Fingerprint Image Enhancement and Minutiae Extraction

Raymond Thai

This report is submitted as partial fulfilment
of the requirements for the Honours Programme of the
School of Computer Science and Software Engineering,
The University of Western Australia,
2003

Abstract

Fingerprints are the oldest and most widely used form of biometric identification. Despite the widespread use of fingerprints, there is little statistical theory on the uniqueness of fingerprint minutiae. A critical step in studying the statistics of fingerprint minutiae is to reliably extract minutiae from the fingerprint images. However, fingerprint images are rarely of perfect quality. They may be degraded and corrupted due to variations in skin and impression conditions. Thus, image enhancement techniques are employed prior to minutiae extraction to obtain a more reliable estimation of minutiae locations.

In this dissertation, I firstly provide discussion on the methodology and implementation of techniques for fingerprint image enhancement and minutiae extraction. Experiments using a mixture of both synthetic test images and real fingerprint images are then conducted to evaluate the performance of the implemented techniques. In combination with these techniques, preliminary results on the statistics of fingerprint images are then presented and discussed.

Keywords: fingerprint, image enhancement, filtering, minutiae extraction, image postprocessing, fingerprint statistics.
CR Categories: I.4.3 Pattern Recognition - Computer Vision.

Acknowledgements

I would most of all like to thank my supervisor Dr Peter Kovesi for the invaluable advice and positive encouragement he has provided me throughout the course of this project. Special thanks go to Mohan Obeysekera for his help in proofreading this dissertation. Thanks also go to all my fellow Honours and BE(Software) students for making the Honours year an enjoyable one. Finally, to my girlfriend Cat for her continuous love and support, and for always being there for me.

Contents

List of Tables

List of Figures

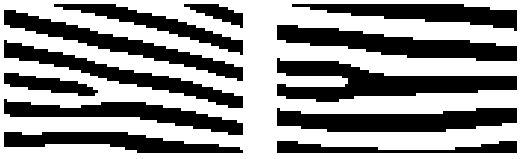
CHAPTER 1

Introduction

Fingerprints have been used for over a century and are the most widely used form of biometric identification. Fingerprint identification is commonly employed in forensic science to support criminal investigations, and in biometric systems such as civilian and commercial identification devices. Despite this widespread use of fingerprints, there has been little statistical work done on the uniqueness of fingerprint minutiae. In particular, the issue of how many minutiae points should be used for matching a fingerprint is unresolved.

The fingerprint of an individual is unique and remains unchanged over a lifetime. A fingerprint is formed from an impression of the pattern of ridges on a finger. A ridge is defined as a single curved segment, and a valley is the region between two adjacent ridges. The minutiae, which are the local discontinuities in the ridge flow pattern, provide the features that are used for identification. Details such as the type, orientation, and location of minutiae are taken into account when performing minutiae extraction [9].

Galton [5] defined a set of features for fingerprint identification, which since then, has been refined to include additional types of fingerprint features. However, most of these features are not commonly used in fingerprint identification systems. Instead the set of minutiae types are restricted into only two types, ridge endings and bifurcations, as other types of minutiae can be expressed in terms of these two feature types. Ridge endings are the points where the ridge curve terminates, and bifurcations are where a ridge splits from a single path to two paths at a Y-junction. Figure 1.1 illustrates an example of a ridge ending and a bifurcation. In this example, the black pixels correspond to the ridges, and the white pixels correspond to the valleys.



(a) Ridge ending (b) Bifurcation

Figure 1.1: Example of a ridge ending and a bifurcation.

Fingerprint images are rarely of perfect quality. They may be degraded and corrupted with elements of noise due to many factors including variations in skin and impression conditions. This degradation can result in a significant number of spurious minutiae being created and genuine minutiae being ignored. A critical step in studying the statistics of fingerprint minutiae is to reliably extract minutiae from fingerprint images. Thus, it is necessary to employ image enhancement techniques prior to minutiae extraction to obtain a more reliable estimate of minutiae locations.

The primary aim of this project is to implement a series of techniques for fingerprint image enhancement and minutiae extraction. Experiments using both synthetic test images and real fingerprint images are used to assess the performance of the implemented techniques. These techniques are then used to extract minutiae from a sample set of fingerprint images. By using the extracted minutiae data, preliminary experiments on the statistics of fingerprints can then be conducted.

This dissertation is organised into three main topics, with each chapter focusing on a different topic. Each chapter builds on the work discussed in earlier chapters. Chapter 2 describes the methodology and implementation of a series of techniques for fingerprint image enhancement. Chapter 3 provides discussion on minutiae extraction and fingerprint image postprocessing. Both Chapters 2 and 3 are structured in the same way in that each chapter contains a literature review, methodology, and an experimental results section. Chapter 4 presents the methodology and results of conducting statistical experiments on fingerprint images. Chapter 5 provides concluding comments and also lists further extensions that can be made to this project.

CHAPTER 2

Fingerprint Image Enhancement

The quality of the ridge structures in a fingerprint image is an important characteristic, as the ridges carry the information of characteristic features required for minutiae extraction. Ideally, in a well-defined fingerprint image, the ridges and valleys should alternate and flow in locally constant direction. This regularity facilitates the detection of ridges and consequently, allows minutiae to be precisely extracted from the thinned ridges. However, in practice, a fingerprint image may not always be well defined due to elements of noise that corrupt the clarity of the ridge structures. This corruption may occur due to variations in skin and impression conditions such as scars, humidity, dirt, and non-uniform contact with the fingerprint capture device [9]. Thus, image enhancement techniques are often employed to reduce the noise and enhance the definition of ridges against valleys.

This chapter provides discussion on the methodology and implementation of a fingerprint image enhancement algorithm. First, I will review existing techniques in the field of fingerprint image enhancement. This will be followed by a description of the methods used to implement the enhancement algorithm. The results of the experiments involving each stage of the fingerprint enhancement algorithm will then be presented and discussed.

2.1 Literature review

One of the most widely cited fingerprint enhancement techniques is the method employed by Hong et al. [8], which is based on the convolution of the image with Gabor filters tuned to the local ridge orientation and ridge frequency. The main stages of this algorithm include normalisation, ridge orientation estimation, ridge frequency estimation and filtering.

The first step in this approach involves the normalisation of the fingerprint image so that it has a prespecified mean and variance. Due to imperfections in the fingerprint image capture process such as non-uniform ink intensity or

non-uniform contact with the fingerprint capture device, a fingerprint image may exhibit distorted levels of variation in grey-level values along the ridges and valleys. Thus, normalisation is used to reduce the effect of these variations, which facilitates the subsequent image enhancement steps.

An orientation image is then calculated, which is a matrix of direction vectors representing the ridge orientation at each location in the image. The widely employed gradient-based approach is used to calculate the gradient [18, 20, 22], which makes use of the fact that the orientation vector is orthogonal to the gradient. Firstly, the image is partitioned into square blocks and the gradient is calculated for every pixel, in the x and y directions. The orientation vector for each block can then be derived by performing an averaging operation on all the vectors orthogonal to the gradient pixels in the block. Due to the presence of noise and corrupted elements in the image, the ridge orientation may not always be correctly determined. Given that the ridge orientation varies slowly in a local neighbourhood, the orientation image is then smoothed using a low-pass filter to reduce the effect of outliers.

The next step in the image enhancement process is the estimation of the ridge frequency image. The frequency image defines the local frequency of the ridges contained in the fingerprint. Firstly, the image is divided into square blocks and an oriented window is calculated for each block. For each block, an x-signature signal is constructed using the ridges and valleys in the oriented window. The x-signature is the projection of all the grey level values in the oriented window along a direction orthogonal to the ridge orientation. Consequently, the projection forms a sinusoidal-shape wave in which the centre of a ridge maps itself as a local minimum in the projected wave. The distance between consecutive peaks in the x-signature can then be used to estimate the frequency of the ridges.

Fingerprint enhancement methods based on the Gabor filter have been widely used to facilitate various fingerprint applications such as fingerprint matching [17, 19] and fingerprint classification [12]. Gabor filters are bandpass filters that have both frequency-selective and orientation-selective properties [4], which means the filters can be effectively tuned to specific frequency and orientation values. One useful characteristic of fingerprints is that they are known to have well defined local ridge orientation and ridge frequency. Therefore, the enhancement algorithm takes advantage of this regularity of spatial structure by applying Gabor filters that are tuned to match the local ridge orientation and frequency.

Based on the local orientation and ridge frequency around each pixel, the Gabor filter is applied to each pixel location in the image. The effect is that the filter enhances the ridges oriented in the direction of the local orientation, and decreases anything oriented differently. Hence, the filter increases the contrast

between the foreground ridges and the background, whilst effectively reducing noise.

An alternative approach to enhancing the features in a fingerprint image is the technique employed by Sherlock [21] called directional Fourier filtering. The previous approach was a spatial domain technique that involves spatial convolution of the image with filters, which can be computationally expensive. Alternatively, operating in the frequency domain allows one to efficiently convolve the fingerprint image with filters of full image size.

The image enhancement process begins by firstly computing the orientation image. In contrast to the previous method, which estimates the ridge orientation using a continuous range of directions, this method uses a set of only 16 directions to calculate the orientation. An image window is centred at a point in the raw image, which is used to obtain a projection of the local ridge information. The image window is then rotated in each of the 16 equally spaced directions, and in each direction a projection along the window's y axis is formed. The projection with the maximum variance is used as the dominant orientation for that point in the image. This process is then repeated for each pixel to form the orientation image.

Similar to the filtering stage applied by Hong et al.: after the orientation image has been computed, the raw image is then filtered using a set of bandpass filters tuned to match the ridge orientation. The image is firstly converted from the spatial domain into the frequency domain by application of the two-dimensional discrete Fourier transform. The Fourier image is then filtered using a set of 16 Butterworth filters with each filter tuned to a particular orientation. The number of directional filters corresponds to the set of directions used to calculate the orientation image. After each directional filter has been independently applied to the Fourier image, the inverse Fourier transform is used to convert each image back to the spatial domain, thereby producing a set of directionally filtered images called prefiltered images.

The next step in the enhancement process is to construct the final filtered image using the pixel values from the prefiltered images. This requires the value of the ridge orientation at each pixel in the raw image and the filtering direction of each prefiltered image. Each point in the final image is then computed by selecting, from the prefiltered images the pixel value whose filtering direction most closely matches the actual ridge orientation. The output of the filtering stage is an enhanced version of the image that has been smoothed in the direction of the ridges.

Lastly, local adaptive thresholding is applied to the directionally filtered image, which produces the final enhanced binary image. This involves calculating

the average of the grey-level values within an image window at each pixel, and if the average is greater than the threshold, then the pixel value is set to a binary value of one; otherwise, it is set to zero. The grey-level image is converted to a binary image, as there are only two levels of interest, the foreground ridges and the background valleys.

Overall, it can be seen that most techniques for fingerprint image enhancement are based on filters that are tuned according to the local characteristics of fingerprint images. Both of the examined techniques employ the ridge orientation information for tuning of the filter. However, only the approach by Hong et al. takes into account the ridge frequency information, as Sherlock's approach assumes the ridge frequency to be constant. By using both the orientation and ridge frequency information, it allows for accurate tuning of the Gabor filter parameters, which consequently leads to better enhancement results. Hence, I have chosen to employ the Gabor filtering approach by Hong et al. to perform fingerprint image enhancement.

2.2 Methodology

This section describes the methods for constructing a series of image enhancement techniques for fingerprint images. The algorithm I have implemented is built on the techniques developed by Hong et al. This algorithm consists of four main stages:

- normalisation,
- orientation estimation,
- ridge frequency estimation, and
- Gabor filtering.

In addition to these four stages, I have implemented three additional stages that include:

- segmentation,
- binarisation, and
- thinning.

In this section, I will discuss the methodology for each stage of the enhancement algorithm, including any modifications that have been made to the original techniques.

2.2.1 Segmentation

The first step of the fingerprint enhancement algorithm is image segmentation. Segmentation is the process of separating the foreground regions in the image from the background regions. The foreground regions correspond to the clear fingerprint area containing the ridges and valleys, which is the area of interest. The background corresponds to the regions outside the borders of the fingerprint area, which do not contain any valid fingerprint information. When minutiae extraction algorithms are applied to the background regions of an image, it results in the extraction of noisy and false minutiae. Thus, segmentation is employed to discard these background regions, which facilitates the reliable extraction of minutiae.

In a fingerprint image, the background regions generally exhibit a very low grey-scale variance value, whereas the foreground regions have a very high variance. Hence, a method based on variance thresholding [16] can be used to perform the segmentation. Firstly, the image is divided into blocks and the grey-scale variance is calculated for each block in the image. If the variance is less than the global threshold, then the block is assigned to be a background region; otherwise, it is assigned to be part of the foreground. The grey-level variance for a block of size $W \times W$ is defined as:

$$V(k) = \frac{1}{W^2} \sum_{i=0}^{W-1} \sum_{j=0}^{W-1} (I(i,j) - M(k))^2 \tag{2.1}$$

where $V(k)$ is the variance for block k, $I(i,j)$ is the grey-level value at pixel (i,j), and $M(k)$ is the mean grey-level value for the block k.

2.2.2 Normalisation

The next step in the fingerprint enhancement process is image normalisation. Normalisation is used to standardise the intensity values in an image by adjusting the range of grey-level values so that it lies within a desired range of values. Let $I(i,j)$ represent the grey-level value at pixel (i,j), and $N(i,j)$ represent the normalised grey-level value at pixel (i,j). The normalised image is defined as:

$$N(i,j) = \begin{cases} M_0 + \sqrt{\frac{V_0(I(i,j)-M)^2}{V}} & \text{if } I(i,j) > M, \\ M_0 - \sqrt{\frac{V_0(I(i,j)-M)^2}{V}} & \text{otherwise,} \end{cases} \tag{2.2}$$

where M and V are the estimated mean and variance of $I(i,j)$, respectively,

and M_0 and V_0 are the desired mean and variance values, respectively. Normalisation does not change the ridge structures in a fingerprint; it is performed to standardise the dynamic levels of variation in grey-level values, which facilitates the processing of subsequent image enhancement stages.

2.2.3 Orientation estimation

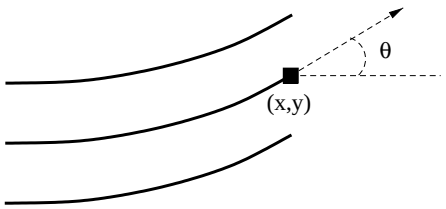


Figure 2.1: The orientation of a ridge pixel in a fingerprint.

