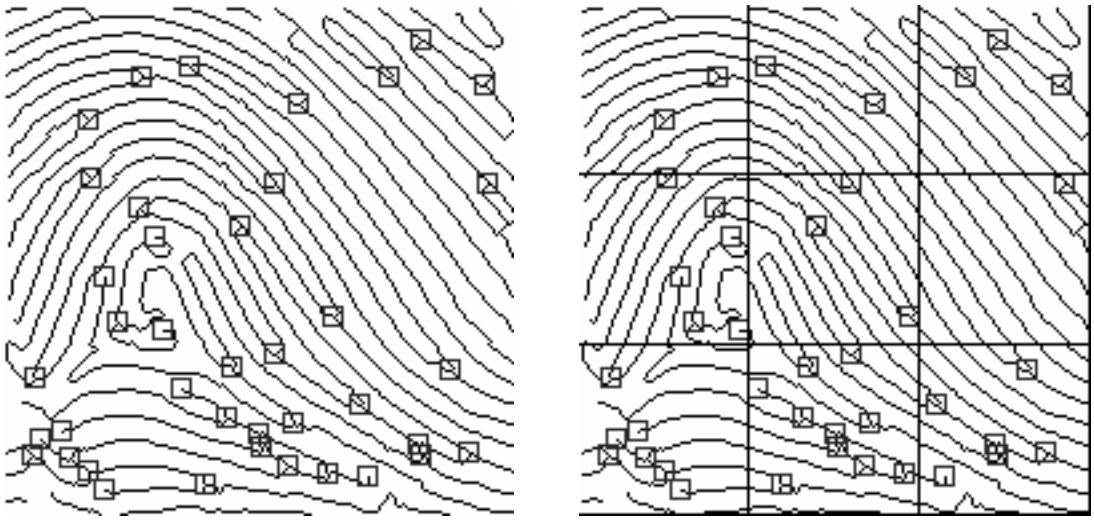
APPENDIX D

Results from Tented Class

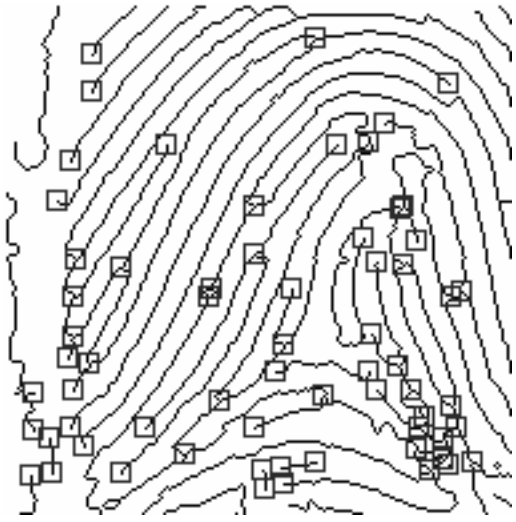
Result from fingerprint identification and verification in tented class is presented here. Here, we list ten fingerprint images structured according to the captions in the boxes below.

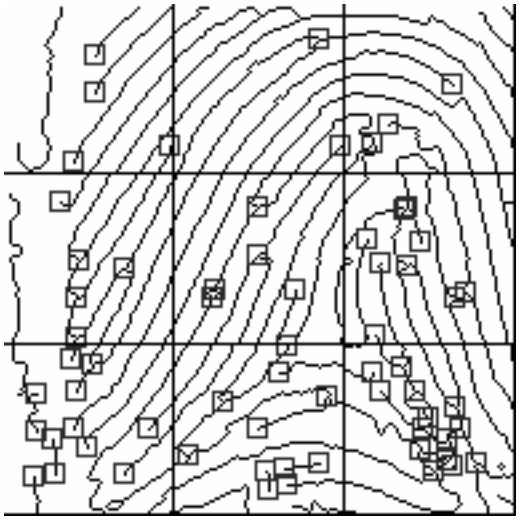
Original fingerprint image	Fingerprint image from pre-processing and enhancement
Fingerprint directional image	New reconstructed fingerprint image
Fingerprint image after segmentation	Fingerprint image after thinning
Minutiae extraction	Block minutiae template extraction

