Fingerprint Distortion Removal and Enhancement by Effective Use of Contextual Filtering

by

Mubeen Ghafoor
PE081002

A thesis submitted to the
Electronics Engineering Department
in partial fulfillment of the requirements for the degree of
DOCTOR OF PHILOSOPHY IN ELECTRONIC ENGINEERING

Faculty of Engineering
Mohammad Ali Jinnah University
Islamabad
Nov, 2014

Copyright © 2014 by Mubeen Ghafoor, PE081002

All rights reserved. Reproduction in whole or in part in any form requires the prior written permission of Mubeen Ghafoor or designated representative.

dedicated to my late grandparents Mr. & Mrs. Abdul Ghafoor Piracha
(May their souls rest with pleasures)

ACKNOWLEDGMENT

I would like to thank Dr. Imtiaz (my supervisor), Dr. Noman Jafri and the whole team of Vision and Pattern Recognition Group (VisPRS) for their kind support and guidance. I would also like to thank National ICT R&D fund [63], Ministry of IT and Telecom, Pakistan for funding part of this research. It is also acknowledged that the fingerprint databases from fingerprint verification competition [53], [55] are used in this research along with the open source implementation of Hong’s [30] and Chikkerur’s [31], [41] algorithm. I am thankful to my parents and family for their prayers and continuous support.

ABSTRACT

Automated personal authentication has become increasingly important in our information driven society and in this regard fingerprint based personal identification is considered to be the most effective tool. In order to ensure reliable fingerprint identification and improve fingerprint ridge structure, novel fingerprint enhancement approaches are proposed based on local adaptive contextual filtering. This dissertation presents the aims and objectives of the research along with the motivation and proposed research approaches for fingerprint enhancement mentioned below.

In the first part of this research, a twofold enhancement technique is proposed that involves processing both in frequency and spatial domain. The fingerprint image is first filtered in frequency domain and then local directional filtering in spatial domain is applied to obtain enhanced fingerprint. Two major advantages of the proposed enhancement technique are; avoiding the hazards of frequency estimation errors and making spatial domain filtering quite simple by using a global 1D Gaussian smoothing filter. In order to determine the performance evaluation of the proposed enhancement, extensive evaluation of the proposed method against well-known enhancement approaches has been carried out on publicly available standard databases. Experimental results demonstrate that proposed enhancement performs better as compared to other well-known enhancement techniques.

In the second part of this work, a fingerprint image with non-uniform ridge frequencies is considered as a 2-D dynamic signal. A non-uniform stress on the sensor’s sensing area applied during fingerprint acquisition may result in a nonlinear distortion that disturbs the local frequency of ridges, adversely affecting the matching performance. This study presents a new approach based on short time Fourier transform (STFT) analysis and local adaptive contextual filtering for frequency distortion removal and enhancement. In the proposed approach, the whole image is divided into sub-images and local dominant frequency band and orientation are estimated. Gaussian Directional band pass filtering is then adaptively applied in frequency domain. These filtered sub-images are then combined using our proposed technique to obtain the enhanced fingerprint image of high ridge quality and uniform inter-ridge distance. Experimental results show efficacy of the proposed enhancement technique.

LIST OF PUBLICATIONS

[1] **Mubeen Ghafoor,** Imtiaz A. Taj, Waqas Ahmad, M Noman Jafri, “Efficient 2-fold Contextual Filtering Approach for Fingerprint Enhancement”, IET Image Processing, Volume 8, Issue 7, p. 417 – 425, July 2014.

[2] **Mubeen Ghafoor,** Imtiaz A. Taj, M Noman Jafri, “Fingerprint Frequency Normalization and Enhancement using 2-D STFT Analysis”, submitted to IET Image processing, (under review).

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

AFIS	Automated Fingerprint Identification System
ROI	Region of Interest
FFT	Fast Fourier Transform
IFFT	Inverse Fast Fourier Transform
STFT	Short Time Fourier Transform
FT	Fourier Transform
BFS	Breath First Search
EER	Equal Error Rate
DET	Detection Error Tradeoff
ROC	Receiver Operating Characteristic
FMR	False Matching Rate
FNMR	False Non-Matching Rate
DNA	Deoxyribonucleic Acid

Chapter 1

INTRODUCTION

1.1 Overview

Biometrics are the measurable biological and behavioral characteristics of human beings and employed to authentication techniques for their identification. The use of biometrics in identifying humans goes back to prehistoric times when humans used to adorn the caves with their biometrics like fingerprints and handprints as their signatures. There are evidences that Babylonian used clay tablets with fingerprints as signatures in their business transactions as early as 500 BC. The Chinese merchants used fingerprints as signatures to settle their business deals in the Fourteenth century.

In the recent years, due to the importance of security applications and increased dependency on network based e-business and e-banking transactions, research in security related fields especially secured identification and identity verification has attracted a lot of interest. For robust authentication, human biometrics based approaches are becoming increasingly common as compared to passwords or card based systems, because the password or the card is likely to be hacked or stolen. Therefore, biometrics provide the most accurate, safe and, acceptable authentication method [1]. Biometrics market has grown considerably in the last a few years and market trends suggest that its growth is still on the rise in the coming years due to rising security concerns world-wide. In a recent report, published by USA Homeland Security News Wire [2], the current biometrics market is estimated to be around $5 billion, and it is forecast to reach around $12 billion by the year 2015.

1.2 Biometrics Traits used for Authentication

Different types of biological and behavioral biometrics are used for human authentication. Some of the examples of human biometrics, like fingerprints, Deoxyribonucleic Acid (DNA), eye retinas and irises, voice patterns, signatures, gait patterns, facial patterns and palmprints are shown in Figure 1.1. The reliability of biological biometrics is far higher than that of behavioral biometrics. That is why, biological biometrics are more commonly used in security applications. A brief description of some of the biological biometrics is given in following subsections.

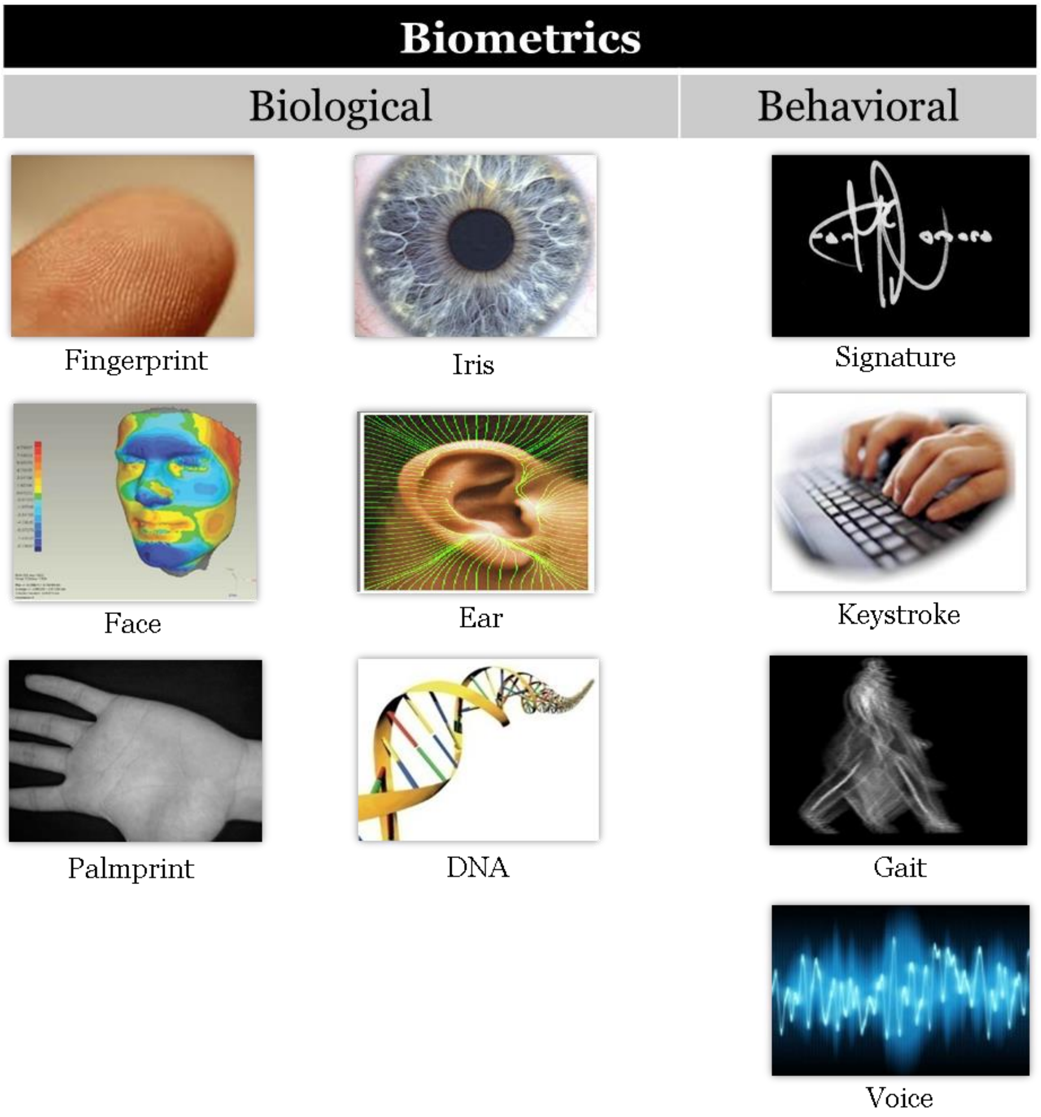
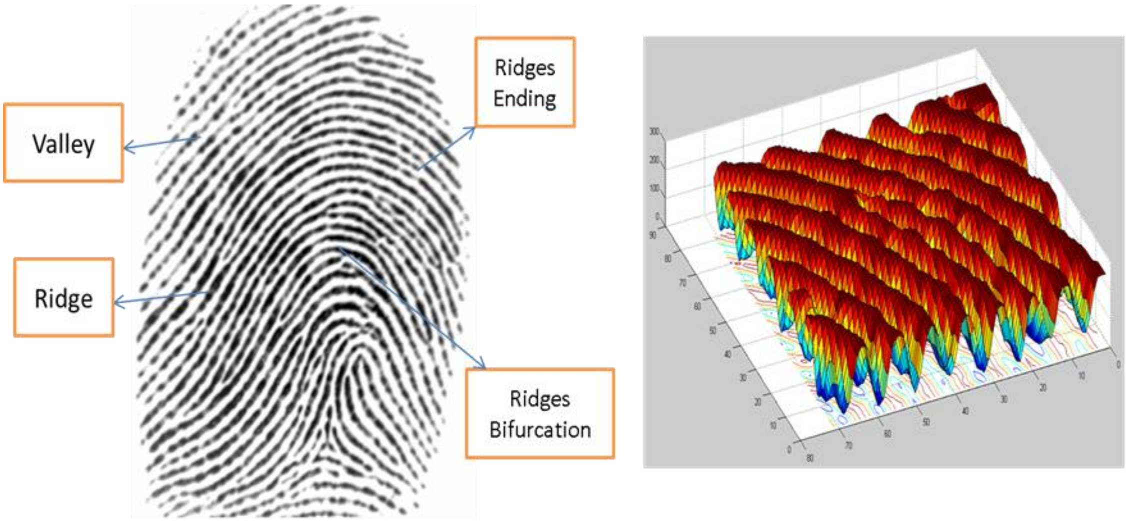


Figure 1.1: Different Biometrics used for human authentication

1.2.1 Fingerprint as Biometric

A fingerprint image can be considered as a sinusoidal wave pattern of oriented ridges having almost same frequency throughout the fingerprint. The crests and troughs of these alternating wave patterns are commonly known as ridges and valleys as shown in Figure 1.2 (a). Figure 1.2 (b) shows a zoomed 3-D plot of small portion taken from fingerprint image of Figure 1.2 (a). The ridges on the fingerprint image possess high degree of uniqueness because of certain landmarks commonly known as Minutiae.

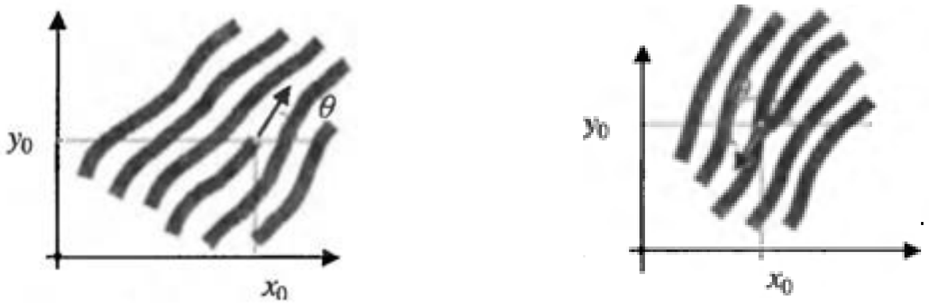
The term ‘minutia’ means small details, and in fingerprint context, it is an information that can be derived from various types of discontinuities of ridges.



(a) (b)

Figure 1.2: Fingerprint Image Structure (a) Fingerprint image, (b) Zoomed 3-D plot of small portion taken from (a)

Although there are several types of ridge discontinuities, but ridge endings (end point of a ridge) and ridge bifurcations (a ridge divided into two ridges) are considered more important because of their reliability and uniqueness. A pictorial representation of ridge ending and bifurcation is shown in Figure 1.3 [3].



(a) (b)

Figure 1.3 Common Minutia types (a) Minutia Ending. (b) Minutia Bifurcation.

1.2.2 Face as Biometric

Face is the most common biometric that human uses for authentication. The need for face based automated authentication systems is high because it can be acquired non-intrusively and then can be used for authentication in wide range of commercial and law enforcement applications. Even though face recognition algorithms reached a certain level of maturity, yet there remain major challenges that limit the use of face in real world authentication systems. These challenges include, change of illumination conditions, pose variations, facial disguise, aging affects, expressions variations, etc [4].

1.2.3 Iris as Biometric

Human iris is a complex structure that exist between the cornea and the lens of the eye in the form of circular diaphragm. It is considered as one of the most accurate biometric because of its unique patterns like furrows, freckles, crypts, and coronas and other features as shown in Figure 1.4 [5]. These features are highly consistent over the person's life span making iris a reliable choice in authentication applications.