The orientation field of a fingerprint image defines the local orientation of the ridges contained in the fingerprint (see Figure 2.1). The orientation estimation is a fundamental step in the enhancement process as the subsequent Gabor filtering stage relies on the local orientation in order to effectively enhance the fingerprint image. The least mean square estimation method employed by Hong et al. is used to compute the orientation image. However, instead of estimating the orientation block-wise, I have chosen to extend their method into a pixel-wise scheme, which produces a finer and more accurate estimation of the orientation field. The steps for calculating the orientation at pixel (i, j) are as follows:

1. Firstly, a block of size $W \times W$ is centred at pixel (i, j) in the normalised fingerprint image.

2. For each pixel in the block, compute the gradients $\partial_x(i, j)$ and $\partial_y(i, j)$, which are the gradient magnitudes in the x and y directions, respectively. The horizontal Sobel operator is used to compute $\partial_x(i, j)$:

$$\begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix} \tag{2.3}$$

   The vertical Sobel operator is used to compute $\partial_y(i, j)$:

$$\begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix} \tag{2.4}$$

3. The local orientation at pixel (i, j) can then be estimated using the following equations:

$$V_x(i,j) = \sum_{u=i-\frac{W}{2}}^{i+\frac{W}{2}} \sum_{v=j-\frac{W}{2}}^{j+\frac{W}{2}} 2\partial_x(u,v)\partial_y(u,v), \tag{2.5}$$

$$V_y(i,j) = \sum_{u=i-\frac{W}{2}}^{i+\frac{W}{2}} \sum_{v=j-\frac{W}{2}}^{j+\frac{W}{2}} \partial_x^2(u,v)\partial_y^2(u,v), \tag{2.6}$$

$$\theta(i,j) = \frac{1}{2} tan^{-1} \frac{V_y(i,j)}{V_x(i,j)}, \tag{2.7}$$

where $\theta(i, j)$ is the least square estimate of the local orientation at the block centred at pixel (i, j).

4. Smooth the orientation field in a local neighbourhood using a Gaussian filter. The orientation image is firstly converted into a continuous vector field, which is defined as:

$$\Phi_x(i,j) = cos(2\theta(i,j)), \tag{2.8}$$

$$\Phi_y(i,j) = sin(2\theta(i,j)), \tag{2.9}$$

where Φ_x and Φ_y are the x and y components of the vector field, respectively. After the vector field has been computed, Gaussian smoothing is then performed as follows:

$$\Phi'_x(i,j) = \sum_{u=-\frac{w_\Phi}{2}}^{\frac{w_\Phi}{2}} \sum_{v=-\frac{w_\Phi}{2}}^{\frac{w_\Phi}{2}} G(u,v)\Phi_x(i-uw, j-vw), \tag{2.10}$$

$$\Phi'_y(i,j) = \sum_{u=-\frac{w_\Phi}{2}}^{\frac{w_\Phi}{2}} \sum_{v=-\frac{w_\Phi}{2}}^{\frac{w_\Phi}{2}} G(u,v)\Phi_y(i-uw, j-vw), \tag{2.11}$$

where G is a Gaussian low-pass filter of size $w_\Phi \times w_\Phi$.

5. The final smoothed orientation field O at pixel (i, j) is defined as:

$$O(i,j) = \frac{1}{2} tan^{-1} \frac{\Phi'_y(i,j)}{\Phi'_x(i,j)}. \tag{2.12}$$

2.2.4 Ridge frequency estimation

In addition to the orientation image, another important parameter that is used in the construction of the Gabor filter is the local ridge frequency. The frequency image represents the local frequency of the ridges in a fingerprint. The first step in the frequency estimation stage is to divide the image into blocks of size $W \times W$. The next step is to project the grey-level values of all the pixels located inside each block along a direction orthogonal to the local ridge orientation. This projection forms an almost sinusoidal-shape wave with the local minimum points corresponding to the ridges in the fingerprint. An example of a projected waveform is shown in Figure 2.2.

I have modified the original frequency estimation stage used by Hong et al. to include an additional projection smoothing step prior to computing the ridge spacing. This involves smoothing the projected waveform using a Gaussian low-pass filter of size $w \times w$ to reduce the effect of noise in the projection. The ridge spacing $S(i, j)$ is then computed by counting the median number of pixels between consecutive minima points in the projected waveform. Hence, the ridge frequency $F(i, j)$ for a block centred at pixel (i, j) is defined as:

$$F(i,j) = \frac{1}{S(i,j)}. \tag{2.13}$$

Given that the fingerprint is scanned at a fixed resolution, then ideally the ridge frequency values should lie within a certain range. However, there are cases where a valid frequency value cannot be reliably obtained from the projection. Examples are when no consecutive peaks can be detected from the projection, and also when minutiae points appear in the block. For the blocks where minutiae points appear, the projected waveform does not produce a well-defined sinusoidal-shape wave, which can lead to an inaccurate estimation of the ridge frequency. Thus, the out of range frequency values are interpolated using values from neighbouring blocks that have a well-defined frequency.

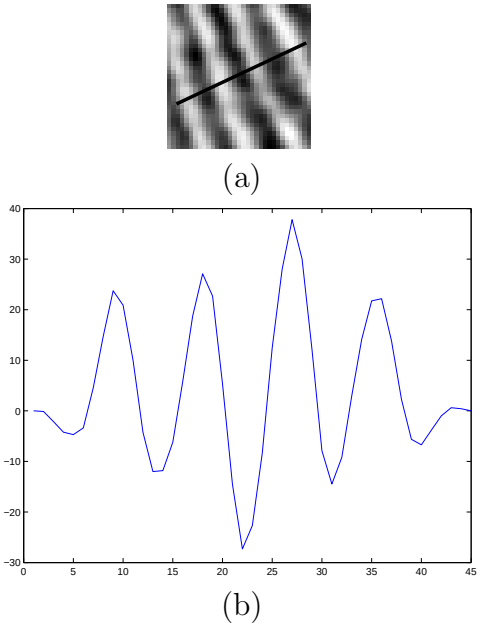


Figure 2.2: The projection of the intensity values of the pixels along a direction orthogonal to the local ridge orientation. (a) A 32×32 block from a fingerprint image. (b) The projected waveform of the block.

2.2.5 Gabor filtering

Once the ridge orientation and ridge frequency information has been determined, these parameters are used to construct the even-symmetric Gabor filter. A two-dimensional Gabor filter consists of a sinusoidal plane wave of a particular orientation and frequency, modulated by a Gaussian envelope [4]. Gabor filters are employed because they have frequency-selective and orientation-selective properties. These properties allow the filter to be tuned to give maximal response to ridges at a specific orientation and frequency in the fingerprint image. Therefore, a properly tuned Gabor filter can be used to effectively preserve the ridge structures while reducing noise.

The even-symmetric Gabor filter is the real part of the Gabor function, which is given by a cosine wave modulated by a Gaussian (see Figure 2.3). An even-symmetric Gabor filter in the spatial domain is defined as [10]:

$$G(x, y; \theta, f) = \exp\left\{-\frac{1}{2}\left[\frac{x_\theta^2}{\sigma_x^2} + \frac{y_\theta^2}{\sigma_y^2}\right]\right\} \cos(2\pi f x_\theta), \tag{2.14}$$

$$x_\theta = x\cos\theta + y\sin\theta, \tag{2.15}$$

$$y_\theta = -x\sin\theta + y\cos\theta, \tag{2.16}$$

where θ is the orientation of the Gabor filter, f is the frequency of the cosine wave, σ_x and σ_y are the standard deviations of the Gaussian envelope along the x and y axes, respectively, and x_θ and y_θ define the x and y axes of the filter coordinate frame, respectively.

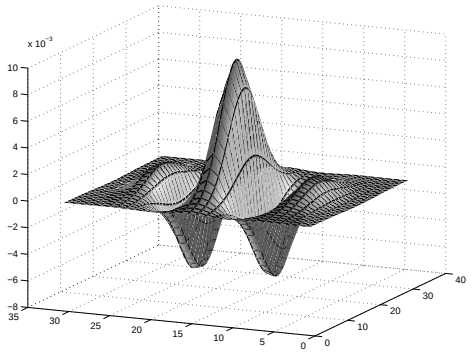


Figure 2.3: An even-symmetric Gabor filter in the spatial domain.

The Gabor filter is applied to the fingerprint image by spatially convolving the image with the filter. The convolution of a pixel (i, j) in the image requires the corresponding orientation value $O(i, j)$ and ridge frequency value $F(i, j)$ of that pixel. Hence, the application of the Gabor filter G to obtain the enhanced image E is performed as follows:

$$E(i,j) = \sum_{u=-\frac{w_x}{2}}^{\frac{w_x}{2}} \sum_{v=-\frac{w_y}{2}}^{\frac{w_y}{2}} G(u, v; O(i,j), F(i,j)) N(i-u, j-v), \tag{2.17}$$

where O is the orientation image, F is the ridge frequency image, N is the normalised fingerprint image, and w_x and w_y are the width and height of the Gabor filter mask, respectively.

The filter bandwidth, which specifies the range of frequency the filter responds to, is determined by the standard deviation parameters σ_x and σ_y. Since the bandwidth of the filter is tuned to match the local ridge frequency, then it can be deduced that the parameter selection of σ_x and σ_y should be related with the ridge frequency. However, in the original algorithm by Hong et al., σ_x and σ_y were empirically set to fixed values of 4.0 and 4.0, respectively.

A drawback of using fixed values is that it forces the bandwidth to be constant, which does not take into account the variation that may occur in the values of the ridge frequency. For example, if a filter with a constant bandwidth is applied to a fingerprint image that exhibits significant variation in the frequency values, it could lead to non-uniform enhancement or other enhancement artefacts. Thus, rather than using fixed values, I have chosen the values of σ_x and σ_y to be a function of the ridge frequency parameter, which are defined as:

$$\sigma_x = k_x F(i,j), \tag{2.18}$$

$$\sigma_y = k_y F(i,j), \tag{2.19}$$

where F is the ridge frequency image, k_x is a constant variable for σ_x, and k_y is a constant variable for σ_y. This allows a more adaptable approach to be used, as the values of σ_x and σ_y can now be specified adaptively according to the local ridge frequency of the fingerprint image.

Furthermore, in the original algorithm, the width and height of the filter mask were both set to fixed values of 11. The filter size controls the spatial extent of the filter, which ideally should be able to accommodate the majority of the

useful Gabor waveform information. However, a fixed filter size is not optimal in that it does not allow the accommodation of Gabor waveforms of different sized bandwidths. Hence, to allow the filter size to vary according to the bandwidth of the Gabor waveform, I have set the filter size to be a function of the standard deviation parameters:

$$w_x = 6\sigma_x, \tag{2.20}$$

$$w_y = 6\sigma_y, \tag{2.21}$$

where w_x and w_y are the width and height of the Gabor filter mask, respectively, and σ_x and σ_y are the standard deviations of the Gaussian envelope along the x and y axes, respectively. In the above equation, the width and height of the filter mask are both specified as 6σ, due to most of the Gabor wave information being contained within the region $[-3\sigma, 3\sigma]$ away from the y axis. Hence, this selection of parameters allows the filter mask to capture the majority of the Gabor waveform information.

2.2.6 Binarisation

Most minutiae extraction algorithms operate on binary images where there are only two levels of interest: the black pixels that represent ridges, and the white pixels that represent valleys. Binarisation is the process that converts a grey-level image into a binary image. This improves the contrast between the ridges and valleys in a fingerprint image, and consequently facilitates the extraction of minutiae.

One useful property of the Gabor filter is that it has a DC component of zero, which means the resulting filtered image has a mean pixel value of zero. Hence, straightforward binarisation of the image can be performed using a global threshold of zero. The binarisation process involves examining the grey-level value of each pixel in the enhanced image, and, if the value is greater than the global threshold, then the pixel value is set to a binary value one; otherwise, it is set to zero. The outcome is a binary image containing two levels of information, the foreground ridges and the background valleys.

2.2.7 Thinning

The final image enhancement step typically performed prior to minutiae extraction is thinning. Thinning is a morphological operation that successively erodes away the foreground pixels until they are one pixel wide. A standard thinning algorithm [7] is employed, which performs the thinning operation using two subiterations. This algorithm is accessible in MATLAB via the `'thin'` operation under the `bwmorph` function. Each subiteration begins by examining the neighbourhood of each pixel in the binary image, and based on a particular set of pixel-deletion criteria, it checks whether the pixel can be deleted or not. These subiterations continue until no more pixels can be deleted.

The application of the thinning algorithm to a fingerprint image preserves the connectivity of the ridge structures while forming a skeletonised version of the binary image. This skeleton image is then used in the subsequent extraction of minutiae. The process involving the extraction of minutiae from a skeleton image will be discussed in the next chapter.

2.3 Experimental results

2.3.1 Implementation environment

All the methods and algorithms described in this dissertation were implemented using MATLAB V6.5 on the Red Hat Linux operating system. The experiments were performed on a Pentium 4 - 2.4Ghz with 512MB of RAM. When testing the performance of the enhancement algorithm, the computational time was not measured. The aim of the experimental results section is to illustrate the results of each stage in the enhancement algorithm and to assess how well each stage performs.

Where appropriate, the experiments for some stages were conducted on a set of synthetic test images. Each synthetic image is formed by a series of circular patterns equally sized apart via the `circsine` function [14]. In addition, synthetic noise is generated via the MATLAB function `random` using a normal distribution. The orientation estimation (Section 2.3.4), ridge frequency estimation (Section 2.3.5), and Gabor filtering (Section 2.3.6) stages all employ the `circsine` function to generate the synthetic images, and use the `random` function to generate synthetic noise.

The majority of the real fingerprint images used in the experiments were obtained from the National Institute of Standards (NIST) fingerprint data set [6]. Each image is an 8-bit grey-level image scanned at approximately 500-dpi resolution and of size 832×768 pixels. For the purpose of conducting the experiments in MATLAB, the images were converted from their original Wavelet Scalar Quantisation (WSQ) format to the Portable Network Graphics (PNG) image format. In addition to the NIST data set, experiments were also conducted on images from the 2002 Fingerprint Verification Competition (FVC2002) database [15].