Fingerprint 1

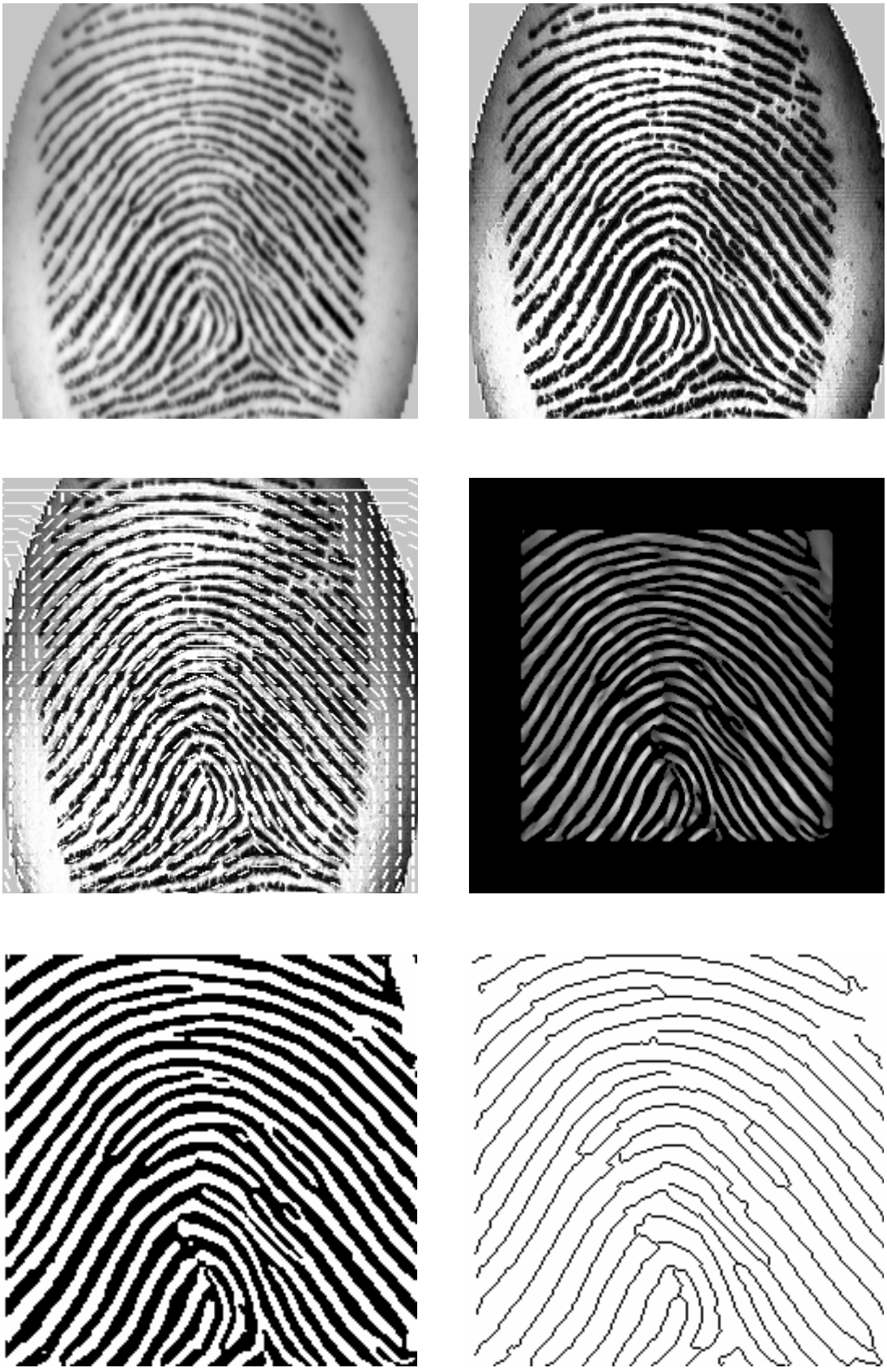


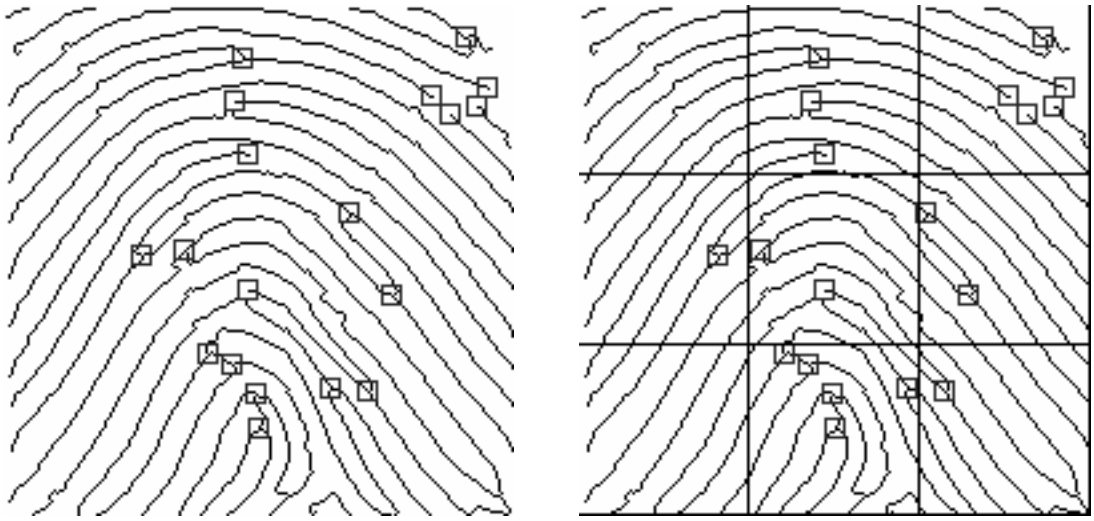
Fingerprint 2



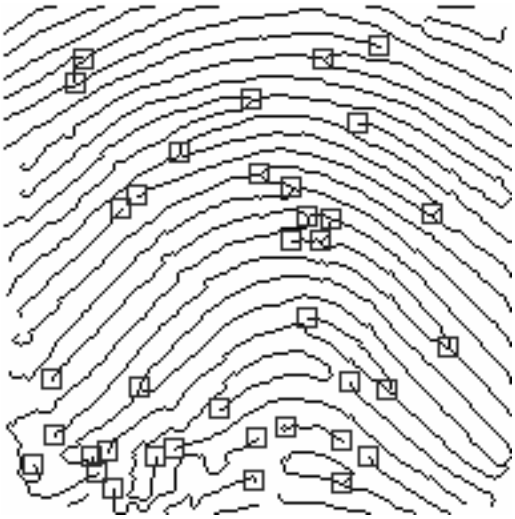


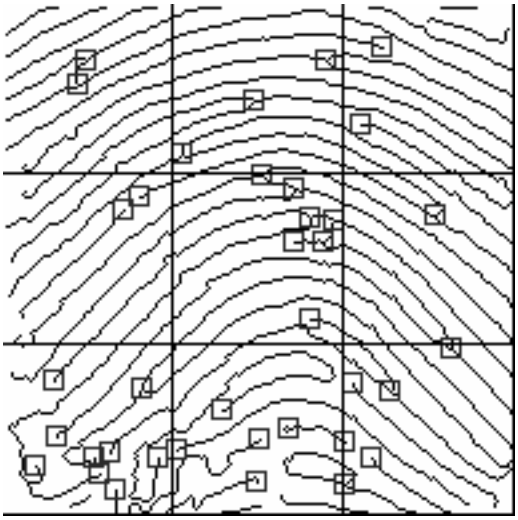