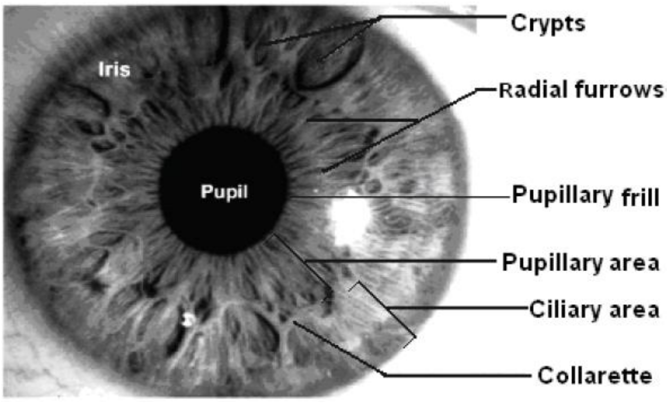


Figure 1.4 Iris Structure

Although, iris based authentication systems give high accuracy rates, yet acquiring the complex iris patterns demand good quality images that require expensive sensors. Additionally, iris image acquisition is a bit complicated and tiresome. Also user cooperation is required to ensure that the iris is captured properly from a predetermined distance from the focal plane of the camera.

1.2.4 Palmprint as Biometric

Palmprint image is acquired from the inner part of human hand. It starts from wrist and ends at roots of fingers as shown in Figure 1.5 [6]. It provides much larger area as compared to fingerprint and also possesses additional features like principal lines, wrinkles, minutia, ridges and texture etc. The principal lines consist of life line, heart line and head line.

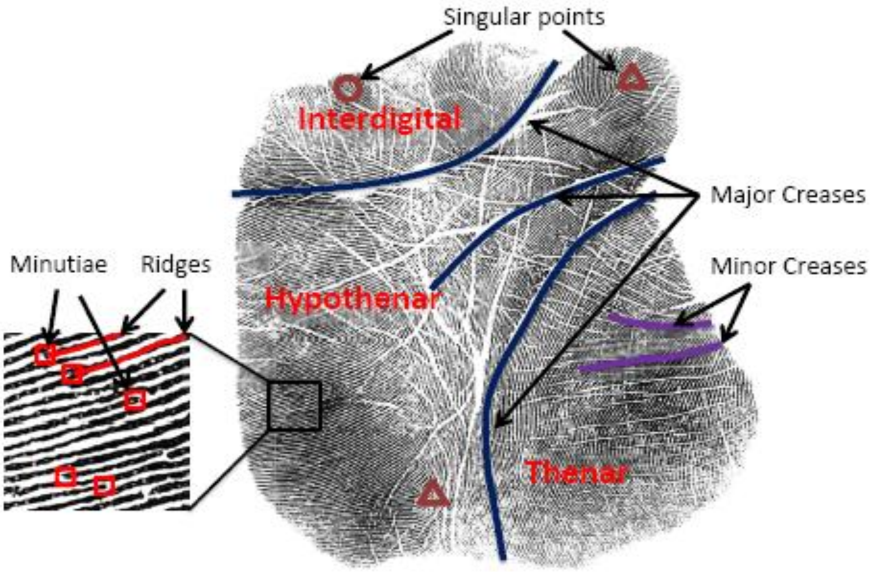
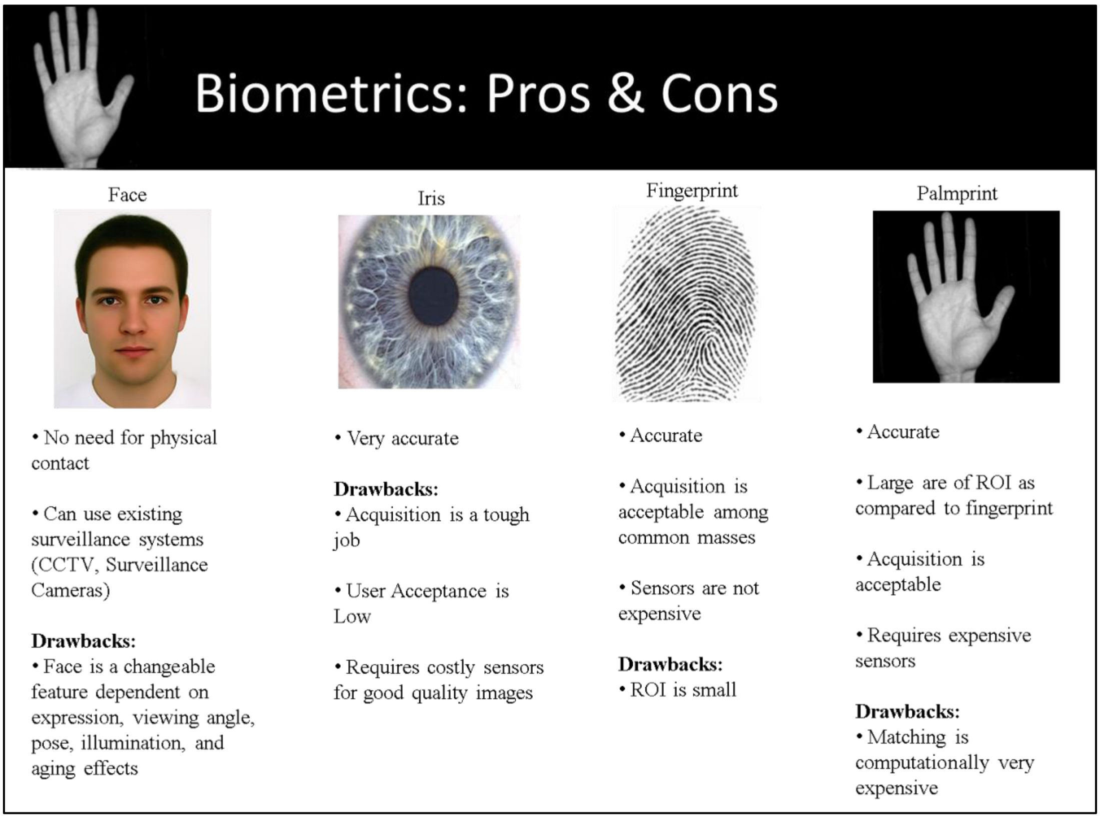


Figure 1.5: Creases, Ridges and Minutiae points in palmprint

1.3 Motivation: Fingerprint as Biometric

The assessment of biological biometrics like fingerprint, iris, face, palmprint, etc. are based on three important characteristics; uniqueness of features, stability over time and acceptability among masses. Based on these characteristics, every biometric has its own strengths and weaknesses as shown and described in Figure 1.6. For example, Face biometric is the most commonly used for authentication, but its automated authentication systems are not very reliable as face is a changeable feature dependent on expressions, viewing angle, pose, illumination conditions and aging effects [4]. Similarly, iris authentication systems are considered to be the most accurate among all the biometrics [7], yet due to their tedious acquisition process, the acceptability of these systems is very low. Palmprint is a relatively new biometric being used in human authentication systems [8] and provides much larger area as well as additional features as compared to fingerprint. However, owing to the large size of palmprint image, matching is not possible in real-time for large-scale databases [9]. For example, a well-known fingerprint matcher VeriFinger [10], that performs more than 15,000 fingerprint matches per second; gives only three matches per second for palmprint images. Fingerprint authentication, on the other hand, is reliable, fast and more acceptable among common masses. Therefore, it remains the most suitable biometric for large scale national ID, civil identification and security programs [3]. According to a survey [2], automated fingerprint authentication system (AFIS) dominates other biometrics having shares of around 33% of overall biometrics market. This provides an incentive of research and development in fingerprint based authentication systems in the coming years.



Biometrics: Pros & Cons

Face

- No need for physical contact
- Can use existing surveillance systems (CCTV, Surveillance Cameras)

Drawbacks:

- Face is a changeable feature dependent on expression, viewing angle, pose, illumination, and aging effects

Iris

- Very accurate

Drawbacks:

- Acquisition is a tough job
- User Acceptance is Low
- Requires costly sensors for good quality images

Fingerprint

- Accurate
- Acquisition is acceptable among common masses
- Sensors are not expensive

Drawbacks:

- ROI is small

Palmprint

- Accurate
- Large are of ROI as compared to fingerprint
- Acquisition is acceptable
- Requires expensive sensors

Drawbacks:

- Matching is computationally very expensive

Figure 1.6: Human authentication based on different biometrics: Pros and Cons

1.4 Main Modules of Fingerprint Authentication System

Like other biometrics, fingerprint can be used as a metric for personal verification and /or identification. Fingerprint verification involves 1-1 match because a query (probe) fingerprint image with a given identity is matched against a template stored in the database against the same identity. Whereas, fingerprint identification involves 1-N matches because a query (probe) fingerprint image is matched against all the template images stored in a database in order to find the identity of the person. The overall block diagram of fingerprint authentication is shown in Figure 1.7. It involves image acquisition, pre-processing and enhancement, feature extraction, feature encoding and matching stages.

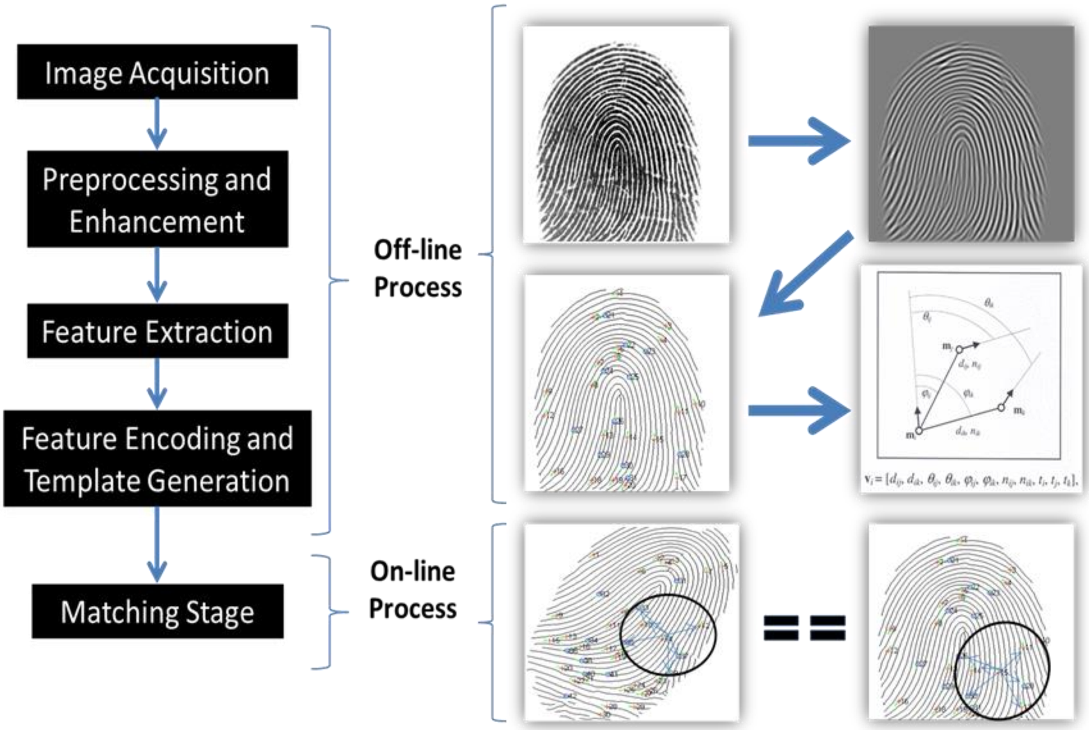


Figure 1.7: Block diagram of Fingerprint Authentication System

During the enrollment process which involves registration of a person in a database, all the initial processes up to the template generation stage are performed, and once the template is generated, it is stored in the database. During authentication all these initial processes are followed for only query image and are not required for templates from database. Therefore, matching is considered to be an on-line process and enrollment is considered to be an off-line process. If there is no wear and tear in the fingerprints, it will remain stable throughout the life time. However, due to wear and tear, the templates are generally considered stable over 5 to 10 years and enrollment is performed again after this period.

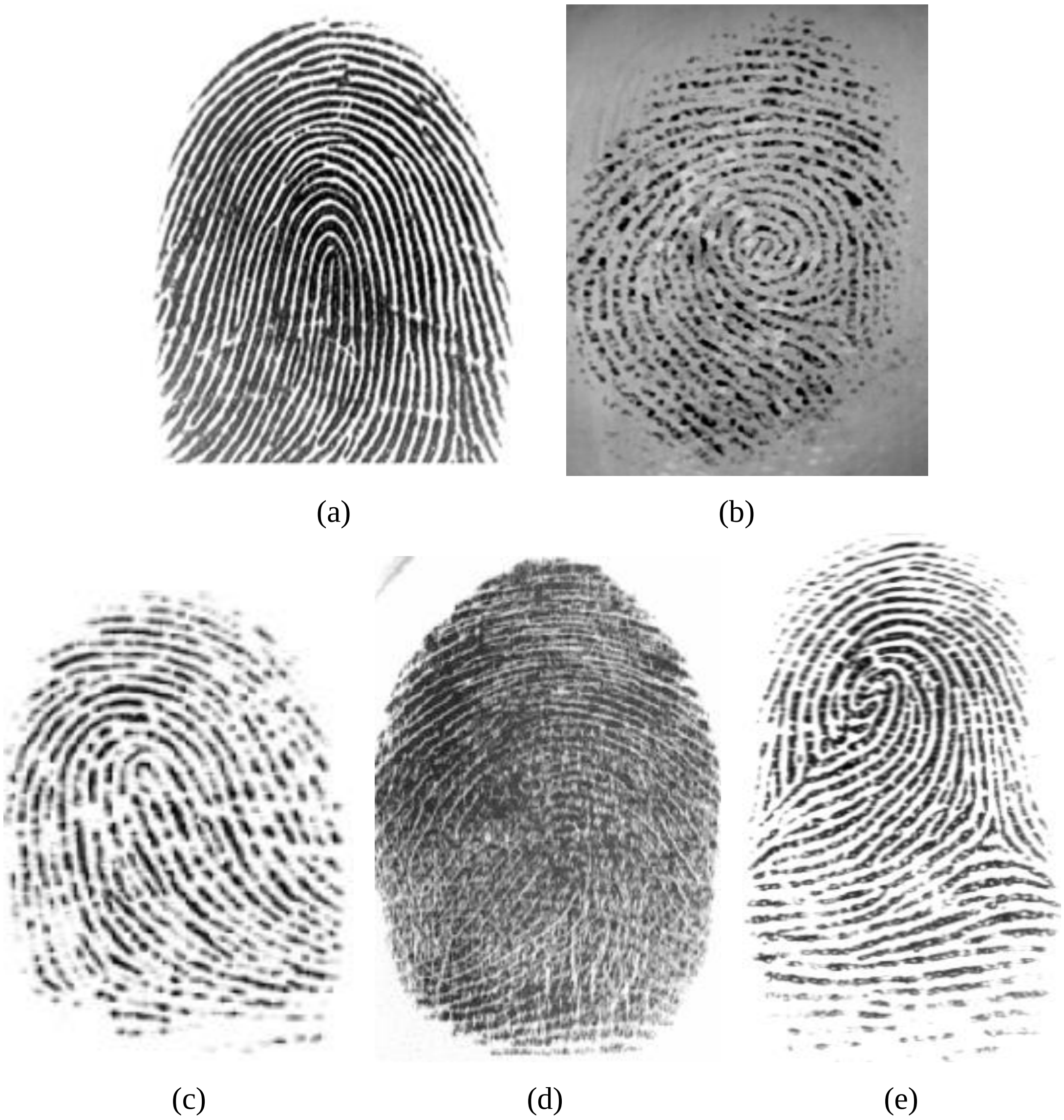
The fingerprint template is an abstract representation of specific fingerprint features. For fingerprint authentication systems the most reliable features are considered to be minutiae points which generally consist of ridge ending and bifurcation [11], explained above. The minutiae based matching methods are very sensitive to the occurrence and quality of minutiae, as any missing minutia or existence of false minutia can degrade the performance and error rate of the matching method [12], [13].