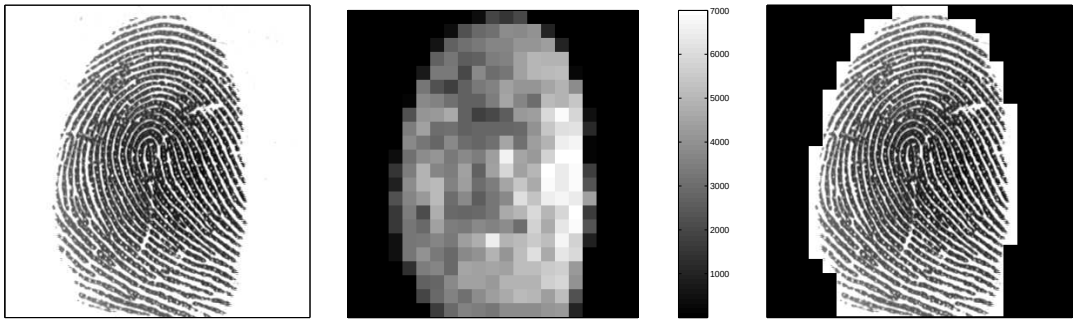
2.3.2 Segmentation

Figure 2.4 illustrates the results of segmenting a fingerprint image based on variance thresholding. The variance image in Figure 2.4(b) shows that the central fingerprint area exhibits a very high variance value, whereas the regions outside this area have a very low variance. Hence, a variance threshold is used to separate the fingerprint foreground area from the background regions. The final segmented image is formed by assigning the regions with a variance value below the threshold to a grey-level value of zero, as shown in Figure 2.4(c). These results show that the foreground regions segmented by this method comprise only of areas containing the fingerprint ridge structures, and that regions are not incorrectly segmented. Hence, the variance thresholding method is effective in discriminating the foreground area from the background regions.

There is a trade-off involved when determining the threshold value used to segment the image. If the threshold value is too large, results have shown that foreground regions may be incorrectly assigned as background regions. Conversely, if the threshold value is too small, background regions may be mistakenly assigned as part of the fingerprint foreground area. Hence, a variance threshold of 100 gives the optimal results in terms of differentiating between the foreground and background regions.

2.3.3 Normalisation

For all images submitted to the image enhancement process: a desired mean value of zero, and a variance of one are used for normalisation. Therefore, each image is normalised to a predetermined level before proceeding on to the subsequent enhancement stages. Figure 2.5 shows the results of normalising a fingerprint image so that it has a desired mean of zero and a variance of one. The histogram of the original image (Figure 2.5(c)) illustrates that all the intensity values lie on the right hand side of the 0–255 scale, with no pixels in the left hand side. This



(a) Original image (b) Variance image (c) Segmented image

Figure 2.4: The result of segmentation using a variance threshold of 100 and a block size of 16×16.

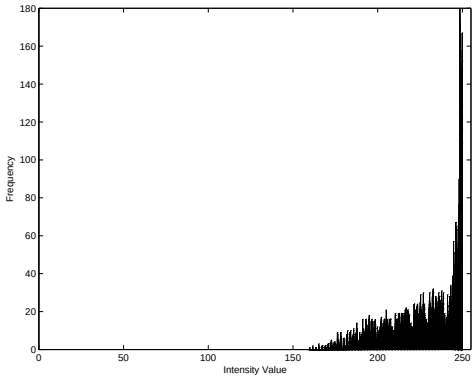
results in the image having a very low contrast, as shown in Figure 2.5(a). On the other hand, the histogram of the normalised image (Figure 2.5 (d)) shows that the range of intensity values has been adjusted such that there is a more balanced distribution between the dark and light pixels. Hence, normalising the image improves the contrast between the ridges and valleys, as shown in Figure 2.5(b). Additionally, the histograms plots in Figure 2.5(c) and Figure 2.5(d) show that the normalisation process does not alter the shape of the original histogram plot; only the relative position of the values along the x axis is shifted, which means the structure of the ridges and valleys are not changed.

2.3.4 Orientation estimation

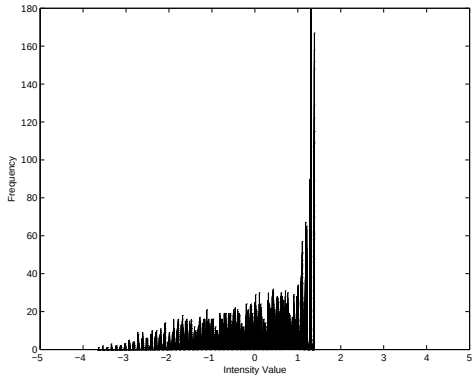
The Gabor filtering stage of the enhancement process relies heavily on filtering along the local ridge orientation in order to enhance the ridge structure and reduce noise. Hence, it is important to obtain an accurate estimation of the orientation field. As the orientation estimation stage plays a central role in the enhancement process, I have conducted an extensive series of experiments to evaluate the performance of the orientation estimation algorithm. A combination of both synthetic test images and real fingerprint images were used in the experiments. The default set of parameters specified by Hong et al. were used throughout the experiments: an averaging block size of 16×16, and a Gaussian filter size of 5×5.

(a) Original image

(b) Normalised image



(c) Histogram of the original image



(d) Histogram of the normalised image

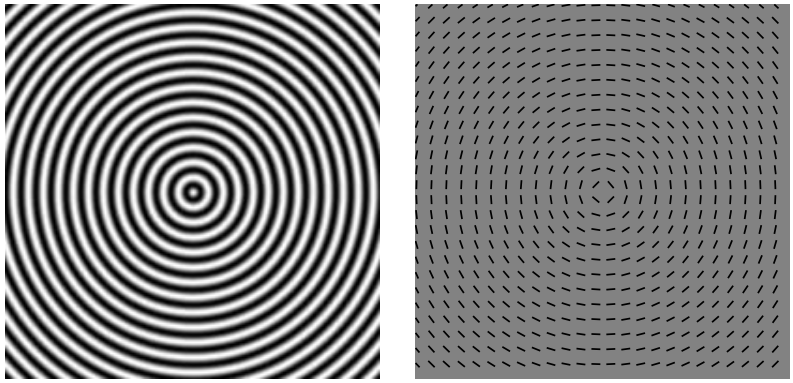
Figure 2.5: The result of normalisation using a desired mean and variance of zero and one, respectively.

Synthetic test image results

The first set of experiments were conducted on a set of circular synthetic images. By using synthetic images, the pre-determined orientation values can serve as a basis for comparison with the orientation values computed by the orientation estimation algorithm. The accuracy of the algorithm can be quantitatively measured by the mean square error between the estimated and actual values. The mean square error represents the difference between the actual orientation values and the estimated orientation values in radians.

The algorithm was firstly tested on a number of well-defined synthetic images. The results shown in Figure 2.6 were achieved using different image sizes and wavelengths to construct the synthetic images. Visual inspection of these results

shows that the estimated orientation field is smooth and well defined throughout the image. Furthermore, the small mean square error values indicate that there is minimal difference between the actual orientation values and the estimated values from the algorithm. Thus, it can be shown that the algorithm is able to produce accurate orientation estimates for synthetically well-defined images.



(a) Original image (b) Orientation image

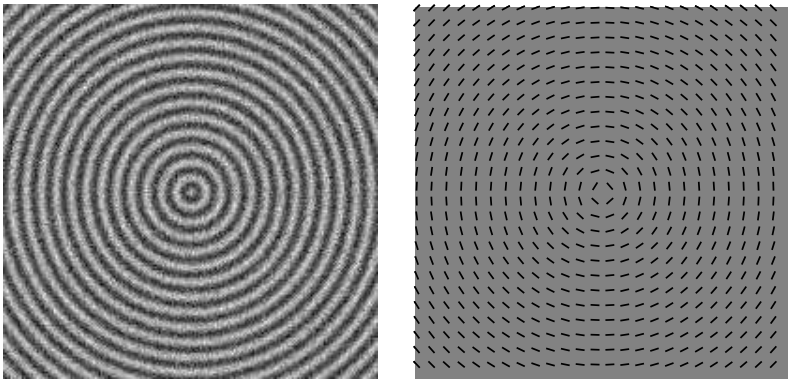
(c) Original image (d) Orientation image

Figure 2.6: The estimated orientation for well-defined synthetic images. (a) A 200×200 synthetic image of wavelength 8. The mean square error between the actual and estimated orientation values is 0.0003 radians. (c) A 500×500 synthetic image of wavelength 15. The mean square error between the actual and estimated orientation values is 0.0002.

Experiments were then conducted with a series of random noisy elements applied to the images. Figure 2.7(b) shows that the presence of small amounts of noise does not significantly affect the accuracy of the algorithm. However, Figure 2.7(d) illustrates that when higher intensities of noise are applied to the

image, the algorithm produces a disordered orientation field with a high proportion of misaligned orientation vectors.

Table 2.1 shows the mean square error values with respect to increasing values of noise intensity when applied to a 200×200 sized synthetic image of wavelength eight. The large mean square errors indicate that the accuracy of the algorithm decreases significantly in the presence of high intensities of noise. Therefore, it can be shown that the algorithm can produce accurate orientation estimates in the presence of minimal amounts of noise, but its performance deteriorates under high levels of noise.



(a) Original image (b) Orientation image

(c) Original image (d) Orientation image

Figure 2.7: The estimated orientation for 200×200 synthetic images of wavelength 8. Random noise with standard deviation values of 0.5 (top row), and 3 (bottom row) are applied to the images, respectively.

Standard Deviation	Mean Square Error
0.0	0.0003
0.5	0.0009
1.0	0.0032
1.5	0.0102
2.0	0.0246
2.5	0.0691
3.0	0.1722
3.5	0.2330
4.0	0.3041
4.5	0.4124
5.0	0.4262

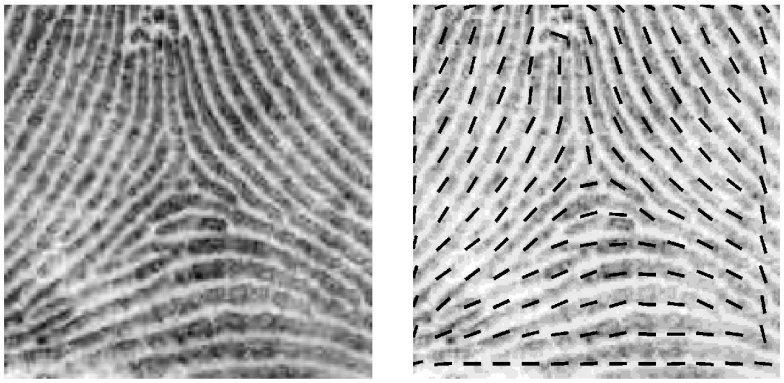
Table 2.1: The mean square error values (radians) for varying intensities of noise. The noise is applied to a 200×200 sized synthetic image of wavelength 8.

Real fingerprint image results

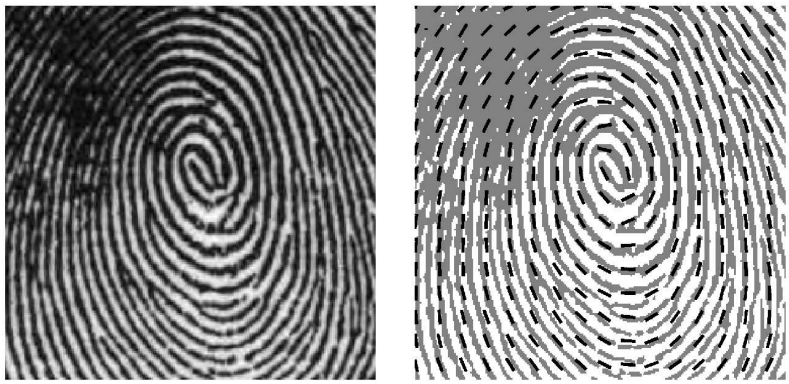
The second set of experiments was conducted on a series of real fingerprint images. A diverse mixture of fingerprint images was selected to assess the algorithm's performance under various fingerprinting conditions. The types of fingerprint images selected include well-defined images containing various singular points, and low quality images. In contrast to synthetic test images, when performing the experiments on real fingerprint images, the results can only be assessed on a qualitative scale by human inspection of the visual quality of the orientation field. The factors used in determining the quality of the orientation include how well the orientation vectors are oriented in respect to the direction of the ridge flow pattern, and the smoothness of the orientation field.

Singular points are prominent features of fingerprints and are widely used for fingerprint classification and matching; hence, it is important to obtain a reliable estimation of the orientation field around these points. Singular points are the points in a fingerprint where the orientation field is discontinuous and unlike the normal ridge flow pattern, the ridge orientation varies significantly. I have performed experiments on a variety of well-defined images containing singular points such as the tented arch (see Figure 2.8(a)) and whorl (see Figure 2.8(b)). From the superimposed image results of each figure, it can be seen that there is little deviation between the actual ridge orientation and the estimated orientation of the vectors. Thus, the algorithm produces an estimation of the orientation vectors such that they flow smoothly and consistently with the direction of the

ridge structures throughout the entire image. In addition, the results illustrate that although the orientation varies significantly around the singular point, the algorithm can still produce an accurate and smooth orientation field estimation.



(a) Tented arch



(b) Whorl

Figure 2.8: The estimated orientation for well-defined fingerprint images containing different singular points. Note that in the superimposed version of each image, a lower contrast version of the original image is used for the background. This is done to enhance the visibility of the orientation vectors against the background.

Well-defined images exhibit a smooth and consistent ridge flow pattern, which allows the orientation field to be reliably extracted. In contrast, for images containing elements of noise, breaks in ridges, and other defects, the ridge flow pattern is not consistently defined throughout the image. In the presence of small amounts of noise (right hand side of each image in Figure 2.9), the results indicate a fairly smooth orientation field that is not greatly affected by the corrupted elements. However, the orientation estimation gives misleading results particu-

larly in areas where there is a large section of corruption evident (left hand side of each image in Figure 2.9).

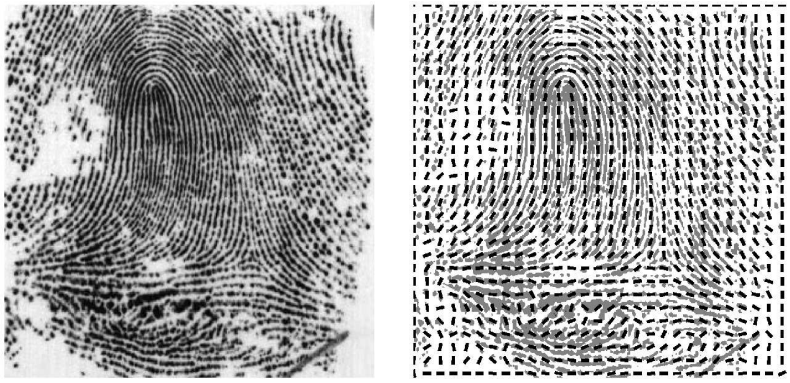


Figure 2.9: The estimated orientation for a low quality image containing noise and corrupted elements.

2.3.5 Ridge frequency estimation

Together with the ridge orientation, the ridge frequency is another important parameter used in the construction of the Gabor filter. Experiments are conducted primarily on synthetic test images to evaluate the performance of the ridge frequency estimation. Note that the results for the ridge frequency values will be presented in terms of ridge wavelength for easier interpretation of the results. For example, if the ridge spatial frequency value is $\frac{1}{8}$ pixels, then the results will present this as a ridge wavelength of 8.