Fingerprint 3



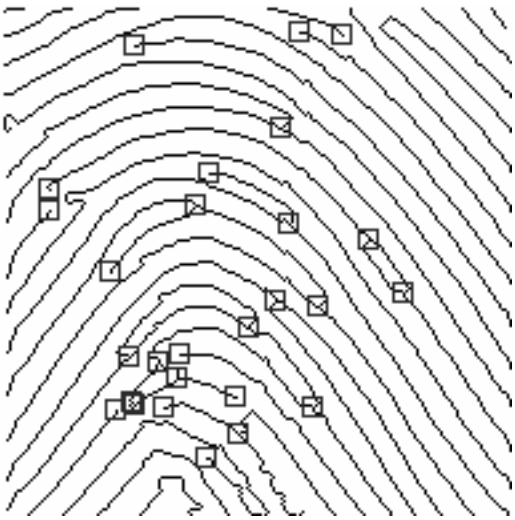


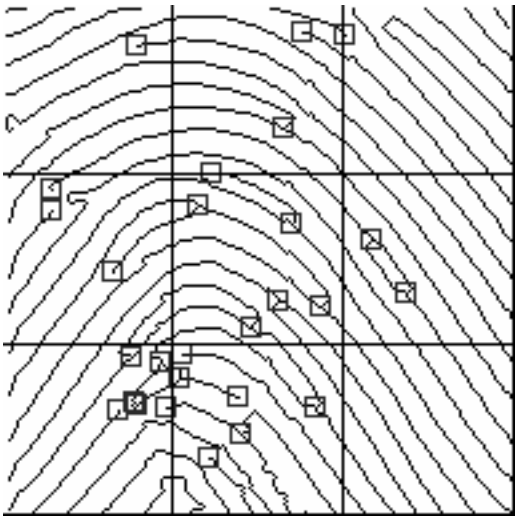
Fingerprint 4





Fingerprint 5





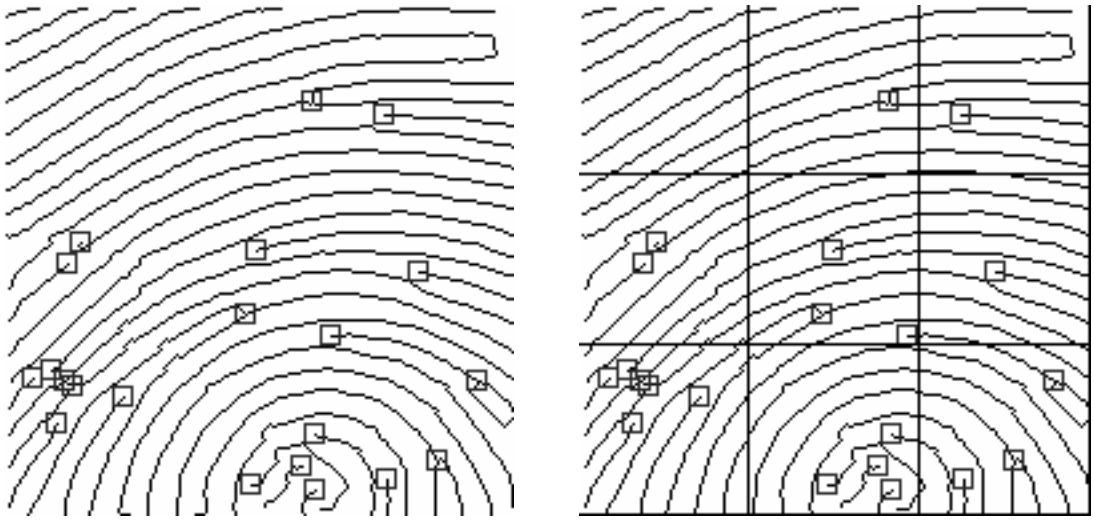
APPENDIX E

Results from Whorl Class

Result from fingerprint identification and verification in whorl class is presented here. Here, we list ten fingerprint images structured according to the captions in the boxes below.