1.5 Statement of Problem

In order to overcome the problem of unreliable features extraction, a pre-processing and enhancement technique is imperative to fingerprint recognition system. The enhancement technique is usually designed to achieve the best results by removing different imperfections contained by fingerprint images. These imperfections can exist in the form of non-uniform illumination due to sweating effects on fingertips or imperfect sensors, scars and broken ridges, and non-uniform ridge separations, which occur due to non-uniform contact with the sensor. Figure 1.8 shows an example of different fingerprint images with varying quality.

Figure 1.8 (a) shows a good quality fingerprint image with high contrast between the ridges and valleys. Figure 1.8 (b) & (c) shows fingerprint images with low contrast and having broken ridges and scars. Figure 1.8 (d) shows a low quality fingerprint with insufficient distinction between ridges and valleys. Figure 1.8 (e) shows a fingerprint having non-uniform inter-ridge separation, which occurs due to non-uniform contact with sensor [14], [3].

In case, minutiae are directly extracted from distorted images there will be many spurious minutiae, resulting in false matching as well as increasing the matching time. In order to overcome this problem, researchers commonly employ the application of directional contextual filtering based enhancement on fingerprint image before the feature extraction stage.



(a) (b)

(c) (d) (e)

Figure 1.8: An example of different fingerprint images with varying quality. (a) Good quality image, (b) & (c) Fingerprint with low contrast and broken ridges, (d) Low quality fingerprint image with insufficient distinction between ridges and valleys, (e) Fingerprint having non-uniform inter-ridge separation

1.6 Purpose of the research

Although fingerprint recognition is a mature field of research and the algorithms for image enhancement, feature point extraction and matching are well established, in case of severely distorted images still there is a lot of room of improvement. This research focuses on robust enhancement techniques which can overcome problems in severely distorted images. The main tasks involved in this research are as follows:

- To review the different stages i.e. preprocessing, enhancement, feature extraction and matching techniques involved in fingerprint authentication systems and then review how they contribute in the overall matching results of the authentication system.
- To review the different distortions in the fingerprint image and also review the different enhancement algorithms to overcome these distortions.
- To perform the distortion removal from fingerprint images. Two studies are presented in this thesis.

  1 First study overcomes the problems like

    - Non-uniform illumination
    - Broken ridges and scars
    - Distorted ridge valley structures (low contrast ridge valley structure)

  In this study, a novel two-fold contextual filtering based enhancement scheme is proposed [15]. In order to determine the performance of the proposed algorithms, a comparative analysis is performed with major recognition schemes by calculating the equal error rates on standard fingerprint databases.

  2 Second study addresses the problem of non-uniform inter-ridge separation along with other imperfection [16].

    - Although contextual filtering based approaches work well to enhance the fingerprints, but they do not rectify the non-uniform inter-ridge separation.
    - In this research a novel enhancement technique is presented that simultaneously normalize non-uniform inter-ridge frequencies and also remove other distortions from fingerprint image.

1.7 Theoretical bases and Organization

This dissertation is organized into five chapters whose contents are described below.

Chapter 1 gives the problem statement and the organization of dissertation.

Chapter 2 presents the literature survey on the different algorithm steps involved in fingerprint enhancement, encoding and matching. A detailed literature survey on some of the most prominent contextual filtering based enhancement techniques is also presented.

Chapter 3 describes a novel fingerprint enhancement technique based on local adaptive contextual filtering. Proposed technique utilizes processing both in frequency and spatial domains to enhance a fingerprint image. This chapter also includes the frequency response of the proposed filters with other contextual filters proposed in literature, followed by the comparative analysis of proposed enhancement scheme with other major enhancement schemes based on contextual filtering.

Chapter 4 describes a novel contextual filtering based enhancement technique to simultaneously address the problems of non-uniform inter-ridge distortion along with other imperfection in the fingerprint image.

Chapter 5 gives the conclusion of the research work and suggest future research.

1.5 Summary

This chapter gives overall introduction of this dissertation. An overview of different biometrics used worldwide is presented and then motivation of fingerprint as a most reliable and acceptable biometric is presented, because of its uniqueness and stability over time. In the end, a brief overview of the purpose and objectives to be achieved in this research are presented.

Chapter 2

LITERATURE SURVEY

Fingerprint is the most mature biometric to be used for human authentication with applications ranging from security to commerce and banking. Although, fingerprint based authentication is an established field of research, yet research is still going on due to stringent time and reliability requirements for large scale applications. This dissertation focuses on the importance of fingerprint enhancement and its significance in making the fingerprint based authentication system more robust. This chapter discusses the overall fingerprint based authentication system and the details regarding different algorithmic for blocks in the system.

The overall fingerprint authentication system can be broadly divided into two phases; fingerprint encoding and matching. In encoding, a fingerprint image is initially acquired from a sensor, followed by a feature extraction stage. In feature extraction phase, features like minutiae are extracted from fingerprint image. Various techniques have been proposed for automatic extraction of minutiae features from a fingerprint image. Maio and Maltoni extracted minutiae points directly from gray images based on a ridge line following algorithm [17]. This study shows acceptable results, however, owing to high computational cost this technique is not readily used for minutiae extraction [3]. Watson et al. [18], working at National Institute of Standards and Technology (NIST), developed MINDTCT, an open source software for minutiae extraction. MINDTCT binarizes a fingerprint image by assigning a binary value to every pixel based on its neighborhood in direction of ridges and minutiae points are then extracted based on minutia searching pattern. The minutiae detection using this method is very simple and fast. However, these direct methods usually fail because fingerprint images contain some imperfections like non-uniform gray levels along the ridges, scars, creases and broken ridges. These imperfections may cause false and inconsistent detection of features especially minutiae points, which may lead to false matching.

The most consistent and reliable techniques for minutiae feature extraction are based primarily on incorporating a pre-processing and enhancement stage prior to feature extraction step. The pre-processing and enhancement stage is usually designed to achieve best results by removing different imperfections contained in fingerprint images. Feature extraction step is followed by feature encoding phase, where a biometric code of the fingerprint is calculated and stored in database as a reference template for that fingerprint. In fingerprint matching stage, the templates (biometric codes) of fingerprint images are loaded from the database and the features from these templates are then matched.

Figure 2.1 shows an overall block diagram of steps involved in fingerprint authentication. A detailed review of these steps involved in fingerprint authentication e.g. fingerprint acquisition, preprocessing, encoding and matching techniques are presented in following subsections.

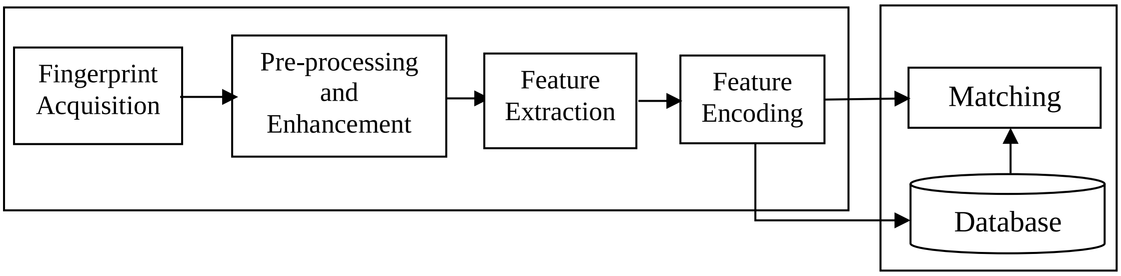


Figure 2.1: Block Diagram of Fingerprint Authentication

2.1 Fingerprint Acquisition

For forensic and criminal investigation, where fingerprint image can be partial (latent), experts rely on minutiae and ridge features, as these features are even accepted in the court of law. Also, in case of automated fingerprint matching, minutia based matching algorithms are considered to be the most accurate and robust. In order to reliably extract minutiae points, fingerprint images must be acquired at a minimum resolution of 400 dpi. Most of the fingerprint sensors readily available in the market are of 500 dpi and can be easily integrated as off the shelf devices. For research purpose Fingerprint Verification Competition (FVC) On-Going [19], conducts evaluation of latent fingerprint algorithms. They use different sensor technology for database collection and the databases are publically available for research purpose.

2.2 Pre-processing and Enhancement

The performance of feature extraction and matching algorithms depends heavily on quality of input fingerprint images. In order to ensure reliable feature extraction, fingerprint images are passed through a series of pre-processing and enhancement steps as shown in Figure 2.2. The main modules included in pre-processing and enhancement are; ROI (Region of Interest) mask segmentation, enhancement followed by binarizing and thinning. In ROI segmentation, the fingerprint region is extracted from background based on local variance and directional coherence of fingerprint image. After segmentation, fingerprint image is enhanced using different general purpose image processing techniques [20], [21], [22]. These enhancement techniques are employed to enhance the overall contrast of the fingerprint image. However, these techniques can not improve the ridge structure that is damaged mainly because of scars and broken ridges and creases.

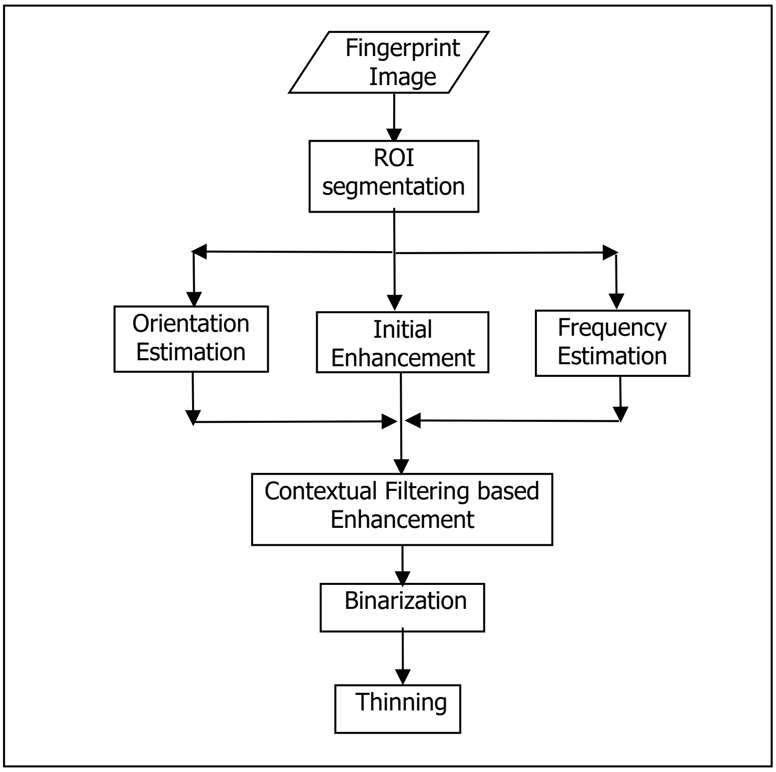


Figure 2.2: Block diagram of pre-processing and enhancement

In order to improve the ridge structure and solve the problem of broken ridges, creases and scars, fingerprint images are generally enhanced using contextual filtering. In enhancement based on contextual filtering, filters are designed using local orientation and frequency of ridges as parameters. For this purpose, ridge orientation and frequency is estimated in valid fingerprint region. Ridge structure quality is then enhanced by using contextual filtering based enhancement technique.

Minutia features are reliably extracted from thinned (skeletonized) image. That is why in the last stage of preprocessing and enhancement, the enhanced image is first binarized followed by a thinning process.

2.2.1 ROI Segmentation

Region of interest (ROI) segmentation is performed to extract fingerprint region and remove background region and other irrelevant information from the fingerprint image. This stage is also referred to as foreground/background detection. ROI extraction is necessary, so that only valid features are extracted during feature extraction stage. In ROI segmentation, the non-overlapping block-wise approaches are generally used for fast computation and easy decision making. Maio and Maltoni [17], proposed a gradient magnitude based method for segmentation of low contrast fingerprint images. In this approach, variance of local intensity is first calculated and then compared against a threshold defined by the global intensity variance of the image. If the local variance is less than the global threshold, then block is classified as a background region, otherwise, a foreground region. The gray level variance for a block of size *T*x*T* is defined as:

$$V(k) = \frac{1}{T^2} \sum_{i=0}^{T-1} \sum_{j=0}^{T-1} \left(G(i, j) - M(i, j) \right)^2, \tag{2.1}$$

where $V(k)$ is the variance for block k, $G(i, j)$ is the gray level value at pixel (i, j) and $M(i, j)$ is the mean gray level value for the block k. The experimental results obtained after applying variance based segmentation are shown in Figure 2.3. In another study, Bazen et al [23], proposed ROI segmentation based on coherence of ridge direction.