Additional smoothing results

A modification was made to the original wavelength estimation stage used by Hong et al. to include smoothing of the projected waveform prior to computing the ridge wavelength. As shown in Figure 2.10(b), the presence of noisy bumps in the original projection result in the creation of false local minima, which mask out the location of the true minimum points. These false minima can then lead to an inaccurate estimation of the ridge wavelength. However, as illustrated by Figure 2.10(c), if the projection is smoothed prior to estimating the wavelength, then the noisy bumps are eliminated, leaving only the true local minimum points. Hence, this additional step has shown to be useful in reducing the effect of noise

in the projection, and can subsequently improve the accuracy of the wavelength estimation.

Synthetic test image results

The wavelength estimation stage was firstly tested on a number of synthetic test images. Figure 2.11 illustrates the results of estimating the ridge wavelength for a well defined image, and Figure 2.12 shows the results of applying a series of random noisy elements to the image. Table 2.2 shows the mean square error values with respect to increasing values of noise intensity when applied to a 500×500 sized synthetic image of wavelength eight. The mean square error represents the difference between the actual wavelength values and the estimated wavelength values in pixels.

The results for well-defined images (see Figure 2.11(b)) and low intensity noisy images (see Figure 2.12(b)) show that the majority of the estimated wavelength values for each 32×32 block match up accurately with the actual wavelength value of 10. Furthermore, the small mean square error values indicate that there is minimal difference between the actual wavelength values and the estimated wavelength values. Hence, it can be shown that the wavelength estimation is accurate for both well-defined images and low intensity noisy images.

However, the accuracy of the wavelength estimation deteriorates when high intensities of noise are applied to the image. Visual inspection of Figure 2.12(d) shows that there is a large proportion of image blocks which give inaccurate wavelength estimate values. Additionally, the large mean square errors shown in Table 2.2 indicate that the accuracy of the estimation decreases considerably in the presence of high levels of noise.

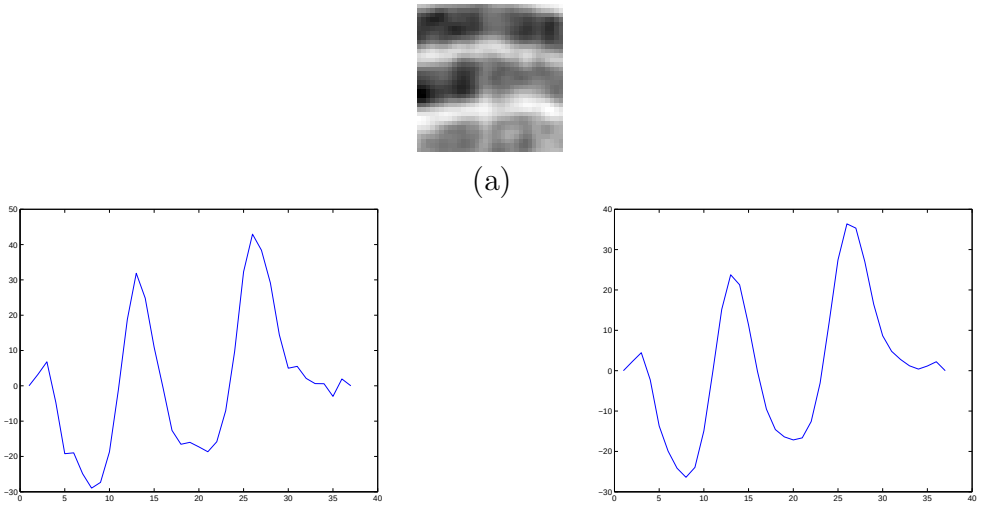


Figure 2.10: The effect of smoothing the projection prior to computing the ridge wavelength. Note how the false local minima have now been removed. (a) A 32×32 block from a fingerprint image. (b) The original projection of the block with a computed ridge wavelength of 5 pixels. (c) The smoothed projection with computed ridge wavelength of 13 pixels.

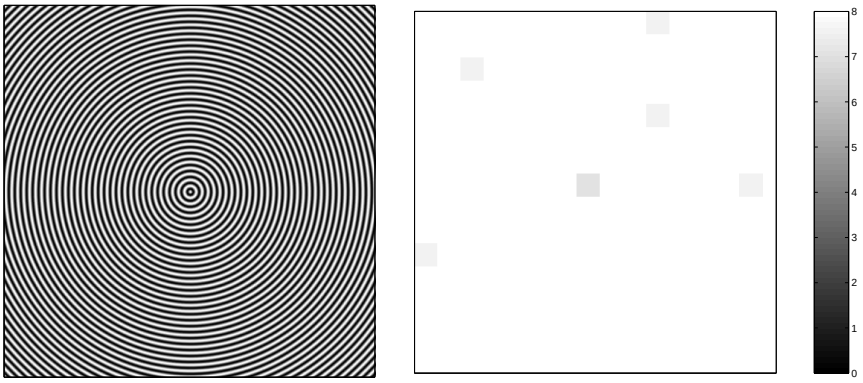
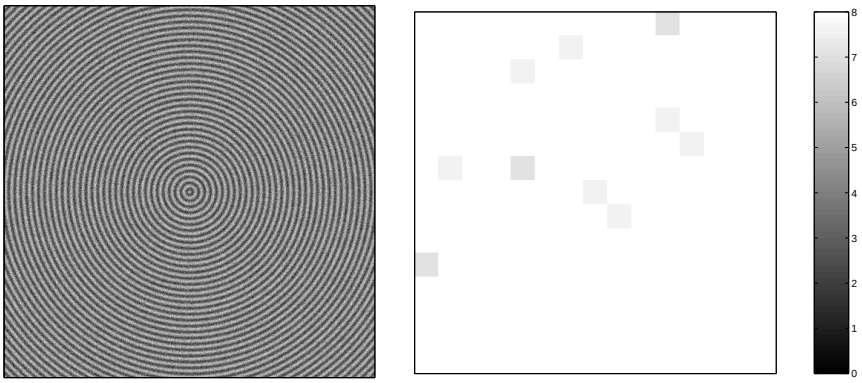
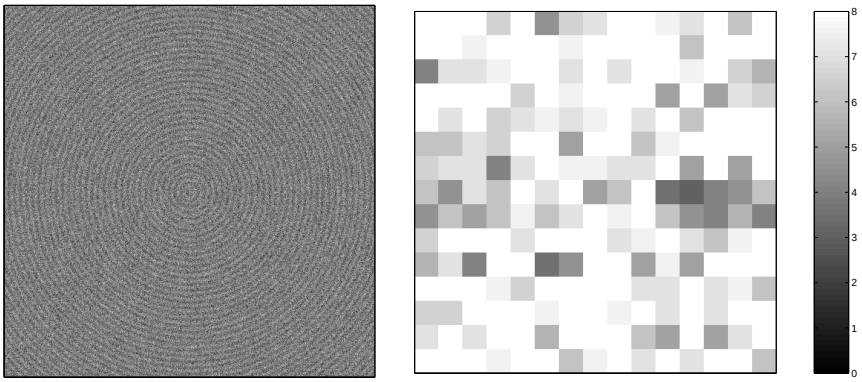


Figure 2.11: The estimated ridge wavelength for a 500×500 synthetic image of wavelength 8.

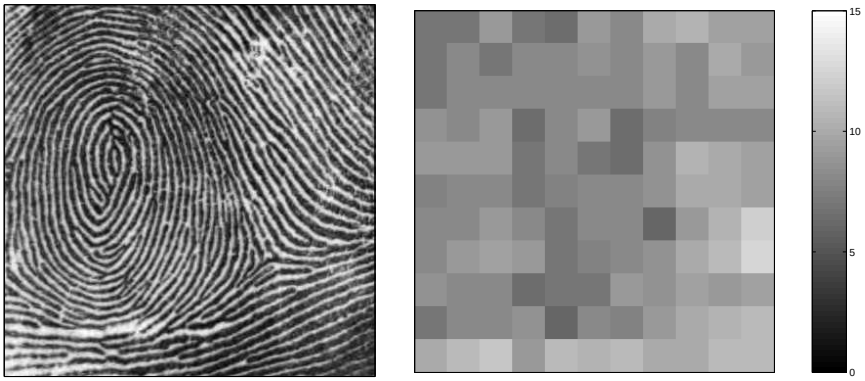


(a) Original image (b) Wavelength image

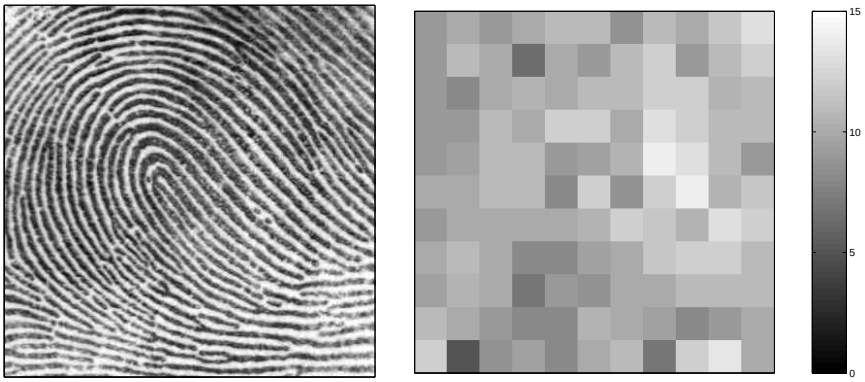


(c) Original image (d) Wavelength image

Figure 2.12: The estimated ridge wavelength for 500×500 synthetic images of wavelength 8. Random noise with standard deviation values of 0.5 (top row), and 3 (bottom row) are applied to the images, respectively.



(a) First image



(a) Second image

Figure 2.13: The estimated ridge wavelength for two different fingerprint images. (a) Mean ridge wavelength of 8.6130 pixels. (b) Mean ridge wavelength of 10.3233 pixels.

Standard Deviation	Mean Square Error
0.0	0.0100
0.5	0.0211
1.0	0.0465
1.5	0.0820
2.0	0.1702
2.5	1.1149
3.0	2.0229
3.5	4.1149
4.0	5.8543
4.5	6.1098
5.0	7.2616

Table 2.2: The mean square error values for varying intensities of noise. The mean square error is the difference between the actual wavelength values and the estimated wavelength values in pixels. The noise is applied to a 500×500 sized synthetic image of wavelength 8.

Real fingerprint image results

Conducting experiments on synthetic images provides a quantitative and accurate measure of the performance of the wavelength estimation. Synthetic images are constructed from a set of predetermined parameters, which allows comparisons to be made between estimated values and actual values. In contrast, when performing experiments on real fingerprint images, the results are more difficult to assess due to a number of reasons. Firstly, the wavelength of the ridges in a fingerprint image is not of a fixed value and can vary across different regions in a fingerprint. Figure 2.13 illustrates the ridge wavelength image for two different fingerprint images. For both of these images, it can be seen that the ridge wavelength varies throughout the image. In addition, not all fingerprints exhibit the same average ridge wavelength and consequently, different ridge wavelength values may result from different fingerprints. For example, Figure 2.13 shows that the average ridge wavelength values between the two images are significantly different from each other. Hence, only synthetic images are used for evaluating the performance of the ridge wavelength estimation.

2.3.6 Gabor filtering

The central part of the enhancement algorithm lies within the Gabor filtering stage. This is the stage that performs the actual enhancement of the fingerprint image. The purpose of the filtering stage is to enhance the clarity of the ridge structures while reducing noise in the image. In order to assess the performance of the Gabor filtering, I have conducted experiments on both synthetic test images and real fingerprint images.

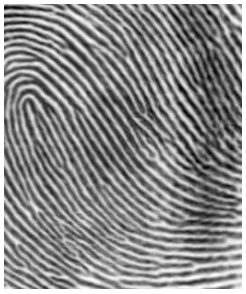
Parameter selection

The Gabor filter parameters σ_x and σ_y control the bandwidth of the filter, and must be chosen carefully as they have a significant effect on the enhancement results. The value of σ_x determines the degree of contrast enhancement between ridges and valleys, and σ_y determines the amount of smoothing applied to the ridges along the local orientation. Figure 2.14 illustrates the results of using different values of σ_x and σ_y to apply the Gabor filter to a fingerprint image. Recall from section 2.2.5 that σ_x and σ_y are determined by $\sigma_x = k_x F(i,j)$ and $\sigma_y = k_y F(i,j)$, where F is the ridge frequency image, and k_x and k_y are constant variables.

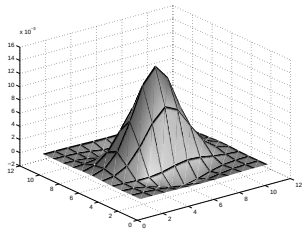
Figure 2.14(f) shows that large values of σ_x and σ_y lead to enhancement artefacts and a significant amount of blurring of the ridge structures. This blurring occurs due to the over-smoothing of the image by the Gabor filter. On the other hand, if the values are too small, the filter is not effective in removing noise from the image as the resulting image is simply a smoothed version of the original image (see Figure 2.14(b)). This smoothing of the image occurs due to the Gabor filter evolving into the shape of a pure low pass filter (see Figure 2.14(c)).

Hence, it can be seen that the selection of σ_x and σ_y involves a trade-off between values that are too small and values that are too large. Experiments conducted with the Gabor filter for varying values of σ_x and σ_y have shown that using $k_x = 0.5$ and $k_y = 0.5$ provides a reasonable trade-off. An example of an enhanced image using these parameters is given in Figure 2.14(d) with its corresponding Gabor filter in Figure 2.14(e). These results indicate a well enhanced ridge definition and an improved contrast between the ridge and valley structures.