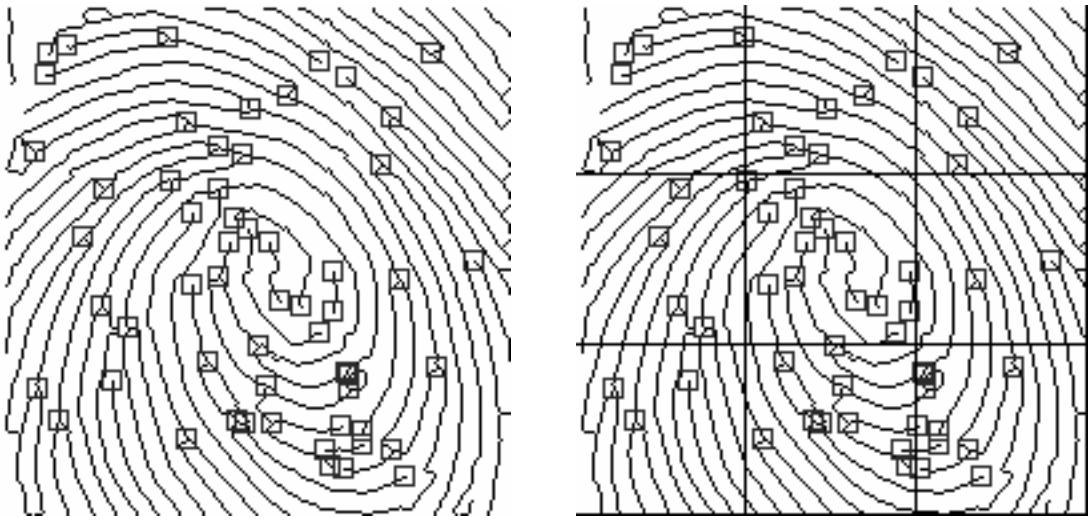
Original fingerprint image	Fingerprint image from pre-processing and enhancement
Fingerprint directional image	New reconstructed fingerprint image
Fingerprint image after segmentation	Fingerprint image after thinning
Minutiae extraction	Block minutiae template extraction

Fingerprint 1

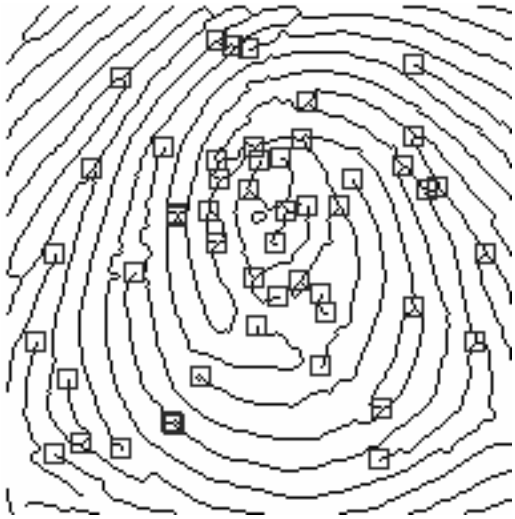


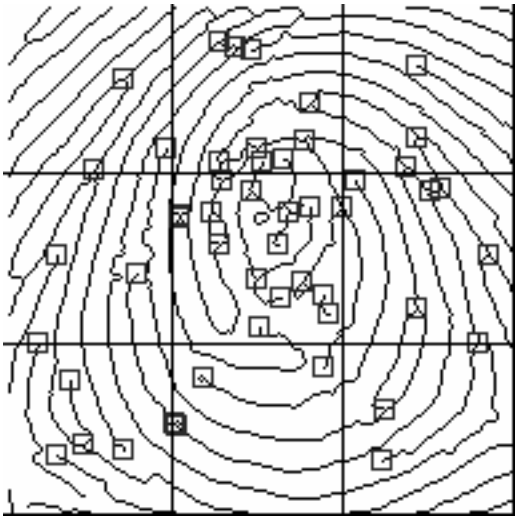
Fingerprint 2

Fingerprint 3



Fingerprint 4





Fingerprint 5