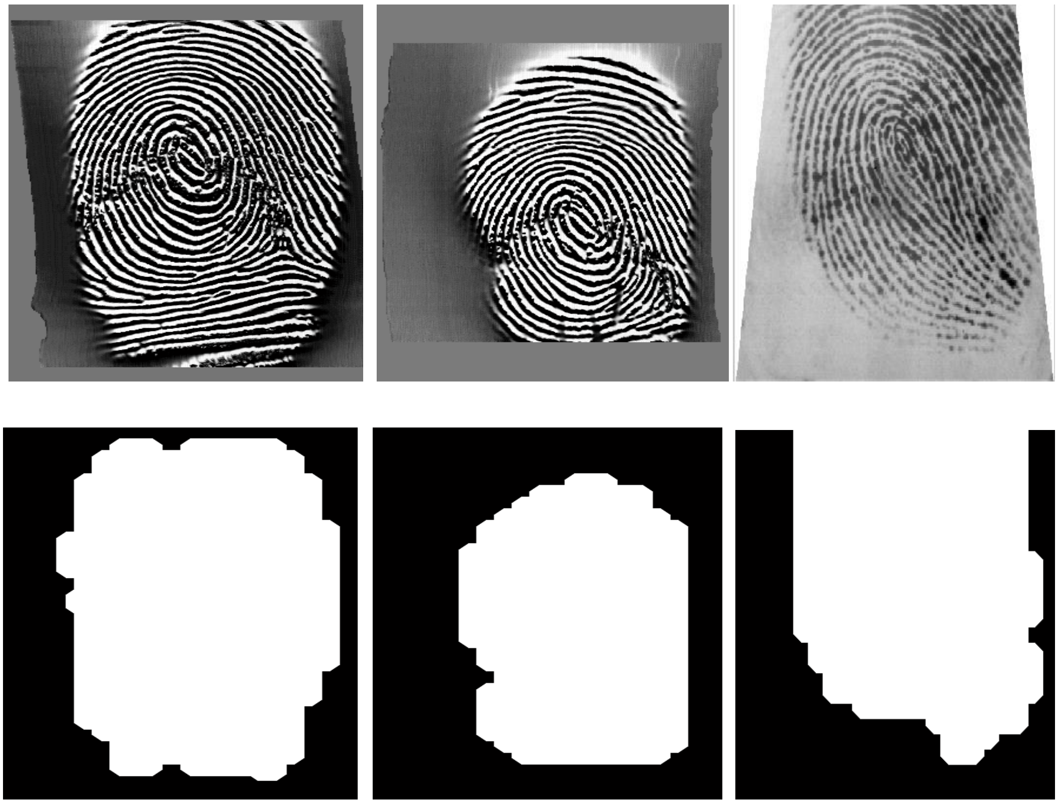


Figure 2.3: Images segmented based on intensity variance based method

2.2.2 Initial Enhancement

Fingerprint images usually contain some imperfections like non-uniform ink intensity due to sweating effects, humidity and sensor defects. This results in distorted non-uniform gray levels along the ridges and valleys patterns. Removal of non-uniform illumination and unwanted additive noise has been tackled in initial preprocessing steps of mean and variance normalization by Kim et al. [20]. Normalization basically standardizes the gray levels adjustment, so that the processed image can have desired range of mean and variance values over the whole fingerprint image. Let *g*(*i, j*) represents the gray level value at pixel (*i, j*) and *I*(*i, j*) is the normalized gray level value at pixel (*i, j*). Mathematically,

$$\begin{aligned} I(i,j) &= \left\{ M_0 + \sqrt{{}^{V_o}\!/_{V}\left(g(i,j)-M\right)^2} \right\}, \quad if\ g(i,j) < M \\ I(i,j) &= \left\{ M_0 - \sqrt{{}^{V_o}\!/_{V}\left(g(i,j)-M\right)^2} \right\}, \quad otherwise \end{aligned} \tag{2.2}$$

where M and V are estimated mean and variance of $g(i,j)$ and M_0 and V_0 are the desired mean and variance values. The normalization is usually carried out globally. However, local normalization can be used to remove slow intensity range variations due to non-uniform contact of finger with the acquisition device [20]. The experimental results obtained after applying normalization processing as discussed above are shown in Figure 2.4.

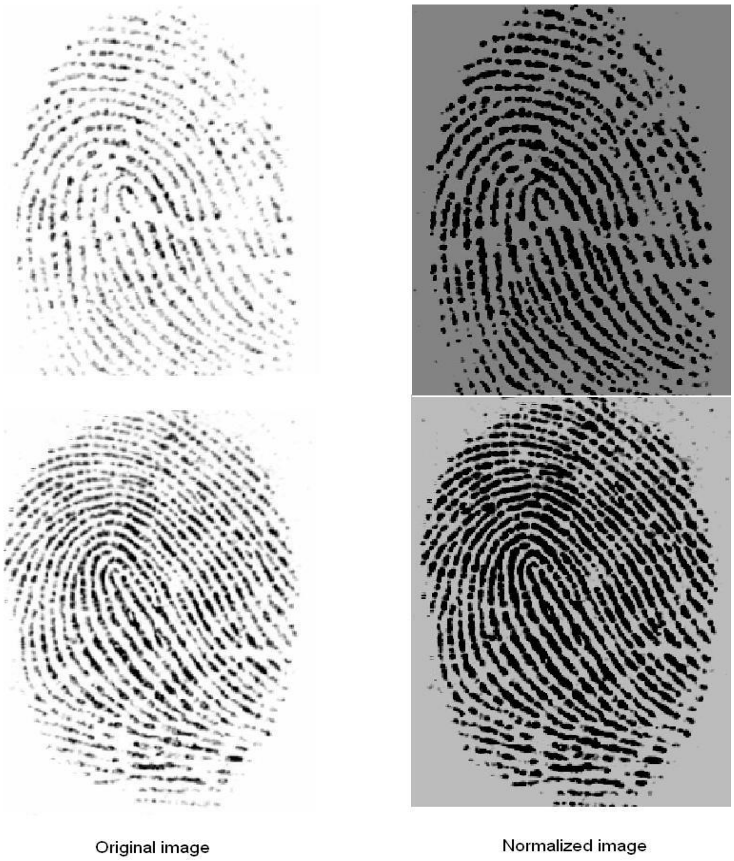


Figure 2.4: Normalized images

In another interesting approach, Greenberg et al. [21] proposed Wiener filtering for fingerprint enhancement. Willis et al [22], employed a simple frequency domain enhancement technique for removal of illumination changes and unwanted scars in fingerprint in one step. In this approach, the enhanced image is calculated by multiplying Fourier transform of a block by the k^{th} power of power spectrum and then computing its inverse Fourier transform. Chandra et al. [24] proposed a computationally efficient enhancement scheme based on median filtering.

These general purpose enhancement techniques work well to enhance overall contrast and remove noise from fingerprint image. However, these techniques cannot improve the ridge structure that is damaged mainly because of scars and broken ridges. In order to improve the ridge structure and solve the problem of broken ridges, creases and scars, fingerprint images are generally enhanced using contextual filtering. For this purpose, ridge orientation and frequency images are estimated before applying filtering.

2.2.3 Orientation Estimation

In orientations estimation, an orientation map or image $O(x, y)$ is constructed that provides the dominant direction of underlying local ridge structures [3]. This orientation image $O(x, y)$ gives dominant ridge orientation $\theta_{i,j}$ of each pixel (x, y) with the horizontal axis as shown in Figure 2.5 [3].

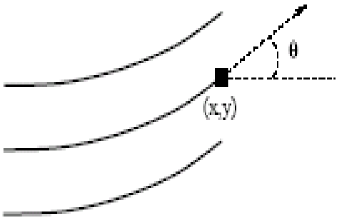


Figure 2.5: The angle of orientation

Reliable orientation estimation is very important as enhancement is dependent on it. Furthermore, orientation image is also used in feature extraction, and feature encoding. Turroni et al. [29], proposed an adaptive orientation estimation algorithm for low quality fingerprint images containing many creases and scars. The most common methods for orientation estimation are generally based on computing gradients in spatial domain [25]. However, simple gradient based methods leads to problems due to the non-linearity and discontinuity of sines and cosines. For fingerprint images, various methods are proposed to overcome this problem. Bazen et al. [28], computed orientation based on principal component analysis of the gradient vectors. Kass and Witkin [26] proposed orientation estimation based on doubling the angles. Ratha et al. [27], estimated the dominant ridge orientations $\theta_{i,j}$ by calculating multiple gradient estimates within a *17* x *17* window centered at (x_i, y_i) as shown in following equations [27].

$$\theta_{ij} = 90 + \tan^{-1}\theta\left(\frac{2G_{xy}}{2G_{xx} - 2G_{yy}}\right), \tag{2.3}$$

$$G_{xy} = \sum_{h=-8}^{8}\sum_{k=-8}^{8}\nabla_x\left(x_i + h, y_j + k\right)\nabla_y\left(x_i + h, y_j + k\right), \tag{2.4}$$

$$G_{xx} = \sum_{h=-8}^{8}\sum_{k=-8}^{8}\nabla_x\left(x_i + h, y_j + k\right)^2, \tag{2.5}$$

$$G_{yy} = \sum_{h=-8}^{8}\sum_{k=-8}^{8}\nabla_y\left(x_i + h, y_j + k\right)^2. \tag{2.6}$$

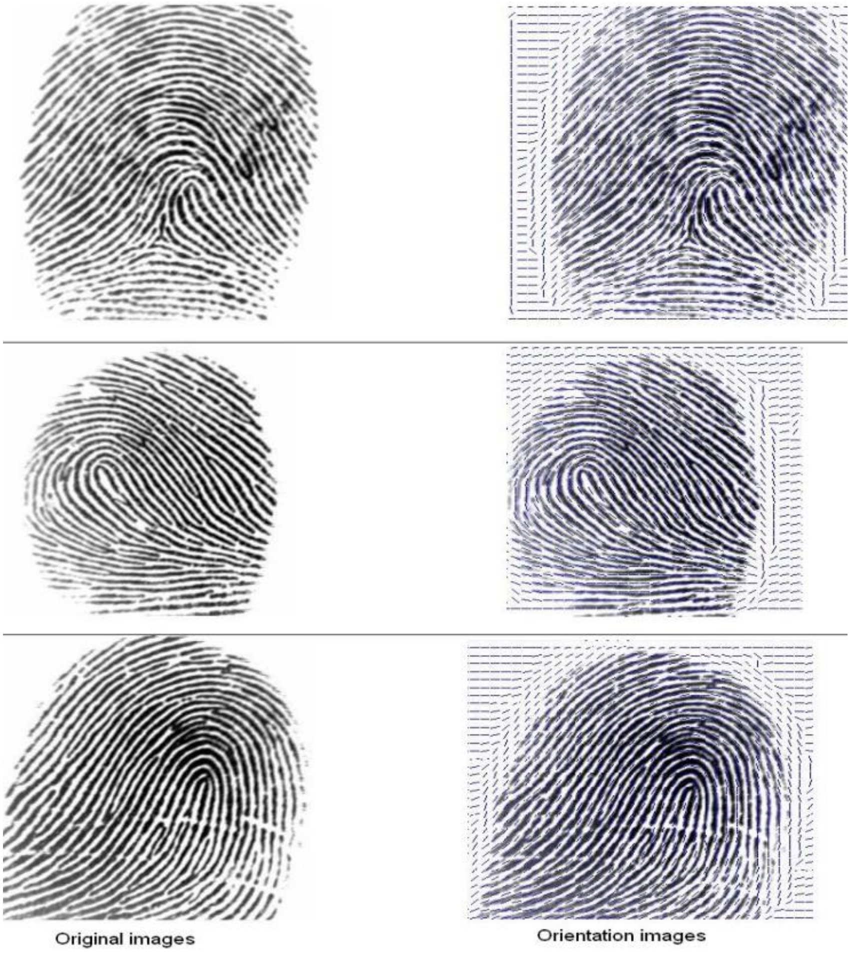


Figure 2.6: Vector fields depicting calculated orientations

The experimental results of gradient based ridge orientation calculation are shown in Figure 2.6. These orientation estimation algorithms work well for fingerprint images and can reliably detect dominant local ridge orientations.

2.2.4 Ridge Frequency Estimation

In frequency estimation, a ridge frequency image or frequency map is constructed that provides ridge density in a local region. Hong et al. [30], estimated frequency in spatial domain by x-signature of ridges within an oriented window as shown in Figure 2.7. In this approach, the fingerprint image is divided into small blocks and then for each block frequency is calculated by counting ridges within an oriented window and then dividing it by the distance of the first and last ridge within that oriented window. This proposed approach is straightforward and efficient. However, for noisy fingerprint images the frequency estimation is not accurate in spatial domain. Chikkerur et al. [31], proposed Short Time Fourier Transform (STFT) analysis for simultaneous estimation of frequency and orientation of fingerprint image.

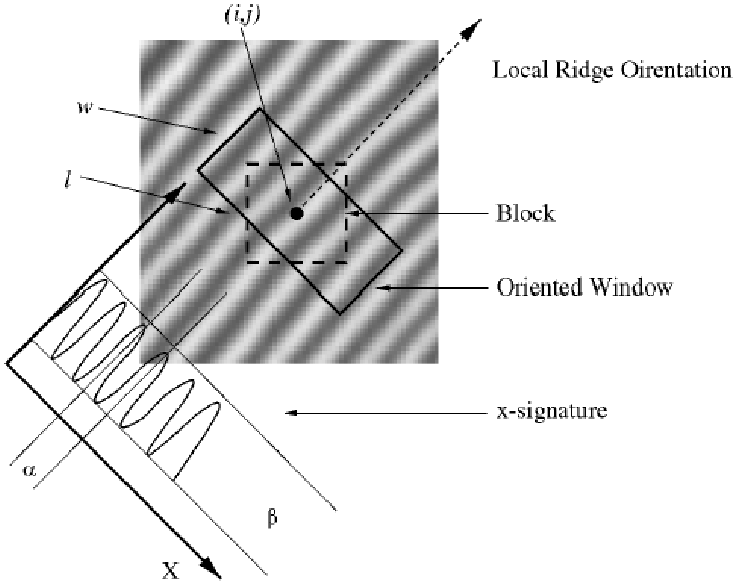


Figure 2.7: Frequency estimation using x-signature

2.2.5 Contextual Filtering based Enhancement

In image enhancement based on contextual filtering, the filter parameters are tuned adaptively during filtering process based on some local characteristic of the image contents. In fingerprint enhancement, since ridge flow patterns are to be enhanced, the filters are, therefore, designed to enhance ridge flow structure by tuning onto local ridge frequency and direction [32]. Although, the techniques of enhancement are different, yet their purpose is same. i.e. perform filtering based on ridge pattern, so that differentiation between ridge valley structure can be enhanced as compared to noise and other artifacts and also fill the broken ridge linkages by applying filtering along ridge direction.

In this section, a detailed literature survey on current state-of-the-art contextual filtering based fingerprint enhancement techniques both in spatial and frequency domains is presented. Although, a lot of algorithms can be found in literature, the most prominent ones are discussed. In order to enhance fingerprint image, O’Gorman and Nickerson [46] were the first to introduce contextual filters. The proposed enhancement technique was based on convolution of each pixel in fingerprint image with a filter. The filter is bell shaped, elongated along the ridge direction and designed for a constant ridge frequency as shown in Figure 2.8 [46].