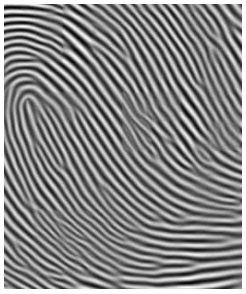
(a) Original image



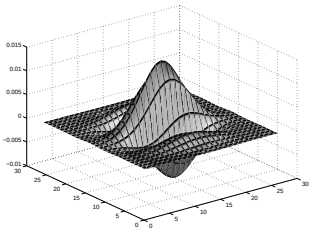
(b) Enhanced image ($k_x = 0.2$, $k_y = 0.2$)



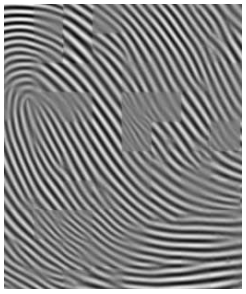
(c) Gabor filter ($k_x = 0.2$, $k_y = 0.2$)



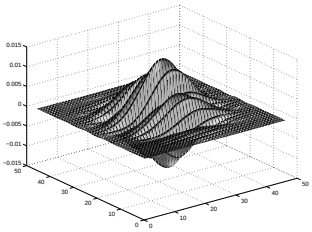
(d) Enhanced image ($k_x = 0.5$, $k_y = 0.5$)



(e) Gabor filter ($k_x = 0.5$, $k_y = 0.5$)



(f) Enhanced image ($k_x = 0.9$, $k_y = 0.9$)

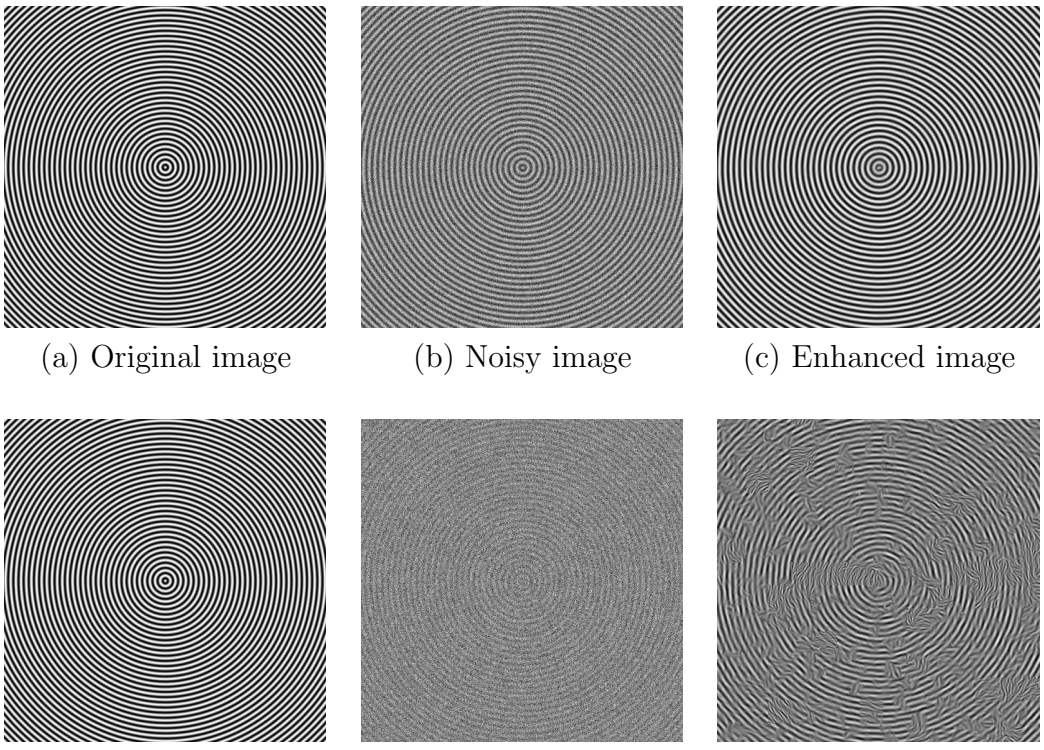


(g) Gabor filter ($k_x = 0.9$, $k_y = 0.9$)

Figure 2.14: Enhancement results using the Gabor filter for different parameter values of k_x and k_y.

Synthetic test image results

The results of applying the Gabor filter to synthetic noisy images are illustrated in Figure 2.15. Figure 2.15(c) shows that under low levels of noise, the filter is able to effectively remove the noise from the image and enhance it to a level that is comparable with the original image. This effective removal of noise is partly due to the accurate estimation of the ridge orientation and frequency for low-level noisy images, as shown in Section 2.3.5. On the other hand, when the filter is applied to images with high intensities of noise (see Figure 2.15(f)), the results show that the filter is not able to effectively remove the noise, and produces a significant amount of spurious features. This poor enhancement of the image is due to the inaccurate estimation of the ridge orientation and frequency, which occurs under high contamination of noise, as demonstrated in the previous sections.



(a) Original image (b) Noisy image (c) Enhanced image

(d) Original image (e) Noisy image (f) Enhanced image

Figure 2.15: Results of applying a Gabor filter with $k_x = 0.5$ and $k_y = 0.5$ to 500×500 synthetic images of wavelength 8. Random noise with standard deviation values of 0.5 (top row), and 3 (bottom row) are applied to the images, respectively.

Real fingerprint image results

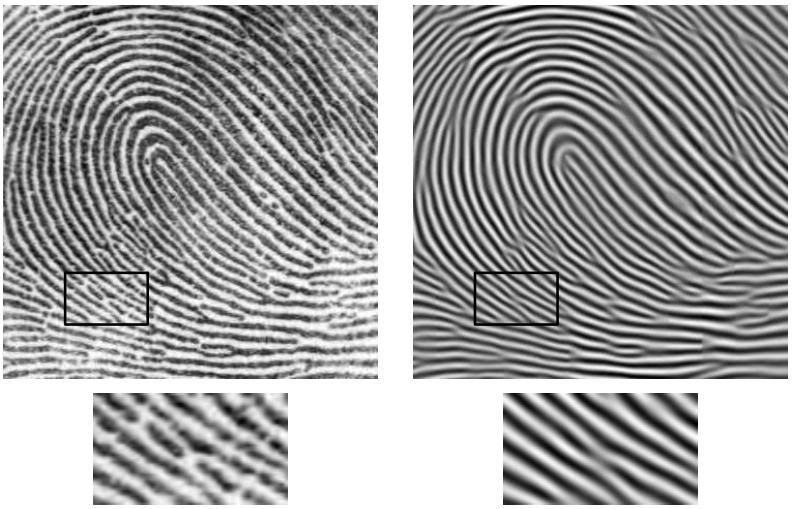
Figure 2.16 illustrates the application of the Gabor filter to a medium quality fingerprint image. The enhancement results show that the filter preserves the continuity of the ridge flow pattern and enhances the clarity of the ridge and valley structures. In addition to reducing noise in the image, the filter is able to fill in small breaks that occur within ridges, as shown by the enlarged sections of the image.

In contrast, the performance of the enhancement algorithm deteriorates when applied to low quality images, as shown in Figure 2.17. It can be seen that the filter has difficulty with regions of the image that are severely corrupted and degraded (bottom right of Figure 2.17), resulting in ineffective image enhancement. Hence, in practice, images of low quality are typically discarded in fingerprint identification systems. Likewise, when performing statistical experiments for this project, I have chosen to exclude images that are of low quality.

The results in Figure 2.16(b) also show that although the clarity of the ridge patterns are well enhanced, the enhancement at the minutiae points are slightly blurred. The shape of the Gabor filter is designed to enhance along the ridge lines that are parallel to one another and have a consistent orientation. However, minutiae points occur as local discontinuities in the ridge flow pattern, which can cause the local orientation and frequency to be inaccurately estimated. Consequently, when compared to minutiae-free regions, the results of applying the filter to regions containing minutia points is less effective in enhancing the image.

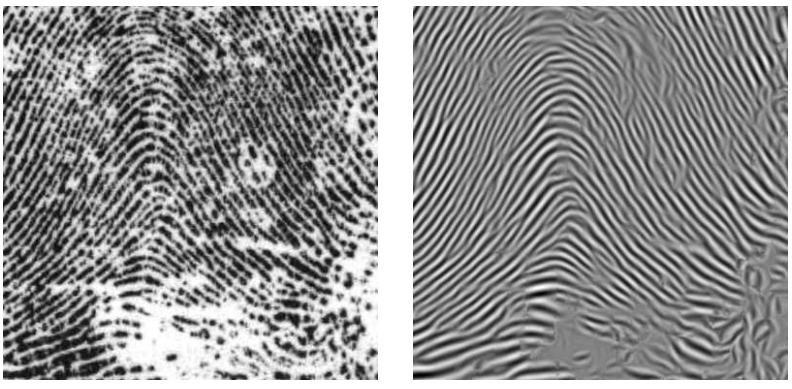
2.3.7 Binarisation and thinning

After the fingerprint image is enhanced, it is then converted to binary form, and submitted to the thinning algorithm which reduces the ridge thickness to one pixel wide. Figure 2.18(b) demonstrates that the global thresholding technique is effective in separating the ridges (black pixels) from the valleys (white pixels). The results of thinning show that the connectivity of the ridge structures is well preserved, and that the skeleton is eight-connected throughout the image (see Figure 2.18(c)). In particular, Figure 2.19 shows that the thinning algorithm is able to accurately extract the skeleton of minutia points without disrupting the continuity of the ridge flow pattern.



(a) Original image (b) Enhanced image

Figure 2.16: Results of applying a Gabor filter with $k_x = 0.5$ and $k_y = 0.5$ to a medium quality fingerprint image.

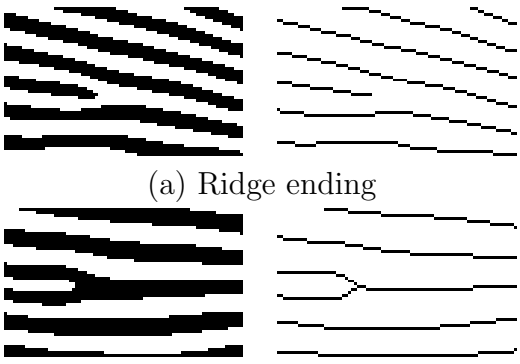


(a) Original image (b) Enhanced image

Figure 2.17: Results of applying a Gabor filter with $k_x = 0.5$ and $k_y = 0.5$ to a low quality fingerprint image.

(a) Enhanced image (b) Binary image (c) Thinned image

Figure 2.18: Results of applying binarisation and thinning to the enhanced image. A global threshold of zero is used to perform the binarisation.



(a) Ridge ending

(b) Bifurcation

Figure 2.19: Results of applying binarisation and thinning to a ridge ending, and a bifurcation. These two minutiae regions are from Figure 2.18.

Figure 2.20 illustrates the result of applying binarisation and thinning to a fingerprint image without any pre-processing stages such as image enhancement. In contrast to Figure 2.18(b), the binary image in Figure 2.20(b) is not well connected and contains significant amounts of noise and corrupted elements. Consequently, when thinning is applied to this binary image, the results in Figure 2.20(c) show that the accurate extraction of minutiae from this image would not be possible due to the large number spurious features produced. Thus, it can be shown that employing a series of image enhancement stages prior to thinning is effective in facilitating the reliable extraction of minutiae.

(a) Original image (b) Binary image (c) Thinned image

Figure 2.20: Results of applying binarisation and thinning directly to the original image (without enhancement). A global threshold of zero is used to perform the binarisation.

CHAPTER 3

Minutiae Extraction and Image Postprocessing

After a fingerprint image has been enhanced, the next step is to extract the minutiae from the enhanced image. Following the extraction of minutiae, a final image postprocessing stage is performed to eliminate false minutiae. This chapter provides discussion on the methodology and implementation of techniques for minutiae extraction and fingerprint image postprocessing. The first section contains a review of existing literature in the field of minutiae extraction and postprocessing. The next section discusses the methodology for implementing each of these two techniques. The last section presents the results from the experiments conducted using the implemented techniques.

3.1 Literature review

3.1.1 Minutiae extraction

The most commonly employed method of minutiae extraction is the Crossing Number (CN) concept [1, 16, 20]. This method involves the use of the skeleton image where the ridge flow pattern is eight-connected. The minutiae are extracted by scanning the local neighbourhood of each ridge pixel in the image using a 3×3 window. The CN value is then computed, which is defined as half the sum of the differences between pairs of adjacent pixels in the eight-neighbourhood. Using the properties of the CN as shown in Table 3.1, the ridge pixel can then be classified as a ridge ending, bifurcation or non-minutiae point. For example, a ridge pixel with a CN of one corresponds to a ridge ending, and a CN of three corresponds to a bifurcation.

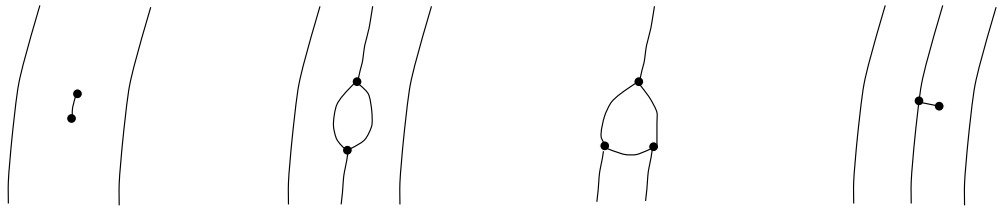
CN	Property
0	Isolated point
1	Ridge ending point
2	Continuing ridge point
3	Bifurcation point
4	Crossing point

Table 3.1: Properties of the Crossing Number.

Other authors such as Jain et al. [9] and Ratha et al. [18] have also performed minutiae extraction using the skeleton image. Their approach involves using a 3×3 window to examine the local neighbourhood of each ridge pixel in the image. A pixel is then classified as a ridge ending if it has only one neighbouring ridge pixel in the window, and classified as a bifurcation if it has three neighbouring ridge pixels. Consequently, it can be seen that this approach is very similar to the Crossing Number method.

3.1.2 Fingerprint image postprocessing

False minutiae may be introduced into the image due to factors such as noisy images, and image artefacts created by the thinning process. Hence, after the minutiae are extracted, it is necessary to employ a postprocessing stage in order to validate the minutiae. Figure 3.1 illustrates some examples of false minutiae structures, which include the spur, hole, triangle and spike structures [27]. It can be seen that the spur structure generates false ridge endings, where as both the hole and triangle structures generate false bifurcations. The spike structure creates a false bifurcation and a false ridge ending point.



(a) Spur (b) Hole (c) Triangle (d) Spike

Figure 3.1: Examples of typical false minutiae structures.

The majority of the proposed approaches for image postprocessing in literature are based on a series of structural rules used to eliminate spurious minutiae. One such approach is the one proposed by Ratha et al. [18], which performs the validation of minutiae based on a set of heuristic rules. For example, a ridge ending point that is connected to a bifurcation point, and is below a certain threshold distance is eliminated. This heuristic rule corresponds to removal of the spike structure shown in Figure 3.1(d). Additional heuristic rules are then used to eliminate other types of false minutiae. Furthermore, a boundary effect treatment is applied where the minutiae below a certain distance from the boundary of the foreground region are deleted.

A novel approach to the validation of minutiae is the postprocessing algorithm proposed by Tico and Kuosmanen [25]. Similar to the above techniques, this algorithm operates on the skeleton image. However, rather than employing a different set of heuristics each time to eliminate a specific type of false minutiae, this approach incorporates the validation of different types of minutiae into a single algorithm. It tests the validity of each minutiae point by scanning the skeleton image and examining the local neighbourhood around the minutiae. The algorithm is then able to cancel out false minutiae based on the configuration of the ridge pixels connected to the minutiae point.

Rather than using a set of ad hoc techniques to validate the minutiae, I have chosen to employ the algorithm employed by Tico and Kuosmanen. Further details regarding this algorithm will be discussed in Section 3.2.2.

3.2 Methodology

This section describes the methodology for performing the minutiae extraction and image postprocessing stages. The minutiae extraction technique I have implemented is based on the widely employed Crossing Number method. For the image postprocessing stage, I have implemented the minutiae validation algorithm by Tico and Kuosmanen. Firstly, the minutiae extraction method will be discussed, followed by details of the minutiae validation algorithm.

3.2.1 Minutiae extraction

The Crossing Number (CN) method is used to perform minutiae extraction. This method extracts the ridge endings and bifurcations from the skeleton image by examining the local neighbourhood of each ridge pixel using a 3×3 window. The CN for a ridge pixel P is given by [24]:

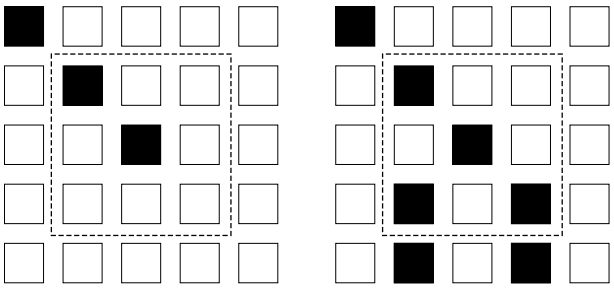
$$CN = 0.5\sum_{i=1}^{8}|P_i - P_{i+1}|, \qquad P_9 = P_1 \tag{3.1}$$

where P_i is the pixel value in the neighbourhood of P. For a pixel P, its eight neighbouring pixels are scanned in an anti-clockwise direction as follows:

P_4	P_3	P_2
P_5	P	P_1
P_6	P_7	P_8

After the CN for a ridge pixel has been computed, the pixel can then be classified according to the property of its CN value. As shown in Figure 3.2, a ridge pixel with a CN of one corresponds to a ridge ending, and a CN of three corresponds to a bifurcation. For each extracted minutiae point, the following information is recorded:

- x and y coordinates,
- orientation of the associated ridge segment, and
- type of minutiae (ridge ending or bifurcation).



(a) CN = 1 (b) CN = 3

Figure 3.2: Examples of a ridge ending and bifurcation pixel. (a) A Crossing Number of one corresponds to a ridge ending pixel. (b) A Crossing Number of three corresponds to a bifurcation pixel.

3.2.2 Fingerprint image postprocessing

In order to eliminate false minutiae, I have chosen to implement the minutiae validation algorithm by Tico and Kuosmanen. This algorithm tests the validity of each minutiae point by scanning the skeleton image and examining the local neighbourhood around the point. The first step in the algorithm is to create an image M of size $W \times W$, where M corresponds to the $W \times W$ neighbourhood centred on the candidate minutiae point in the skeleton image. The central pixel of M corresponds to the minutiae point in the skeleton image, and so this pixel is labelled with a value of -1. The rest of the pixels in M are initialised to values of zero, as shown in Figure 3.3(a) and Figure 3.4(a). The subsequent steps of the algorithm depend on whether the candidate minutiae point is a ridge ending or a bifurcation.

1. For a candidate ridge ending point:

   (a) Firstly, label with a value of 1 all the pixels in M, which are eight-connected with the ridge ending point (see Figure 3.3(b)).

   (b) The next step is to count in a clockwise direction, the number of 0 to 1 transitions (T_{01}) along the border of image M. If $T_{01} = 1$, then the candidate minutiae point is validated as a true ridge ending.

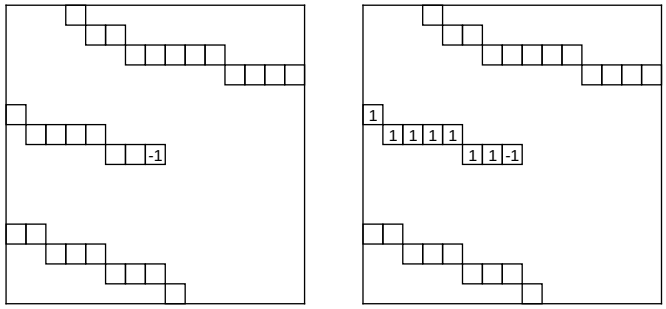
2. For a candidate bifurcation point:

   (a) Firstly, examine the eight neighbouring pixels surrounding the bifurcation point in a clockwise direction. For the three pixels that are connected with the bifurcation point, label them with the values of 1, 2, and 3, respectively. An example of this initial labelling process is shown in Figure 3.4(b).

   (b) The next step is to label the rest of the ridge pixels that are connected to these three connected pixels. This labelling is similar to the ridge ending approach, however, instead of labelling a single ridge branch, three ridge branches are now labelled. Let $l = 1$, 2 and 3 represent the label for each ridge branch. For each l, label with l all the ridge pixels that have a label of 0, and are connected to an l labelled pixel. Examples of the bifurcation labelling process are shown in Figures 3.4(c), (d) and (e).

   (c) The last step is to count in a clockwise direction, the number of transitions from 0 to 1 (T_{01}), 0 to 2 (T_{02}), and 0 to 3 (T_{03}) along

the border of image M. If $T_{01} = 1 \wedge T_{02} = 1 \wedge T_{03} = 1$, then the candidate minutiae point is validated as a true bifurcation.



(a) (b)

Figure 3.3: Example of validating a candidate ridge ending point. $T_{01} = 1$.

3.3 Experimental results

This section presents results of performing the minutiae extraction and image processing stages on a series of real fingerprint images. Experiments are firstly conducted to assess how well the Crossing Number (CN) technique is able to extract the minutiae from the skeleton image. The minutiae validation algorithm is then evaluated to see how effective the algorithm is in detecting the false minutiae.

3.3.1 Minutiae extraction

Figure 3.5 illustrates the results of extracting minutiae from a medium quality fingerprint image. From the skeleton image, it can be deduced that all ridge pixels corresponding to a CN of one and three have been detected successfully. Additionally, the results show that there are no candidate minutiae pixels that have been missed, and no pixels that have been falsely marked as minutiae pixels. Hence, it can be shown that the CN technique is able to accurately detect all valid bifurcations and ridge endings from the skeleton image.

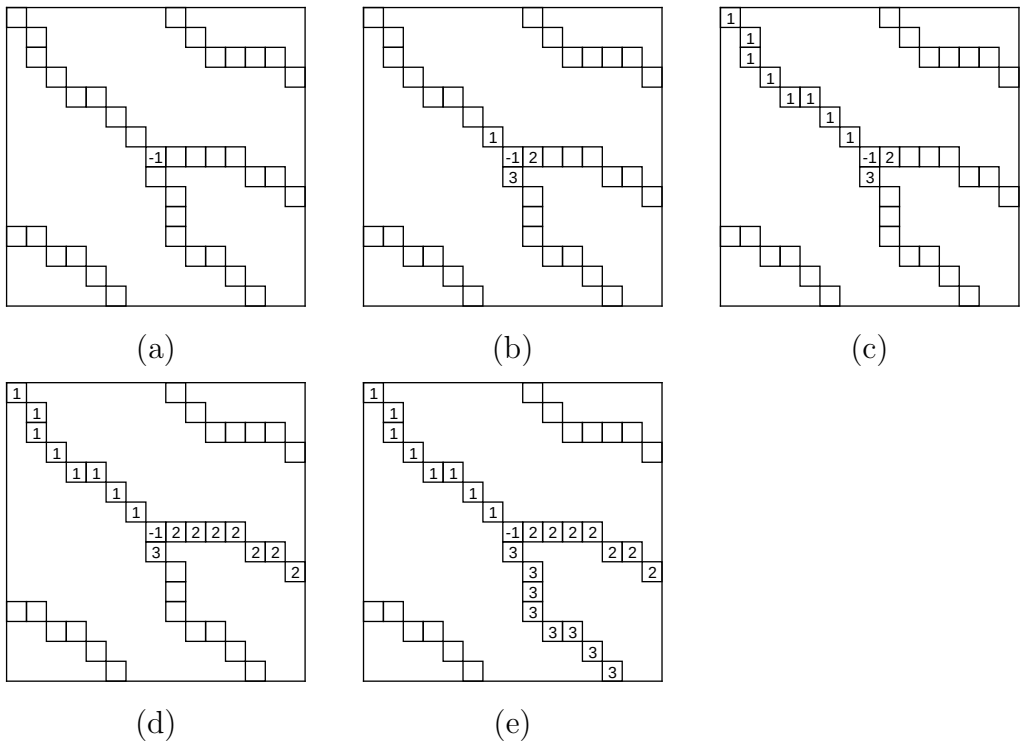


Figure 3.4: Example of validating a candidate bifurcation point. $T_{01} = 1 \wedge T_{02} = 1 \wedge T_{03} = 1$.

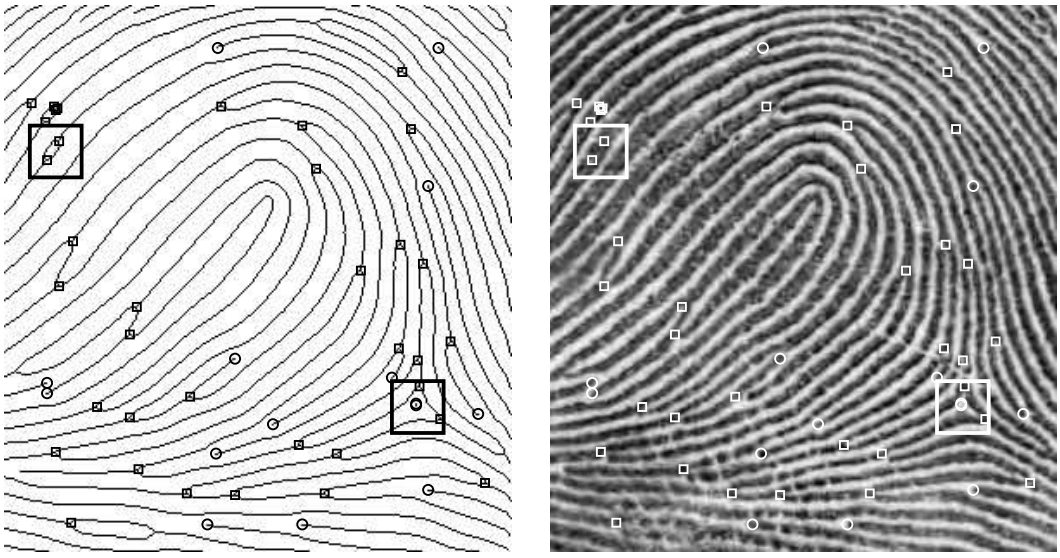
Figure 3.5(b) depicts the extracted minutiae points superimposed on the original image. Visual inspection of the image indicates that the majority of the marked minutiae points from the skeleton image correspond to valid minutiae points in the original image. However, there are a few cases where the extracted minutiae do not correspond to true minutiae points in the original image.

In addition, it should be noted that in some cases the bifurcation and ridge ending points can be difficult to distinguish between each other. Artefacts of the enhancement stage and thinning process can occasionally result in bifurcations being detected as ridge endings, and vice versa. Hence, in practice, most fingerprint identification systems do not make a distinction between bifurcations and ridge endings when matching minutiae points [11].

Two examples of false minutiae are marked with boxes in Figure 3.5, and an enlarged view of these false minutiae are shown in Figure 3.6. Figure 3.6(a) depicts a false minutiae point called a hole structure, which corresponds to the box in the top left hand corner of Figures 3.5(a) and (b). Figure 3.6(b) depicts a false minutiae point called a spur structure, which corresponds to the box in the bottom right corner of Figures 3.5(a) and (b). It can be seen that the hole structure generates two bifurcation points; however, in the original image (Figure 3.5(b)) it can be seen that these two minutiae points do not exist. Likewise, the spur structure generates two ridge endings, which are not present in the original image. Therefore, a postprocessing stage is required to validate the minutiae.

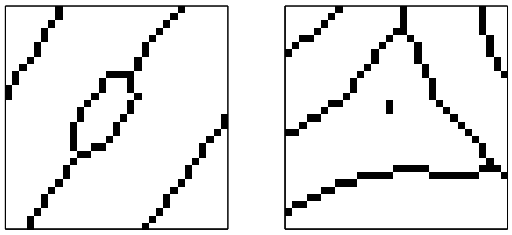
3.3.2 Fingerprint image postprocessing

Figure 3.7 illustrates the results of applying the postprocessing algorithm to the skeleton image. It can be seen from Figure 3.7(a) that the algorithm is able to cancel out the false bifurcations created by the two hole structures at the top left hand corner of the image, and the false ridge endings created by the spur structure. Furthermore, there are no minutiae that have been incorrectly marked as false minutiae.



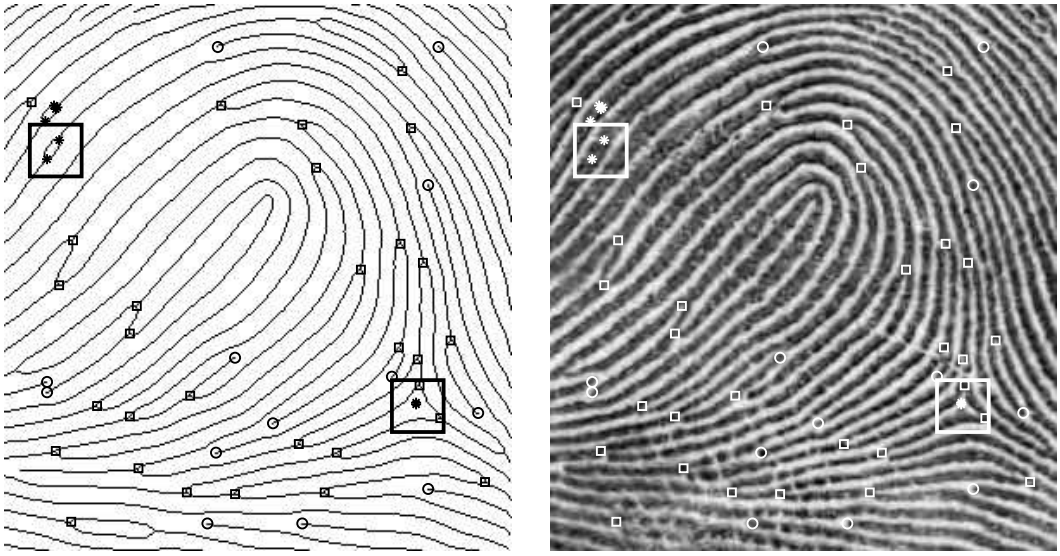
(a) Skeleton image (b) Original image

Figure 3.5: Results of performing minutiae extraction on a fingerprint image. Note that minutiae points which are close to the border (within 10 pixels) are ignored to avoid extracting minutiae which may be artefacts of the thinning process. Ridge endings are denoted by circles, and bifurcations are denoted by boxes.