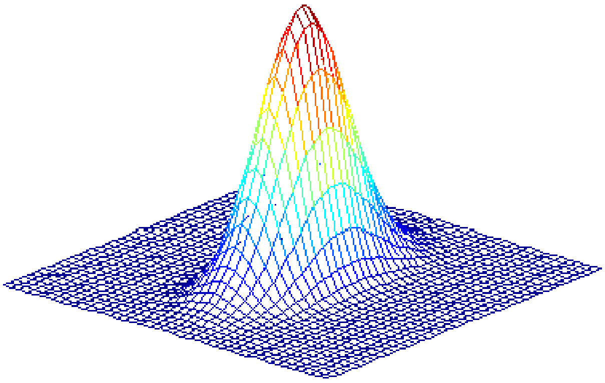


Figure 2.8: Contextual Filter designed by O’Gorman and Nickerson

A set of 16 rotated filters are derived from the proposed filter in order to cater for the different ridge orientations in fingerprint image. Convolution of each pixel is performed using a filter whose orientation matches the orientation of that pixel. Since this filter is designed for a constant ridge frequency, so it does not exploit the sinusoidal wave pattern of fingerprint. In frequency domain, the first most notable contextual filtering technique for fingerprint enhancement is proposed by Sherlock et al. [47], [48] where enhancement is achieved by designing directional filter bank based on the dominant frequency band. In proposed study, the fingerprint image is converted into frequency domain through fast Fourier transform (FFT). The dominant band in the image is then estimated in the frequency domain, which is then used to design directional filter banks suitable for enhancing the required details and suppress the unnecessary or noisy details. The directional filter bank in frequency domain is defined by multiplication of two filters as given by equation (2.7) [47]:

$$H(\rho,\theta) = H_{radial}(\rho)H_{angle}\ (\theta), \tag{2.7}$$

where H_{radial} is the band-pass filter and is dependent on the local ridge frequency ρ (equation (2.8)) and H_{angle} is directional filter and is depends on the local ridge orientation θ (equation (2.9)). Mathematically,

$$H_{radial} = \sqrt{\left[\frac{(\rho\rho_{BW})^{2n}}{(\rho\rho_{BW})^{2n} + (\rho^2 - \rho_0^2)^{2n}}\right]}, \tag{2.8}$$

where ρ_{BW} and ρ_o are the desired bandwidth and center frequency.

$$H_{angle} = \begin{cases} \cos^2 \dfrac{\pi}{2} \dfrac{(\varphi - \varphi_c)}{\varphi_{BW}} & if\ |\varphi| < \varphi_{BW} \\ 0 & otherwise \end{cases} , \tag{2.9}$$

where φ_{BW} is angular bandwidth of the filter and φ_c is the orientation at which value of filter is maximum. Each directional filter H is then multiplied by the Fourier transform of fingerprint image F to obtain a bank of directionally filtered images PF_{i}. After filtering in frequency domain all the filtered images are brought to spatial domain PI_{i} through inverse Fourier transform (IFFT) of each image. The final enhanced image I_{enh} is constructed by selecting pixel values from the bank of directionally filtered images PI_{i} based on the local orientation of each pixel. The algorithmic flow diagram for filter-bank based image enhancement is shown in Figure

2.9. Some of the intermediary results produced by Sherlocks's algorithm [48] are shown in Figure 2.10. However, this scheme works well only if frequency (inter-ridge separation) is constant throughout fingerprint image, as filter bank is designed based on dominant frequency band.

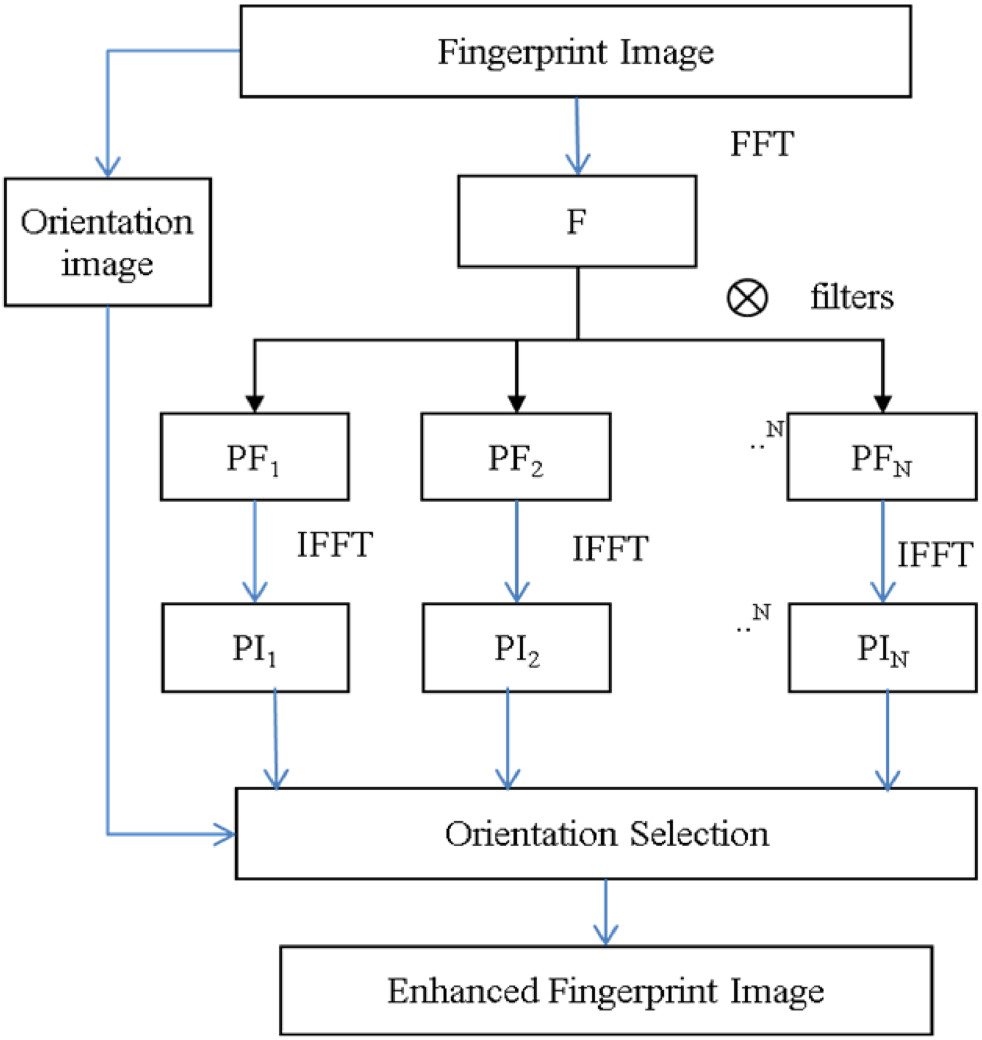
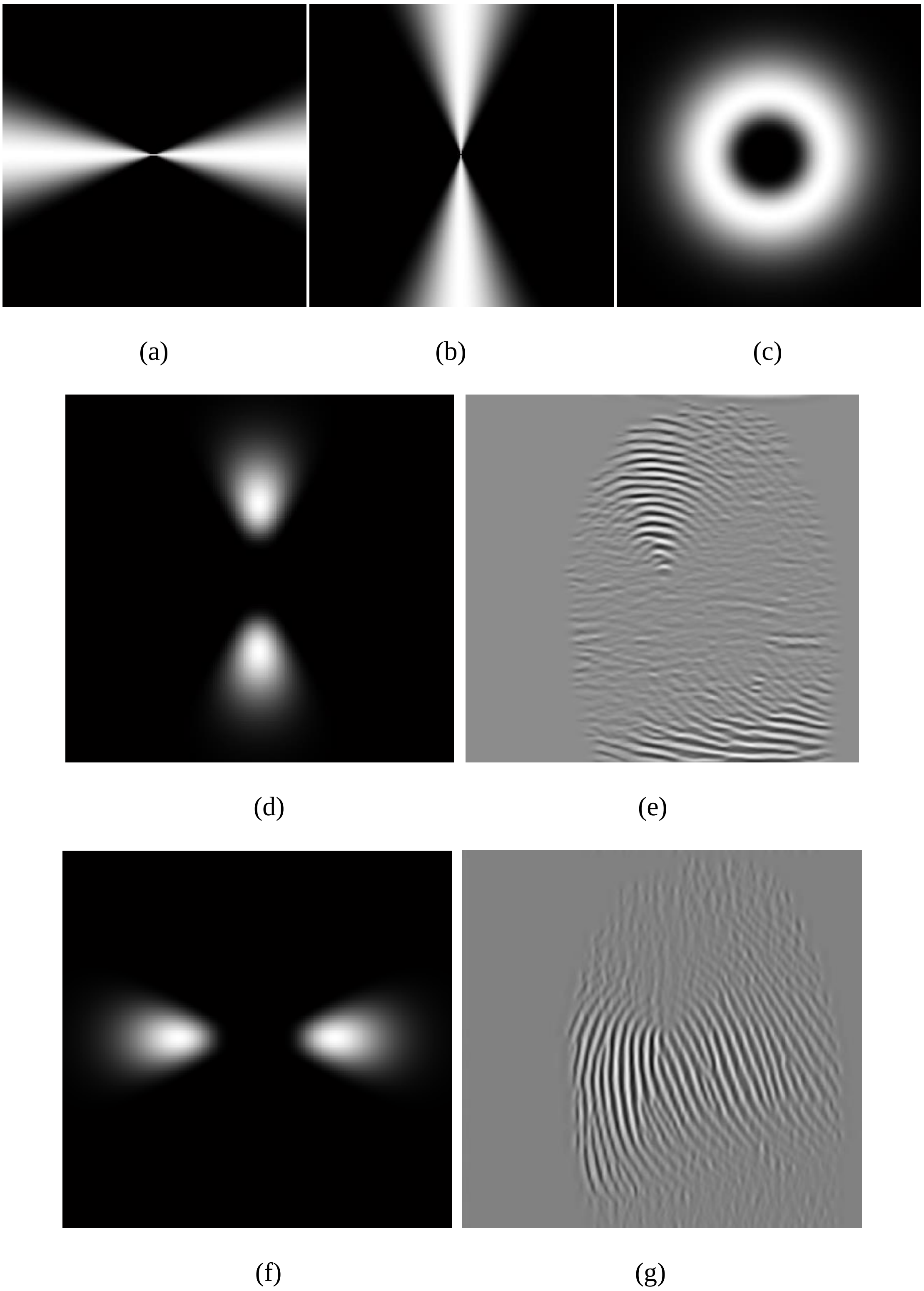


Figure 2.9: Algorithmic flow diagram for filter-bank based image enhancement proposed by Sherlock et al. [48]



(a) (b) (c)

(d) (e)

(f) (g)

Figure 2.10: Some intermediary results of Sherlock et al. [48] proposed enhancement scheme

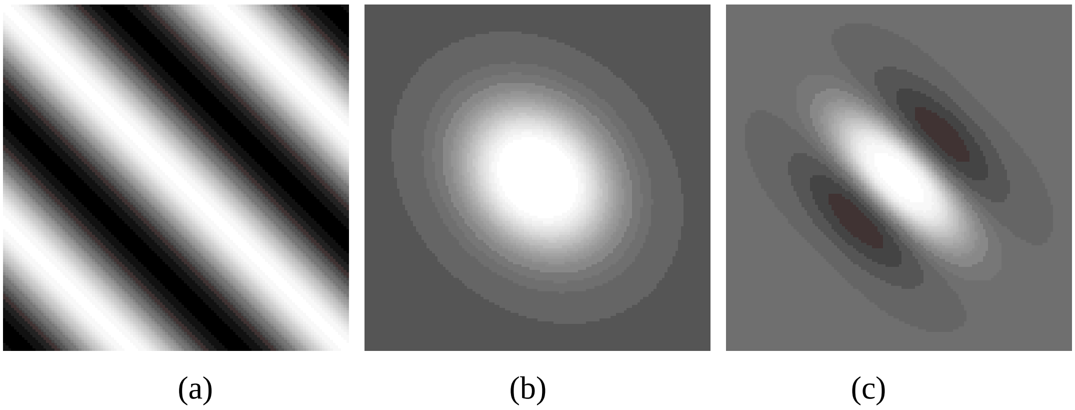
Hong et al. [30], proposed spatial domain based fingerprint enhancement technique using Gabor filters because of their frequency and orientation selective properties. The enhancement procedure using Gabor filter is straightforward and is implemented by convolving each pixel of the fingerprint image with a Gabor filter properly tuned based on local frequency and orientation estimates. A Gabor filter can be represented as a sinusoidal wave of specific frequency and orientation, modulated by a Gaussian envelope. However, for fingerprint enhancement, the real part of 2-D Gabor filter having frequency *f* and oriented at an angle *θ* is used given by equation (2.10).

$$g(x,y:\theta,f)=\exp\left[-\frac{1}{2}\left(\frac{x_{\theta}{}^{2}}{\sigma_x^2}+\frac{y_{\theta}{}^{2}}{\sigma_y^2}\right)\right].\cos(2\pi f x_\theta), \tag{2.10}$$

$$x_\theta = x\cos\theta + y\sin\theta, \tag{2.11}$$

$$y_\theta = -x\sin\theta + y\cos\theta, \tag{2.12}$$

where σ_x and σ_y are the standard deviations of the Gaussian envelope along the x-axes and y-axes, respectively. Figure 2.11 (a) shows a cosine wave with frequency *f*. Figure 2.11 (b) shows a Gaussian envelope. Figure 2.11 (c) shows real part of 2-D Gabor filter. A 3-D representation of real part of Gabor filter is shown in Figure 2.12.



(a) (b) (c)

Figure 2.11: Real part of Gabor filter, (a) cosine wave, (b) Gaussian envelope (c) real part of Gabor filter obtained by multiplying (a) and (b)

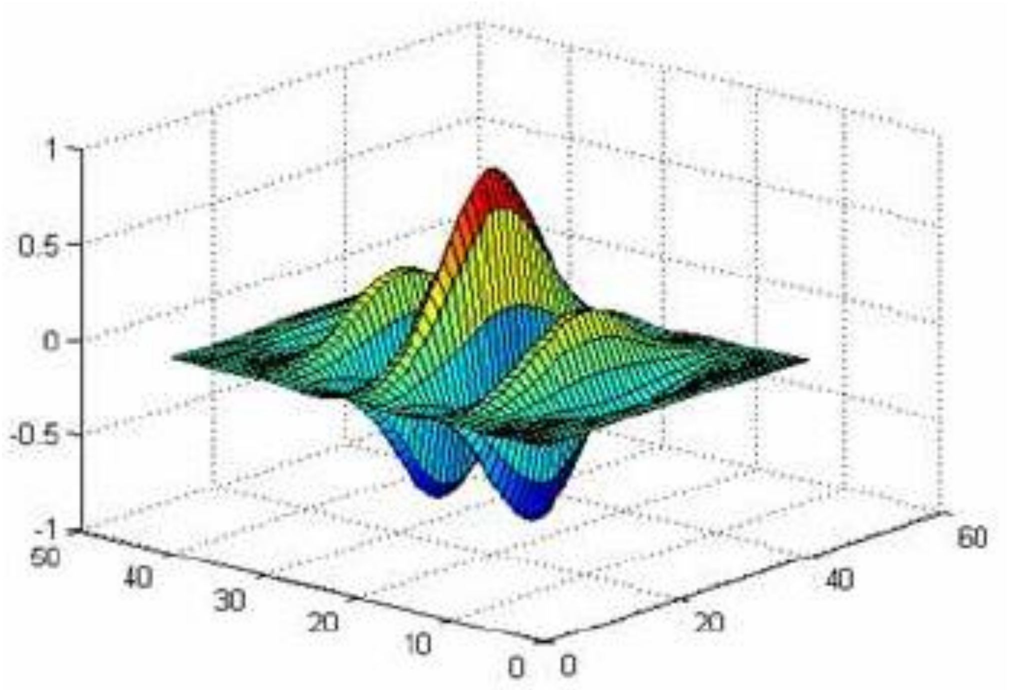


Figure 2.12: 3-D representation of real part of Gabor filter

The block diagram of enhancement based on Gabor filtering presented by Hong et al. [30] is shown in Figure 2.13.

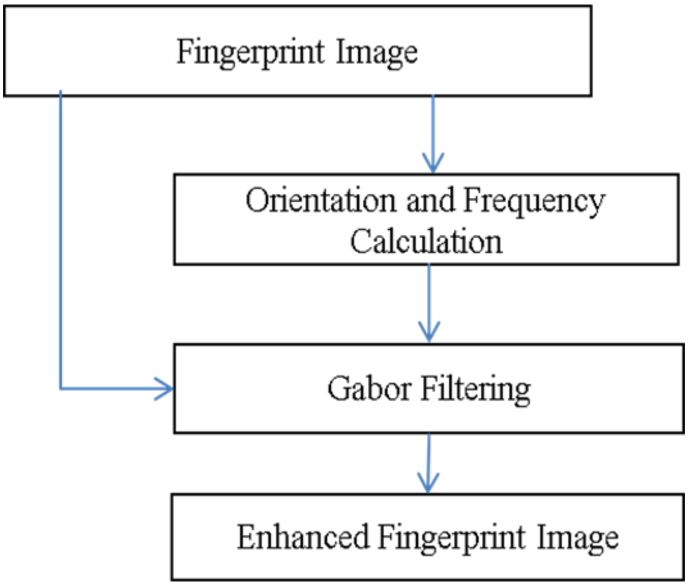
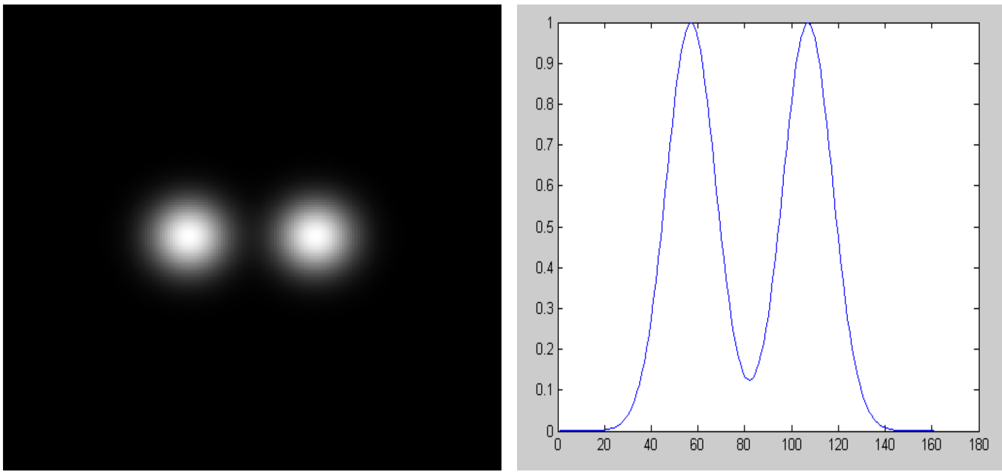

Fingerprint Image
Orientation and Frequency Calculation
Gabor Filtering
Enhanced Fingerprint Image

Figure 2.13: Block diagram of enhancement based on Gabor filtering

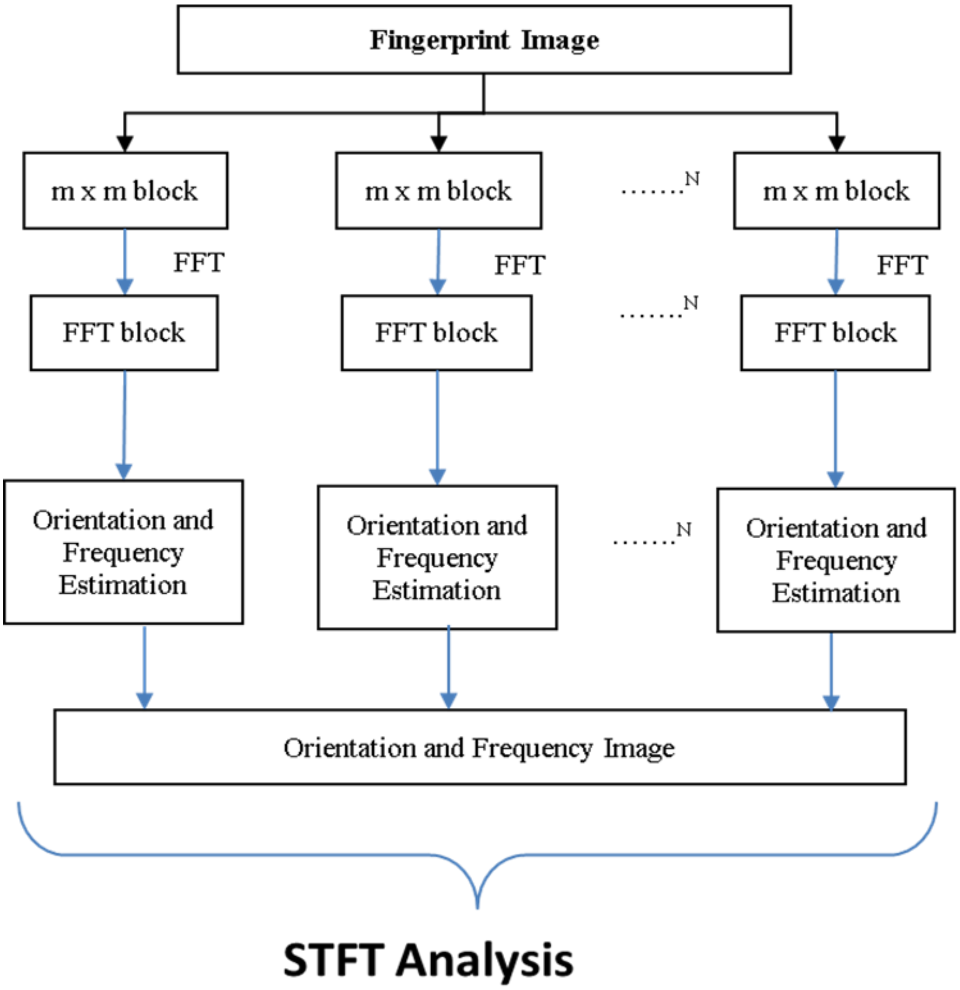
Since this enhancement technique involves 2-D Gabor filtering for each pixel, based on its own frequency and orientation. Therefore, the methodology is effective for enhancing fingerprint images with varying inter-ridge frequencies. The frequency estimates can be erroneous due to presence of scars and broken ridges in a localized region, thus the quality is greatly dependent on the exactness of frequency estimates. Another drawback of Gabor based enhancement is that the frequency response of Gabor filters has a DC component. This DC component can be observed from the frequency plot of the Gabor filter as shown in Figure 2.14. Proper enhancement using these Gabor filters is not possible because the DC component decrease the distinction between the alternating ridge and valley structures. The technique is also computationally intensive, because for every location a separate 2-D Gabor filter is used for enhancement. as discussed in section 3.3.



(a) (b)

Figure 2.14: Frequency response of Gabor Filter, (a) Fourier Spectrum of real part of Gabor Filter, (b) 1-D Frequency response of traditional Gabor Filter

Another variation of contextual filtering based on short time Fourier transform (STFT) is proposed by Chikkerur et al. [31], that applies directional filters in frequency domain locally. The overall algorithm is divided into two stages; STFT analysis followed by a filtering stage. The block diagram of STFT analysis is shown in Figure 2.15. This study models a non-stationary fingerprint image as a locally stationary signal. Therefore, enhancement scheme decomposes image into small overlapping blocks such that inter-ridge distance is almost uniform within a block. Then Fourier transform of each block is computed. The orientation map, frequency map and mask region of the fingerprint image are all simultaneously estimated during STFT analysis.



Fingerprint Image
m x m block
m x m block
.......N
m x m block
FFT
FFT
FFT
FFT block
FFT block
.......N
FFT block
Orientation and Frequency Estimation
Orientation and Frequency Estimation
.......N
Orientation and Frequency Estimation
Orientation and Frequency Image
STFT Analysis

Figure 2.15: STFT Analysis proposed Chikkerur et al. [31]

The next stage is STFT filtering; the overlapping blocks are separately processed in frequency domain to combine the advantage of frequency domain filtering and local adaptation. The enhanced blocks are then obtained by taking inverse Fourier transform. The final enhanced image is obtained by tiling together all the enhanced blocks. The block diagram of STFT based filtering is shown in Figure 2.16. However, due to block by block processing, discontinuities along block boundaries appear, which may also affect the ridge continuity as shown in Figure 2.17.

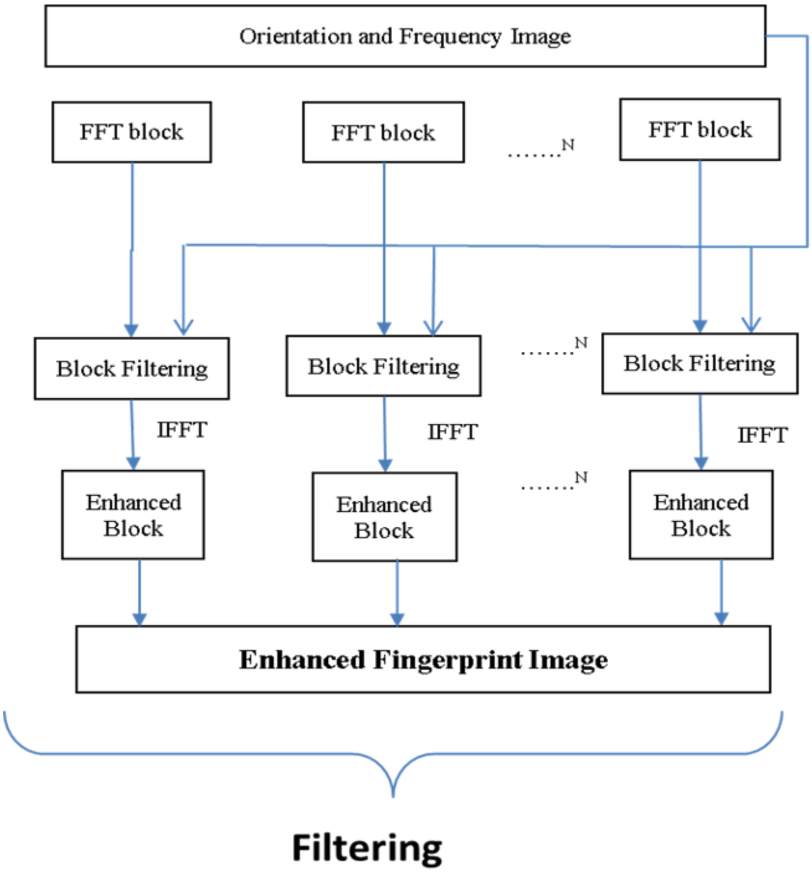
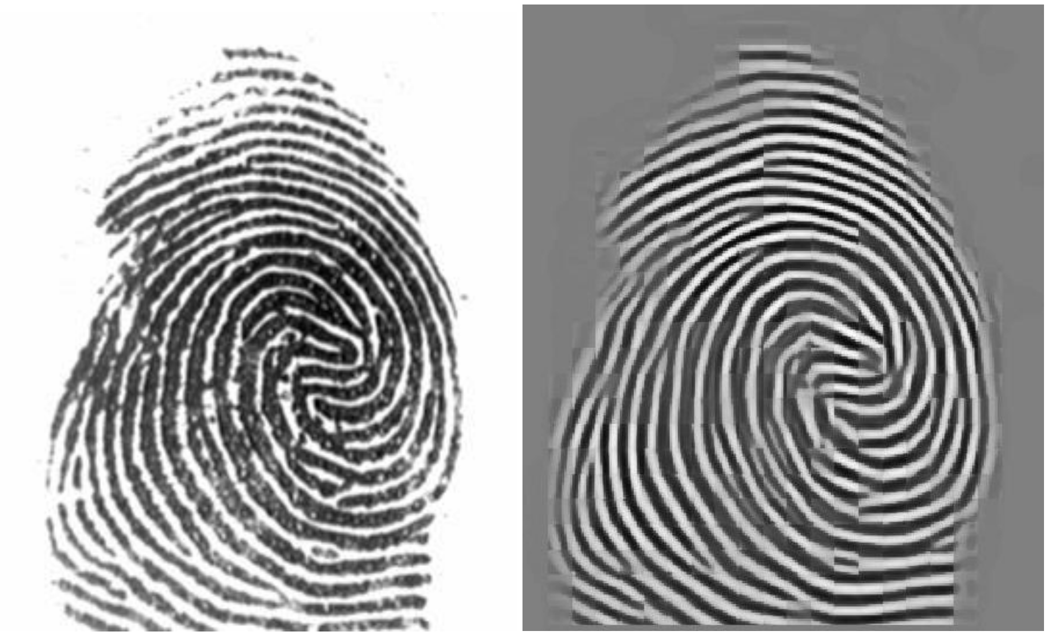


Figure 2.16: STFT filtering proposed Chikkerur et al. [31]



(a) (b)