(a) Hole structure (b) Spur structure

Figure 3.6: Enlarged view of the false minutiae from Figure 3.5(a).

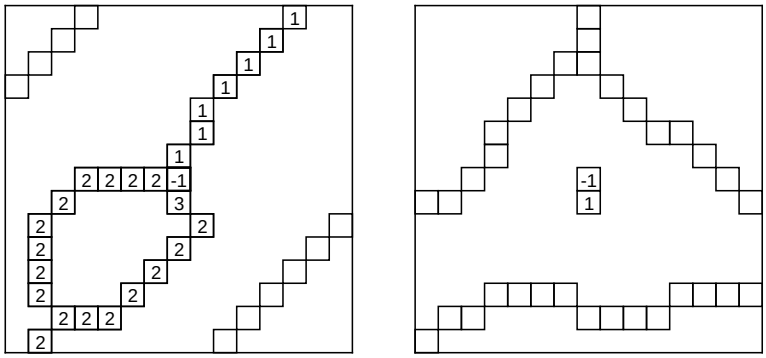


(a) Skeleton image (b) Original image

Figure 3.7: Results of performing minutiae validation. The false minutiae are denoted by asterisks. Ridge endings are denoted by circles, and bifurcations are denoted by boxes.

Details of how the algorithm is able to cancel out these two types of minutiae are shown in Figure 3.8. Each bifurcation point in the hole structure can be eliminated due to the number of zero to two transitions (T_{02}) along the border of the image window not being equal to one. The spur structure contains only two ridge pixels in the centre of the image window, which means the ridge branch connected to the minutiae is not long enough to reach the window border. Hence, the ridge ending points cannot be validated as true minutiae, since the number of zero to one transitions (T_{01}) along the border of the window is zero.

Previous experiments conducted for different sized neighbourhood windows have shown that a window of size 23×23 is most effective in eliminating the false minutiae. If the window size is too small, results have shown that the algorithm is not effective in cancelling out the false minutiae. Conversely, if the window size is too large, then the algorithm may incorrectly cancel out minutiae.



(a) Hole structure (b) Spur structure

Figure 3.8: An example of how the postprocessing algorithm cancels out the false minutiae. A window size of 23×23 is used. (a) $T_{01} = 1 \wedge T_{02} = 1 \wedge T_{03} = 0$. (b) $T_{01} = 0$.

CHAPTER 4

Statistics of Fingerprints

This chapter presents some preliminary results on the statistics of fingerprints. In previous chapters, I have discussed and implemented a set of techniques for fingerprint image enhancement and minutiae extraction. The collection of experimental data is facilitated by the use of these implemented techniques. The following three types of statistical data are collected and presented:

- Minutiae density (Section 4.1),
- Shortest distance between neighbouring minutiae (Section 4.2), and
- Ridge wavelength (Section 4.3).

In contrast to the widely studied and published techniques on fingerprint image enhancement and minutiae extraction, there is limited literature available on the statistics of fingerprints. An examination of the literature available has found that for the three types of statistical data collected in this project, previous work has been done only in the field of minutiae density. Therefore, the minutiae density results obtained in this study can be compared with previous studies.

All experiments discussed in this chapter are conducted on a set of 30 fingerprints images of size 832×768 pixels. This fingerprint data set comprises of 15 images containing the fingerprint whorl pattern, and 15 images containing the fingerprint loop pattern. Whorls and loops are chosen, as they encompass the two main classes of fingerprint patterns. Similar to the previous chapters, these images were obtained from the NIST fingerprint data set.

For each section, I firstly discuss the methods for collecting the statistical data, followed by the experimental results. Note that the primary aim of the project is to implement a series of techniques for fingerprint image enhancement and minutiae extraction. Hence, the statistical results in this chapter are only preliminary results, such that full analysis of the results is reserved for future work.

4.1 Minutiae density

4.1.1 Methodology

Each image is firstly processed using the implemented image enhancement algorithm, and the minutiae are then extracted from the enhanced image. The next step is to count for each fingerprint image, the total number of minutiae and the total number of foreground pixels. The foreground pixels are distinguished from the background pixels by segmenting the fingerprint into foreground and background regions, as discussed earlier in Section 2.2.1. The number of minutiae is divided by the number of foreground pixels, which gives the minutiae density in terms of minutiae per pixel^2. Given that the fingerprint images are scanned at 19.7 pixels per mm, the measurement units are then converted into the standard units of minutiae per mm^2.

4.1.2 Experimental results

Experiments conducted over a sample of 30 fingerprint images indicate a mean minutiae density of 0.204 minutiae per mm^2 with a standard deviation of 0.0285. Results from previous studies on the minutiae density of fingerprints are shown in Table 4.1. Additionally, Figure 4.1 depicts a Gaussian distribution plot for each of the four data sets shown in Table 4.1. These results show that the mean minutiae density of 0.2040 observed in this study, falls within the range of values collected from previous studies. However, the standard deviation value is significantly larger than the values shown in previous results.

The large difference between standard deviation values can be attributed to various factors such as the sample size used, and the method employed to collect the data. For example, the previous studies have used human inspection of the fingerprint image, where manual measurement is used to count the number of minutiae contained within a circular sampling region. Consequently, this method is more precise in measuring the minutiae density. In contrast, the data collected for this study is automated without any human intervention, which may lead to a greater variation in density values.

Data Set Number	Source of Data	Sample Size	Mean Minutiae Density (per mm^2)	Standard Deviation
1	Dankmeijer et al. [3]	1000	0.1900	0.0069
2	Stoney et al. [23]	412	0.2230	0.0045
3	Kingston [13]	100	0.2460	0.0084
4	Current study	30	0.2040	0.0285

Table 4.1: Comparison of the observed minutiae densities with previous studies. The current results are shown on the bottom row of the table.

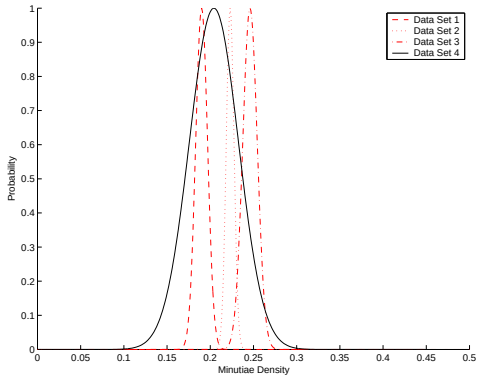


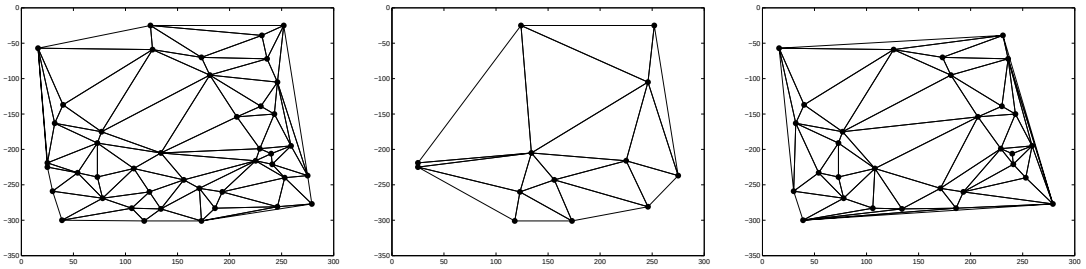
Figure 4.1: Gaussian distribution plot for each data set of minutiae density values from Table 4.1. The current results are depicted by the solid line curve.

4.2 Distance between neighbouring minutiae

4.2.1 Methodology

The focus of this section is to collect some statistical data on the distance between neighbouring minutiae points. Each image from the data set is firstly enhanced, and the minutiae are then extracted from the enhanced image. The next step is to find a method of calculating the distance from one minutiae to its nearest neighbouring minutiae point. Given that the minutiae points do not form a structured topology, the Delaunay triangulation method can be used to create a triangular grid for the scattered minutiae points. This method can be accessed in MATLAB via the `delaunay` function. The Delaunay triangulation creates a set of triangles such that the circumscribed circle of each triangle does not contain any other minutiae points. Therefore, the vertex of a triangle corresponds to a minutiae point, and the edge of the triangle joins together two neighbouring minutiae points.

The Delaunay triangulation for a sample set of minutiae points is shown in Figure 4.2(a). Furthermore, I have performed the Delaunay triangulation for specific types of minutiae, such as bifurcations and ridge endings. Figure 4.2(b) depicts the Delaunay triangulation for ridge ending points, which allows the distance to be calculated between one ridge ending and its nearest ridge ending neighbour. Likewise, Figure 4.2(c) depicts the triangulation plot for bifurcation points.



(a) All minutiae data (b) Ridge ending data (c) Bifurcation data

Figure 4.2: Example plot of the 2-D Delaunay triangulation for a set of minutiae points from a image. (a) Both types of minutiae points are used for this plot (bifurcations and ridge endings). (b) Only ridge ending points are used. (c) Only bifurcation points are used.

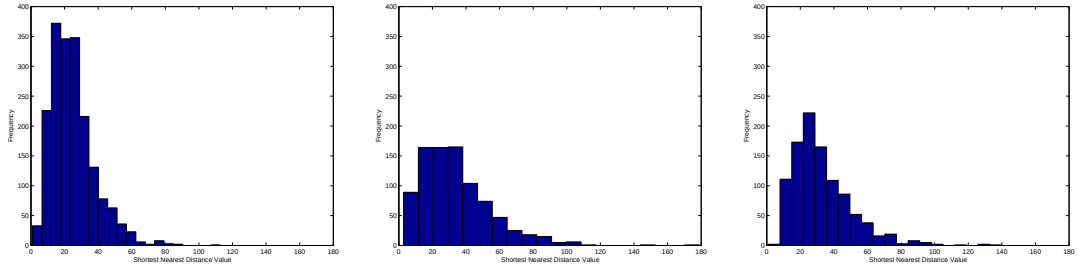
The next step is to calculate the distance between neighbouring minutiae. For each vertex point, the distance between itself and each adjoining vertex is measured. This is analogous to calculating the distances between each minutiae point and its neighbours. The smallest value from the resulting list of distance values is then chosen, which gives us the distance from the minutiae to its nearest neighbouring point.

4.2.2 Experimental results

Table 4.2 summarises the results from experiments conducted on a sample of 30 fingerprint images. The sample size figures shown in the table represent the total number of distances calculated between a minutiae point and its nearest neighbour. Additionally, the histogram plots for each of the data types from the table are shown in Figure 4.3. It can be seen that the results exhibit large standard deviation values. This suggests that the shortest distance values vary greatly for both types of minutiae, and that the configuration of a group of neighbouring minutiae is not evenly distributed throughout a fingerprint image. In addition, further work can be done to fit a probability distribution model to this set of observed data, which can provide further insight into the statistical nature of distances between neighbouring minutiae points.

Data Type	Sample Size	Mean Distance (pixels)	Standard Deviation
Minutiae Data	1894	24.9594	12.8920
Ridge Ending Data	879	34.0183	21.0203
Bifurcation Data	1015	32.6584	17.4864

Table 4.2: Summary of results for calculating the shortest distance between neighbouring minutiae.



(a) All minutiae data (b) Ridge ending data (c) Bifurcation data

Figure 4.3: Histogram plot of the shortest distance values between neighbouring minutiae. The distance values are from Table 4.2 and are measured in pixels. (a) All minutiae data is used for the plot. (b) Triangulation of only the ridge ending points. (c) Triangulation of only the bifurcation points.

4.3 Ridge wavelength

4.3.1 Methodology

The focus of this section is to gain some insight into the statistical characteristics of the ridge wavelength in a fingerprint image. The ridge wavelength data is collected by performing the ridge frequency estimation method discussed in Section 2.2.4 to a set of fingerprint images. Given that the wavelength is estimated block-wise, the ridge wavelength data for each 32×32 block of each image is recorded. Note that only the blocks that are part of the foreground region of the fingerprint image are considered. These ridge wavelength values are then used to form the results set used for statistical analysis.

4.3.2 Experimental results

Experiments were conducted to collect data on the ridge wavelength values of a set of 30 fingerprint images. A total sample size of 5510 ridge wavelength values were collected, with each sample corresponding to the ridge wavelength in each 32×32 block of each image. A mean ridge wavelength of 9.2732 pixels was observed with a standard deviation of 1.8985. The corresponding histogram plot for these results is depicted in Figure 4.4(a). From this plot, it can be seen that the peak of the histogram plot lies approximately in the middle of the histogram curve. The position of this peak point along the histogram x axis is within the ridge wavelength range of eight to nine, which is comparable to the mean ridge wavelength value.

Furthermore, the degree to which the ridge wavelength diverges away from this middle peak is fairly consistent for both sides of the peak.

This observed shape of the histogram plot suggests that the ridge wavelength data may follow a normal distribution. Hence, I have chosen to fit a normal density curve to the histogram plot as a guide to how well the data fits in with the curve. The histogram plot with a superimposed normal density curve is shown in Figure 4.4(b). The shape of the normal curve is determined by the mean and standard deviation of each data set. Visual inspection of the plot shows that although the general shape of the histogram data is similar to the normal density curve, there are outlying areas of the histogram that do not fall within the area of the normal density curve.

As a more accurate measure in testing the normality of the data, I have employed the use of normal probability plots via the MATLAB function `normplot`. I will only be discussing the basic principles of normal probability plots, further details on normal probability plots can be found in textbooks on statistics [2]. The normal probability plot for the ridge wavelength data is depicted in Figure 4.5. The plot consists of two main components: a fitted normal line which is represented by the straight dashed line, and the sample data points denoted by the + symbol.

The normality of the data can be assessed by seeing how well the plotted points fits the normal line. If the plot is linear and fits the line well, then the data can be modelled by a normal distribution. However, Figure 4.5 shows that although the start of the plot fits the line well, the rest of the plotted points are not linear and appear to bend to the right of the normal line. Hence, the observed ridge wavelength data does not quite follow a normal distribution.