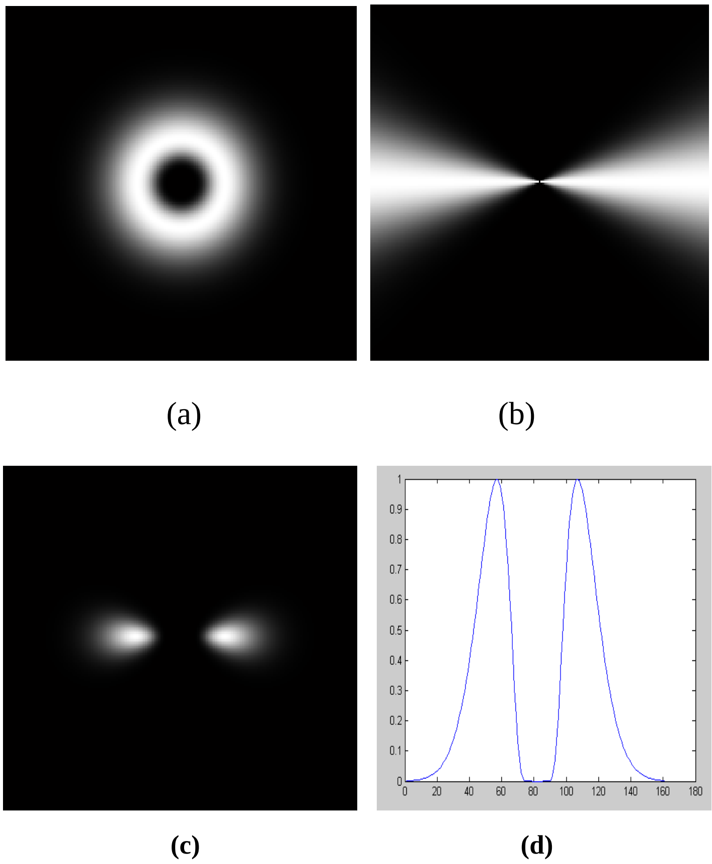
Figure 2.17: Application of STFT enhancement on fingerprint image (a) Fingerprint image 44_1.tif, (b) Enhanced image after applying STFT method

Wang et al. [32], proposed fingerprint enhancement in frequency domain, based on Log-Gabor filters as an improvement to traditional Gabor filters proposed by Hong et al. [30]. A 2-D Log-Gabor filter is given by equation (2.13) and is shown in Figure 2.18.

$$H\ (u,v) = H_r(u,v)H_\theta(u,v) \tag{2.13}$$

$$H_r\ (u,v) = \exp\left[-\frac{1}{2}\left(\frac{\log(r/f_0)^2}{W_r^2}\right)\right] \tag{2.14}$$

$$H_\theta\ (u,v) = \exp\left[-\frac{1}{2}\left(\frac{(\theta-\theta_0)^2}{W_\theta^2}\right)\right] \tag{2.15}$$



(a) (b)

(c) **(d)**

Figure 2.18: Frequency response of Log-Gabor filter (a) 2-D log Band-pass filter, (b) 2-D log fan-filter, (c) 2-D Log-Gabor filter, (d) 1-D Frequency response of Log-Gabor

Unlike traditional Gabor filters, Log-Gabor filters have no DC component and have an extended tail at higher frequencies as shown in Figure 2.18 (d). This scheme uses Hong's method to compute the frequency and orientation map. Whereas, the enhancement procedure is similar to that of Chikkerur et al. [31], that includes application of localized frequency domain filtering. The overall flow diagram of Log-Gabor based enhancement technique is shown in Figure 2.19.

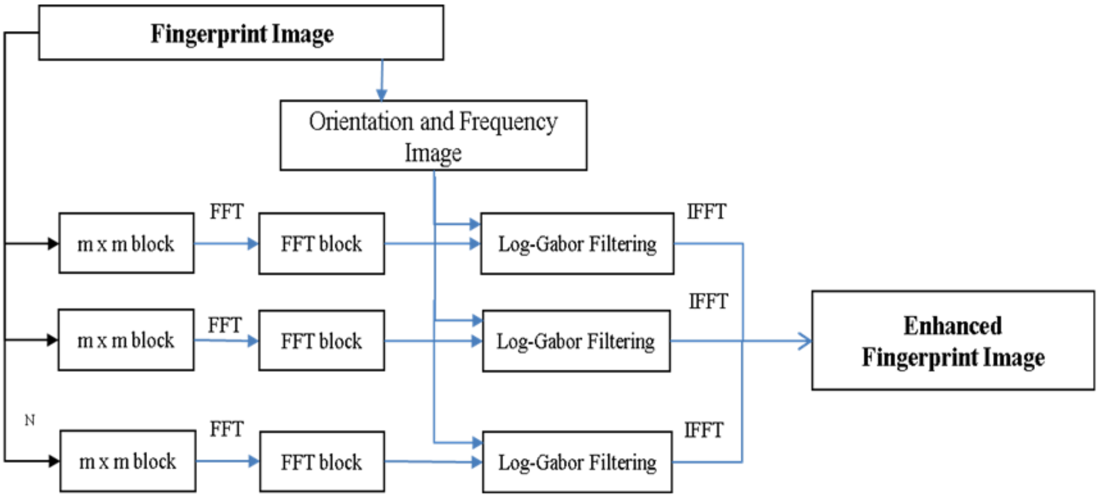


Figure 2.19: Block diagram of enhancement based on Log-Gabor filtering based enhancement technique proposed by Wang et al. [32]

Carsten [49], recently, introduced curved Gabor filters for enhancing the curvy ridge valley structure in fingerprint image. C. Tang et al [50] applied second-order oriented partial differential equations (SOOPDE) to enhance fingerprint images. The proposed technique is effective in joining broken fingerprint ridges as well as it can smooth irregular ridges in fingerprint images, however, it also disturbs the minutia information that is vital for fingerprint matching. F. Turroni et al. [51], proposed an enhancement algorithm where contextual filtering is applied using an iterative method. The algorithm starts enhancing fingerprint image by applying contextual filtering on regions of high-quality and then expanding towards low-quality regions. P. Sutthiwichaiporn et al. [52] proposed an iterative fingerprint enhancement algorithm by using Gaussian-matched filters. These Gaussian-matched filters do not use the local contextual information e.g. local ridge orientation and local ridge frequency. However, the computational complexity of these algorithms is high due to their iterative nature.

2.2.6 Binarization and Thinning

After fingerprint enhancement, the enhanced images are generally binarized before features are extracted from them. Binarization is the process that converts a gray level image into a binary image, where white pixels represent the foreground ridges, and the black pixels represent the background valleys.

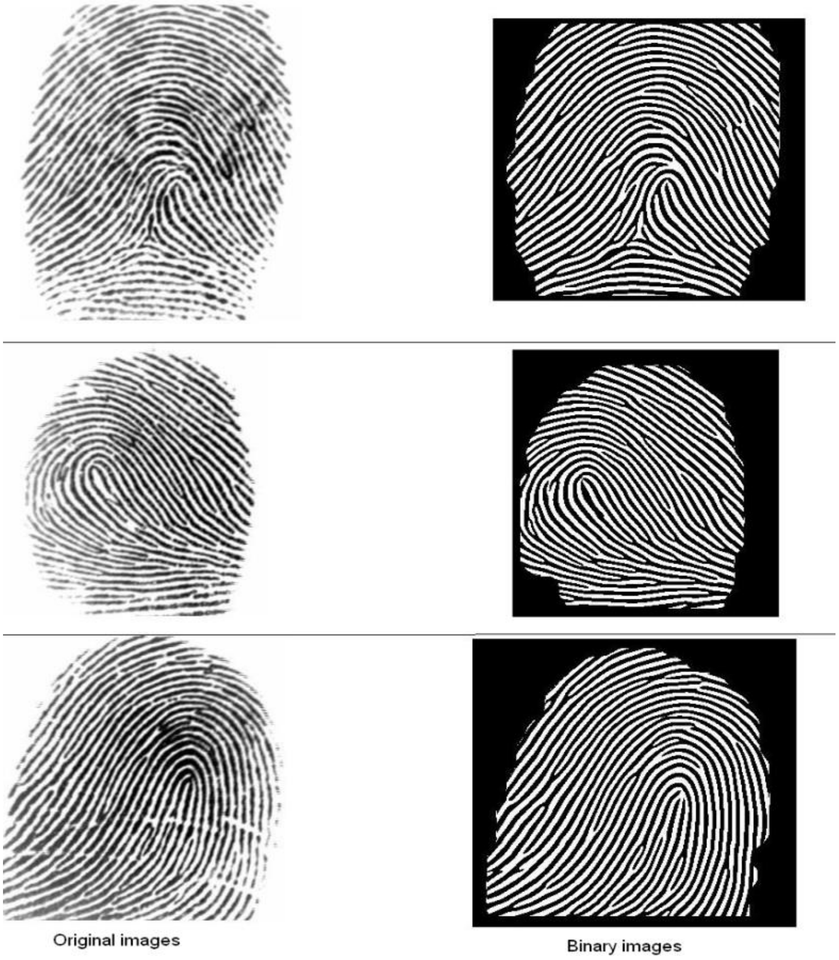


Figure 2.20: Binarization of images

Binarization is performed by choosing a threshold value and then classifying all pixels with values above this threshold as white, and all other pixels as black. The problem is how to select a correct threshold. This problem is resolved by image enhancement using the filter banks, because they reduce the illumination variations and have zero mean due to rejection of all lower frequencies. Thus, the DC level of the filtered image i.e. zero can be used as a global threshold. The results of binarization process are shown in Figure 2.20.

The final enhancement step typically performed prior to feature extraction is thinning. Thinning is a morphological process that reduces the width of ridges to a single pixel [33], [34]. The application of the thinning algorithm to a fingerprint image preserves the connectivity of the ridge structures while forming a skeletonized version of the binary image. This thinned image is then used in the next step of feature extraction. The results of thinning process are shown in Figure 2.21.

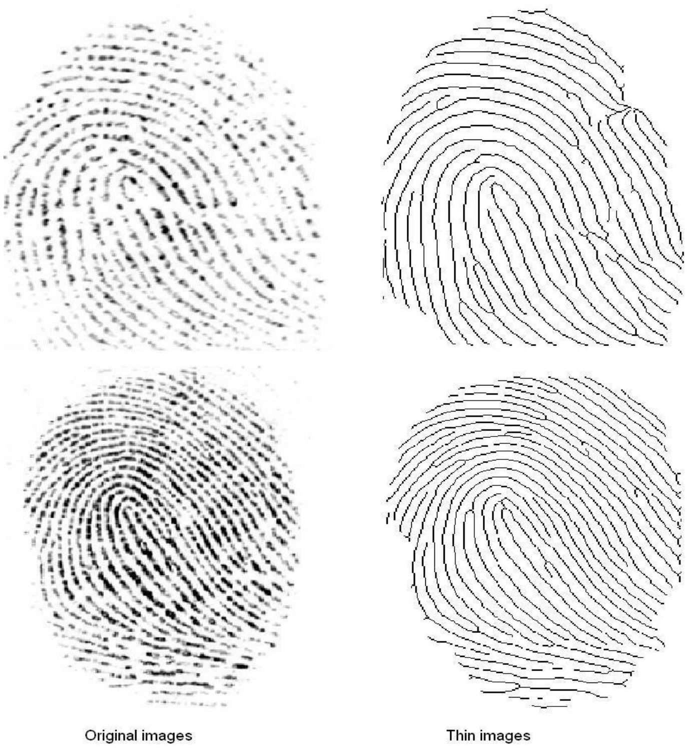


Figure 2.21: Results of Thinning Process

2.3 Feature Extraction

Fingerprint images contain many unique features that are used by researchers for fingerprint matching. For example, ridges, singular points (core point) and minutiae. Core point is the north most point of the inner most ridge in fingerprint image [61].

According to the ANSI/NIST-ITL standard working draft for data format of fingerprint features, “a core is located at the focus of the innermost re-curving ridgeline of a ridge pattern: if the ridge is viewed as a section of a circle, the core is the center of that circle; if the ridge is viewed as an ellipse or parabola, the core is the focal point of that curve”. Core point is considered as reference point in fingerprint and can be effectively used for fingerprint classification and matching [35], [36], [37]. However, in case of latent or partial fingerprint images, presence of core point is doubtful. Furthermore, core point detection and classification algorithms are error prone and may increase the overall error of the system. The most common and reliable features used for fingerprint matching techniques are minutiae.

These minutiae features were first observed by Sir Francis Galton based on the discontinuities of local ridge structure [11]. Therefore, these minutiae points are also known as “Galton details”. Every minutia point has its own position coordinates and angle, and a pattern of such points can be used for reliable discrimination of fingerprints. Although, there are several different types of minutiae, seven most common minutiae types are shown in Figure 2.22 [3].

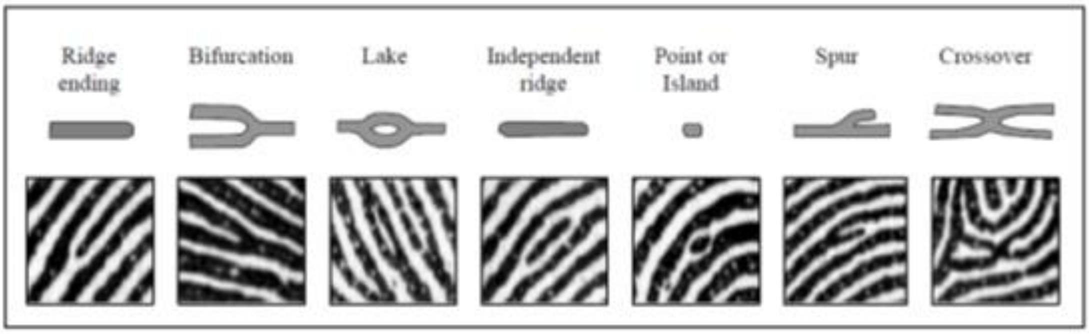


Figure 2.22: Seven most common minutiae types

Among these minutiae types, ridge ending and ridge bifurcation are most commonly used for fingerprint authentication. This is because ridge ending and ridge bifurcation are very stable and the algorithms to automatically detect these features are very accurate in comparison to the rest. Moreover, all other minutiae features can be realized as combinations of ridge endings and ridge bifurcations. Therefore, there is no need to detect other features separately.

These minutiae features are generally extracted from thinned fingerprint image. Thinning algorithms also introduce some irregularities and noise in ridge structure, which leads to detection of spurious minutiae.

Since the matching results heavily depend on quality of extracted minutiae. Therefore, to ensure the detection of genuine minutiae only, researchers proposed certain heuristics to reject the false minutiae as a post-processing step for minutiae validation [38], [39], [40]. The algorithm of minutiae detection and false minutiae removal is discussed below.

2.3.1 Minutiae Extraction

The candidate minutiae are extracted by scanning the local neighborhood of each ridge pixel (having value 0 in thin image) in a window of *3 x 3*. A ridge pixel is classified as a candidate minutia based on the number of 0 to 1 transitions occurred while scanning its 8 neighboring pixels in clockwise direction. If there is only one 0 to 1 transition then the pixel is classified as ridge ending and if the number of 0 to 1 transitions is 3 then the pixel is classified as ridge bifurcation. Ridge ending and ridge bifurcation scenario is represented in Figure 2.23 [3].