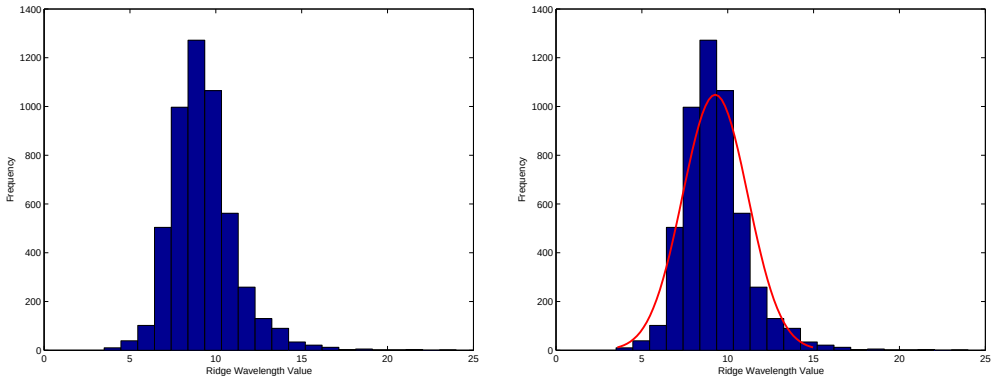


Figure 4.4: Histogram plot of the ridge wavelength values using a mean of 9.2732 with a standard deviation of 1.8985.

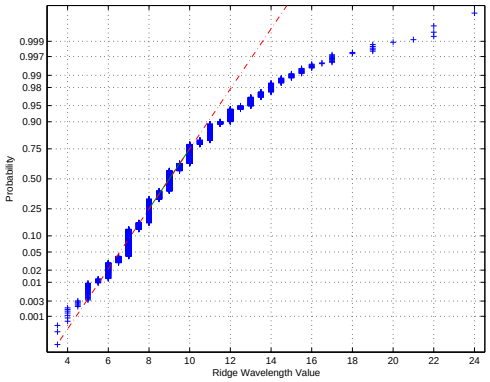


Figure 4.5: Normal probability plot of the ridge wavelength values.

CHAPTER 5

Conclusion and Future Work

The primary focus of the work in this project is on the enhancement of fingerprint images, and the subsequent extraction of minutiae. Firstly, I have implemented a series of techniques for fingerprint image enhancement to facilitate the extraction of minutiae. Experiments were then conducted using a combination of both synthetic test images and real fingerprint images in order to provide a well-balanced evaluation on the performance of the implemented algorithm. The use of synthetic images has provided a more quantitative and accurate measure of the performance. Whereas real images rely on qualitative measures of inspection, but can provide a more realistic evaluation as they provide a natural representation of fingerprint imperfections such as noise and corrupted elements.

The experimental results have shown that combined with an accurate estimation of the orientation and ridge frequency, the Gabor filter is able to effectively enhance the clarity of the ridge structures while reducing noise. In contrast, for low quality images that exhibit high intensities of noise, the filter is less effective in enhancing the image due to inaccurate estimation of the orientation and ridge frequency parameters. However, in practice, this does not pose a significant limitation as fingerprint matching techniques generally place more emphasis on the well-defined regions, and will disregard an image if it is severely corrupted. Overall, the results have shown that the implemented enhancement algorithm is a useful step to employ prior to minutiae extraction.

The Crossing Number method was then implemented to perform extraction of minutiae. Experiments conducted have shown that this method is able to accurately detect all valid bifurcations and ridge endings from the thinned image. However, there are cases where the extracted minutiae do not correspond to true minutia points. Hence, an image postprocessing stage is implemented to validate the minutiae. The experimental results from the minutiae validation algorithm indicate that this additional postprocessing stage is effective in eliminating various types of false minutiae structures.

In combination with the implemented techniques for image enhancement and minutiae extraction, preliminary experiments on the statistics of fingerprints were conducted on a sample set of fingerprint images. Three types of statistical data were collected, which include minutiae density, distance between neighbouring minutiae, and ridge wavelength. Although full analysis of the statistical data was not conducted, the results presented in this dissertation can be used as a basis for future work.

Overall, I have implemented a set of reliable techniques for fingerprint image enhancement and minutiae extraction. These techniques can then be used to facilitate the further study of the statistics of fingerprints. In addition, these techniques can be also employed in other fingerprinting applications such as fingerprint matching and classification.

Further work which can be carried out include the following:

- An investigation into a filter whose primary aim is to specifically enhance the minutia points. This project has followed the approach adopted by most previous work where the emphasis is on enhancing the ridge structures using Gabor, or Gabor-like filters. However, while the ridge structures are enhanced, this approach has shown to be less effective in enhancing areas containing minutiae points, which are the points of main interest.

- To perform the statistical experiments used in this project on a larger sample size, and to conduct a full analysis of the observed results.

- Further study into the statistical theory of fingerprint minutiae. In particular, the Tu and Hartley [26] approach can be investigated to determine the number of degrees of freedom within a fingerprint population. These results can then be used to help us better understand the statistical uniqueness of fingerprint minutiae.

Bibliography

[1] Amengual, J. C., Juan, A., Prez, J. C., Prat, F., Sez, S., and Vilar, J. M. Real-time minutiae extraction in fingerprint images. In *Proc. of the 6th Int. Conf. on Image Processing and its Applications* (July 1997), pp. 871–875.

[2] Daly, F., Hand, D. J., Jones, M. C., Lunn, A. P., and McConway, K. J. *Elements of Statistics*. Addison-Wesley, 1999, pp. 349–352.

[3] Dankmeijer, J., Waltman, J. M., and Wilde, A. G. D. Biological foundations for forensic identification based on fingerprints. *Acta Morphologica Neerlando-scandivanica 18*, 1 (1980), 67–83.

[4] Daugman, J. G. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *Journal of the Optical Society of America (A) 2*, 7 (July 1985), 1160–1169.

[5] Galton, F. *Fingerprints*. Mcmillan, 1982.

[6] Garris, M. D., Watson, C. I., McCabe, R. M., and Wilson, C. L. National Institute of Standards and Technology fingerprint database, November 2001.

[7] Guo, Z., and Hall, R. W. Parallel thinning with two-subiteration algorithms. *Communications of the ACM 32*, 3 (March 1989), 359–373.

[8] Hong, L., Wan, Y., and Jain, A. K. Fingerprint image enhancement: Algorithm and performance evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 20*, 8 (1998), 777–789.

[9] Jain, A., Hong, L., Pankanti, S., and Bolle, R. An identity authentication system using fingerprints. In *Proceedings of the IEEE* (September 1997), vol. 85, pp. 1365–1388.

[10] Jain, A. K., and Farrokhnia, F. Unsupervised texture segmentation using Gabor filters. *Pattern Recognition 24*, 12 (1991), 167–186.

[11] Jain, A. K., Hong, L., and Bolle, R. M. On-line fingerprint verification. *IEEE Transactions on Pattern Analysis and Machine Intelligence 19*, 4 (1997), 302–314.

[12] Jain, A. K., Prabhakar, S., and Hong, L. A multichannel approach to fingerprint classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence 21*, 4 (1999), 348–359.

[13] Kingston, C. R. *Probabilistic Analysis of Partial Fingerprint Patterns.* PhD thesis, University of California, Berkeley, 1964.

[14] Kovesi, P. MATLAB functions for computer vision and image analysis. School of Computer Science and Software Engineering, The University of Western Australia. http://www.cs.uwa.edu.au/∼pk/Research/MatlabFns/index.html Accessed: 20 August 2003.

[15] Maltoni, D., Maio, D., Jain, A. K., and Prabhakar, S. *Handbook of Fingerprint Recognition.* Springer, 2003.

[16] Mehtre, B. M. Fingerprint image analysis for automatic identification. *Machine Vision and Applications 6*, 2 (1993), 124–139.

[17] Prabhakar, S., Wang, J., Jain, A. K., Pankanti, S., and Bolle, R. Minutiae verification and classification for fingerprint matching. In *Proc. 15th International Conference Pattern Recognition (ICPR)* (September 2000), vol. 1, pp. 25–29.

[18] Ratha, N., Chen, S., and Jain, A. Adaptive flow orientation based feature extraction in fingerprint images. *Pattern Recognition 28*, 11 (1995), 1657–1672.

[19] Ross, A., Jain, A., and Reisman, J. A hybrid fingerprint matcher. *Pattern Recognition 36*, 7 (July 2003), 1661–1673.

[20] S. Kasaei, M. D., and Boashash, B. Fingerprint feature extraction using block-direction on reconstructed images. In *IEEE region TEN Conf., digital signal Processing applications, TENCON* (December 1997), pp. 303–306.

[21] Sherlock, D. B. G., Monro, D. M., and Millard, K. Fingerprint enhancement by directional Fourier filtering. In *IEE Proc. Vis. Image Signal Processing* (1994), vol. 141, pp. 87–94.

[22] Simon-Zorita, D., Ortega-Garcia, J., Cruz-Llanas, S., and Gonzalez-Rodriguez, J. Minutiae extraction scheme for fingerprint recognition systems. In *Proceedings of the International Conference on Image Processing* (October 2001), vol. 3, pp. 254–257.

[23] Stoney, D. A., and Thornton, J. I. A systemic study of epidermal ridge minutiae. *Journal of forensic sciences 32*, 5 (1987), 1182–1203.

[24] Tamura, H. A comparison of line thinning algorithms from digital geometry viewpoint. In *Proc. of the 4th Int. Conf. on Pattern Recognition* (1978), pp. 715–719.

[25] Tico, M., and Kuosmanen, P. An algorithm for fingerprint image postprocessing. In *Proceedings of the Thirty-Fourth Asilomar Conference on Signals, Systems and Computers* (November 2000), vol. 2, pp. 1735–1739.

[26] Tu, P., and Hartley, R. Statistical significance as an aid to system performance evaluation. In *ECCV (2) 2000* (2000), vol. 85, pp. 366–378.

[27] Xiao, Q., and Raafat, H. Fingerprint image postprocessing: a combined statistical and structural approach. *Pattern Recognition 24*, 10 (1991), 985–992.

APPENDIX A

Original research proposal

Title: Fingerprint Image Enhancement and Minutiae Extraction

Author: Raymond Thai

Supervisor: Dr Peter Kovesi

Background

Fingerprints are the oldest form of biometric identification. Modern fingerprint based identification is used in forensic science, and in biometric systems such as civilian identification devices. Despite the widespread use of fingerprints, there is little statistical theory on the uniqueness of fingerprint minutiae.

A fingerprint is formed from an impression on a surface of composite curve segments. A ridge is defined as a single curved segment, and a valley is the region between two adjacent ridges. The minutiae, which are the local discontinuities in the ridge flow pattern, provide the details of the ridge-valley structures, like ridge-endings and bifurcations. There are 50 to 150 minutiae on a single fingerprint image. Features such as the type, direction, and location of minutiae are taken into account when performing minutiae extraction [1].

The work of F. Galton [2] defined a set of Galton Features for fingerprint identification, which since then, has been refined and extended to include additional types of fingerprint features. However, most of these features are not used in automatic fingerprint identification systems. Instead the set of minutiae types are restricted into only two types, ridge endings and bifurcations, as other types of minutiae can be expressed in terms of these two feature types. Ridge endings are the points where the ridge curves terminates, and bifurcations are where a ridge splits from a single path to two paths at a Y-junction. In this project, we will be dealing mainly with ridge endings and bifurcations.

Fingerprint images are rarely of perfect quality. They may be degraded and corrupted with elements of noise due to many factors including variations in skin and impression conditions. A critical step in studying the statistics of fingerprint minutiae is to reliably extract minutiae from the input fingerprint images. In automated fingerprint identification systems, image enhancement techniques are often employed prior to minutiae extraction to obtain a more reliable estimate of minutiae locations.

There are various types of approaches proposed in literature for both image enhancement, and minutiae extraction from fingerprints. The literature on these techniques will be examined and reviewed in determining the best approach to develop for this project. In particular, the fingerprint image enhancement algorithm employed by Hong et al. [3] will be evaluated and implemented to understand how the enhancement algorithm works and how well it performs.

Once a reliable minutiae extraction technique has been implemented and tested, this can be used as the basis for statistical analysis of fingerprint minutiae. The work of Tu and Hartley [4] and Pankanti et al. [5] can be examined, in which a statistical framework for analysing system performance has been presented. Tu and Hartley define a means of forming a binary code from a set of fingerprint features and then performing a set of matching experiments on the database to estimate the number of degrees of freedom within the fingerprint population. This information can then be used to investigate the effect of varying the number of minutiae used to make an identification.

Aim

The initial aim of this project is to implement in MATLAB, a set of reliable techniques for fingerprint image enhancement and minutiae extraction. The performance of these techniques will be evaluated on a fingerprint data set obtained from the National Institute of Standards (NIST). Existing techniques (such as MINDTCT from NIST) can then be used as benchmarks for performance comparison.

In combination with these developed techniques, statistical experiments can then be performed on the fingerprint data sets. The results from these experiments can be used to help us better understand what is involved in determining the statistical uniqueness of fingerprint minutiae.

The ideal hypothesis that this project would aim to test is whether 12 minutiae points is enough for an identification (12 points has been the traditional benchmark for identification). Although it is not expected that the work in this

project will reach the stage of being able to fully test this hypothesis, we hope that the results of this project will contribute towards answering this question.

Method

A rough task breakdown for this project is as follows:

- Examine and review available literature on image enhancement and minutiae extraction techniques, including the NIST Fingerprint Image Software.
- Develop a series of image enhancement techniques to aid the minutiae extraction process.
- Develop a set of reliable techniques to extract the minutiae from fingerprint images.
- Evaluate the performance of the techniques using the fingerprint data set obtained from NIST.
- Use existing techniques as the benchmark for comparing the performance of the technique developed.
- After reliable minutiae detection techniques have been developed and tested, then statistical analysis experiments on the fingerprint data set can be performed and documented.

Software and hardware requirements

This project will require a (reasonably fast and reliable) computer running in the Linux operating system environment.
As the majority of the programming and experimentation work in this project will be done using MATLAB, a recent version of MATLAB will be required.

Bibliography

[1] U. Halici, L. C. Jain, and A. Erol. An introduction to fingerprint recognition. In, L.C. Jain, U. Halici, I. Hayashi, S.B. Lee, and S. Tsutsui, editors, *Intelligent Biometric Techniques in Fingerprint and Face Recognition,* pp. 3–34, CRC Press, Florida, 1999.

[2] F. Galton. *Finger Prints*. Mcmillan, London, 1892.

[3] Lin Hong, Yifei Wan, and Anil K. Jain. Fingerprint image enhancement: Algorithm and performance algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 20, no. 8, pp. 777–789, May 1998.

[4] Peter Tu and Richard Hartley. Statistical significance as an aid to system performance evaluation. *ECCV (2) 2000,* pp. 366–378, 2000.

[5] Sharath Pankanti, Salil Prabhakar, and Anil K. Jain. On the individuality of fingerprints. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 8, pp. 1010–1025, 2002.