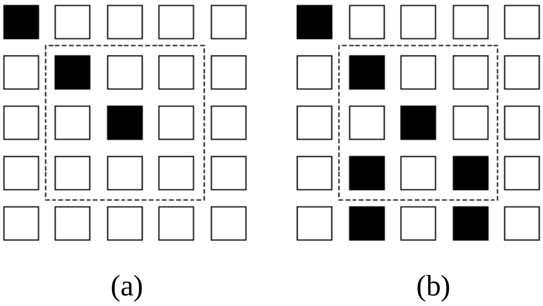


Figure 2.23: Minutia Structure (a) Ridge Ending, (b) Ridge Bifurcation

2.3.2 False Minutiae Removal

False minutiae may be introduced in the fingerprint image due to the factors such as noisy image, and image artifacts created by the thinning process. Minutiae post-processing technique is used to filter out the false minutiae from the fingerprint image. Figure 2.24 illustrates some examples of false minutiae, which includes spur, hole, triangle, and spike structure Maltoni and Maio [3].

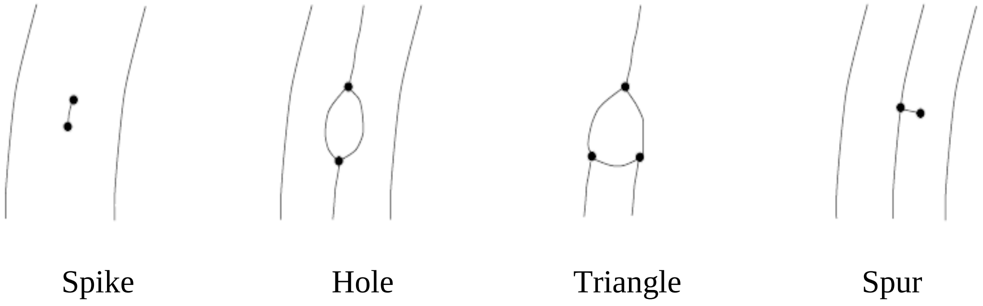


Figure 2.24: Examples of false minutiae

There are numerous proposed approaches for image post-processing based on certain heuristics to eliminate false minutiae. In this study, the technique proposed by Marius Tico and Pauli Kuosmanen [39], has been followed for minutiae post-processing. The proposed technique authenticates each candidate minutia by analyzing the thin image in a window of size *W* x *W* around each candidate minutia. An example of final results of minutiae extraction after false minutiae removal is shown in Figure 2.25.

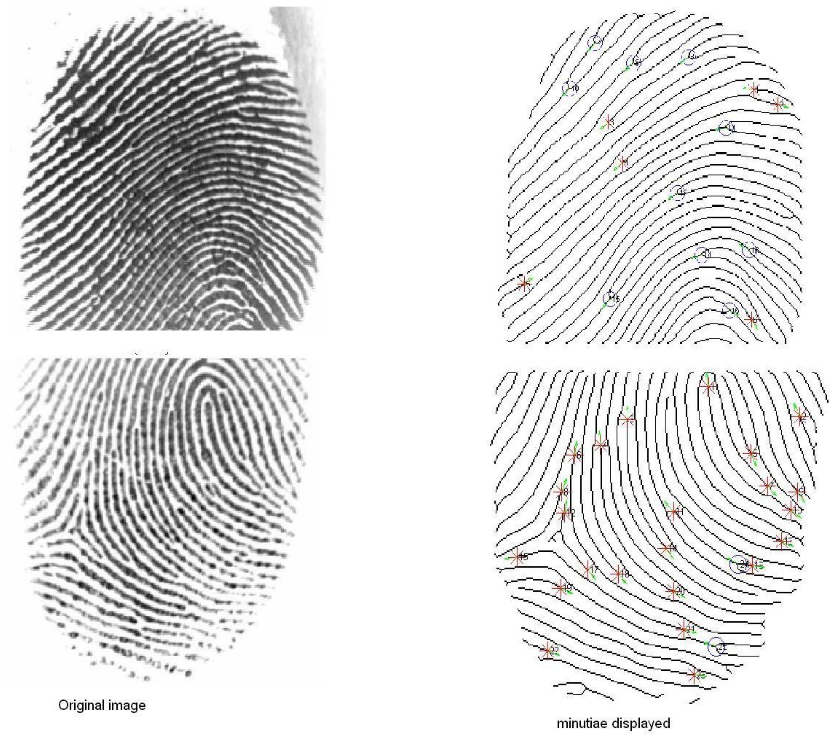


Figure 2.25: Results of Minutia Extraction after False Minutiae Removal

As a result of minutiae extraction and false minutiae removal procedure, a set of minutiae will be obtained, where each minutia point is represented by its location along with its reference direction. Minutia type (ridge ending or bifurcation) can also be made part of this feature set and can be used in matching. However, minutia type detection is erroneous that is why it is not generally made part of minutia feature set.

2.4 Feature Encoding and Matching

Once the features are extracted from a fingerprint image, the next important step is to represent these features into a compact representation where all the information of extracted features remains intact and redundant information is removed. This transformation is called feature encoding process. An important consideration for the encoding scheme is to ensure the suitability of the transformed data to the matching algorithm. Fingerprint encoding and matching algorithm are inter-dependent on each other. These algorithms can be broadly characterized into three groups, i.e. matching based on correlation, matching based on texture descriptors and matching based on minutiae features.

In correlation based algorithms, fingerprint images are superimposed over each other for different alignments and a similarity score is calculated based on their correlation. The drawback of these techniques is that they do not properly handle rotational and scaling misalignments. Moreover, these algorithms have high computational complexity and storage costs which make them unsuitable for fingerprint matching.

In texture based encoding and matching techniques, a reference point is detected initially and fingerprint area around this point is then encoded based on texture patterns [35] or by mapping minutiae in fingerprint image onto that reference point [36], [37].. The reference point in these schemes is normally the core point (singular point) of fingerprint [3]. These schemes are not only computationally efficient, but also require less storage space. However, these techniques are not used in commercial large scale applications, because core-point detection algorithms are not reliable. Therefore, core-point based mapping schemes may increase the overall error rates of matching system.

The most widely encoding and matching techniques rely on minutiae features based point pattern encoding and matching techniques. Apart from being very accurate, these techniques support partial fingerprint matching. The general methodology of fingerprint encoding and matching based on neighboring minutiae is discussed in following subsection.

2.4.1 General Methodology of Local Minutiae Encoding and Matching

Encoding

Consider a template fingerprint with M minutiae $(x_i, y_i, \theta_i)^T$,(i= *1,2,3,…, M*). Each minutia point in the template fingerprint M_i is represented by x, y, θ. Where x and y are the location coordinates, and θ is the orientation or angle along x-axis of each minutia point in the fingerprint image.

In order to encode a template fingerprint image, each minutia is selected as reference and its distance and relative angle is measured from its neighboring N minutia as shown in Figure 2.26.

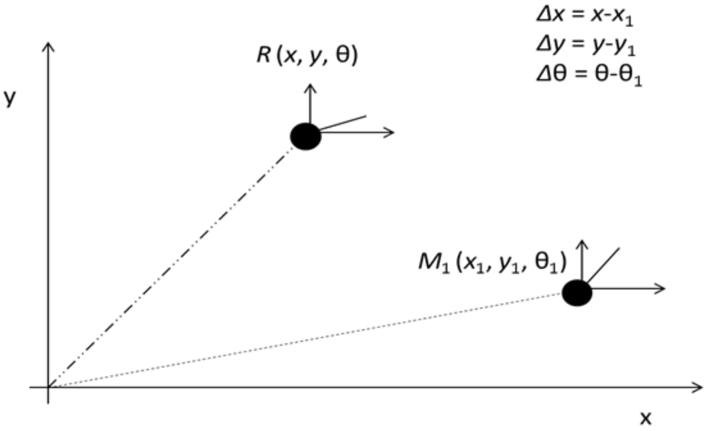


Figure 2.26: Fingerprint encoding based on neighboring Minutia

Let P={(Δx_1, Δy_1, $\Delta\theta_1$) ^{T}, …..(Δx_N, Δy_N, $\Delta\theta_N$)^T}, where P denotes the set of N neighboring minutiae used in encoding of a reference minutia R. The final biometric-code of the fingerprint is a combination of all such individual minutia encodings R_i, where *i=1,2,3…M*. This biometric-code is stored in the database as a template of that fingerprint and used for future matching.

Matching

In order to match a candidate fingerprint with template fingerprints stored in the database, the fingerprint is acquired from the fingerprint sensor. Preprocessing, enhancement and minutiae extraction and encoding steps for the candidate fingerprint image are same as the template fingerprint.

For matching two fingerprints based on point pattern matching technique, an exhaustive search algorithm is employed, where every minutia code from template will be matched with all minutia codes in candidate fingerprint to find the best minutia pair matching of template and candidate fingerprint as shown in Figure 2.27.

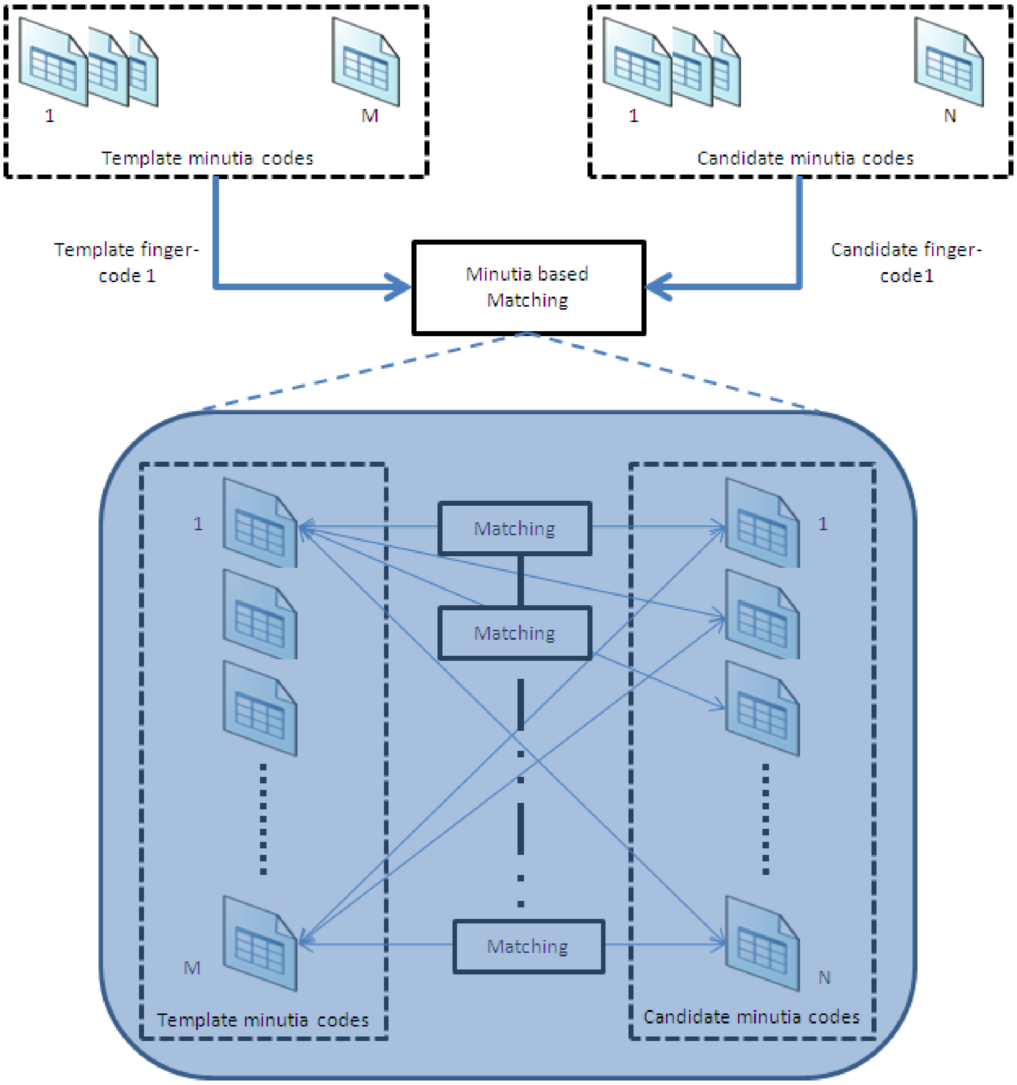


Figure 2.27: Fingerprint matching based on neighboring Minutia

Matching one minutia code of template to one candidate minutia code requires matching of every component (neighboring minutia) of template minutia code with candidate minutia code. This is because the order of occurrence of neighboring minutia is not certain especially in case of any missing or spurious neighboring minutia. Therefore, for finding correct correspondence every component (neighboring minutia) of minutia code of template code must be matched with every component of minutia code of candidate code. In order to match one template minutia code, consisting of n neighbors with one candidate minutia code, n^2 component matching's are required. The accuracy of matching algorithm generally increases as the number of neighboring minutiae for encoding increases. However, by increasing number of neighbors for encoding not only the size of biometric-code increases, but also the number of computations involved in matching.

The overall computational complexity of minutia neighbor based point pattern matching algorithms become M x N x n^2, where n is number of neighbors used for encoding and M and N are number of minutiae in template and candidate fingerprints respectively. Based on minutiae pair matching, a score is calculated that describes the similarity between two fingerprint images.

2.4.2 Existing Techniques of Minutiae Encoding and Matching

Although, the encoding and matching techniques are different, yet the basic methodology remains the same and that is, each minutia is encoded based on the relative distances and angles with neighboring minutiae [41], [42], [43].

Jiang and Yau [12], proposed a rotation and translation invariant encoding and matching technique based on minutia-triplet structure. In minutia-triplet based encoding, each minutia is taken as center and a feature vector is formed by taking its distance and orientation difference with its nearest two neighboring minutiae in polar coordinates. An example of encoding a minutia m_i (x_i, y_i, θ_i) with respect to its two nearest minutiae n_0 (x_{n0}, y_{n0}, θ_{n0}) and n_1 (x_{n1}, y_{n1}, θ_{n1}) is shown in Figure 2.28 [94]. The feature vector for minutia m_i can be represented as:

$$F_\mathrm{i} = (r_{\mathrm{i}0}, r_{\mathrm{i}1}, \theta_{\mathrm{i}0}, \theta_{\mathrm{i}1}, \varphi_{\mathrm{i}0}, \varphi_{\mathrm{i}1}, n_{\mathrm{i}0}, n_{\mathrm{i}1}, t_\mathrm{i}, t_0, t_1)^\mathrm{T} \tag{2.16}$$

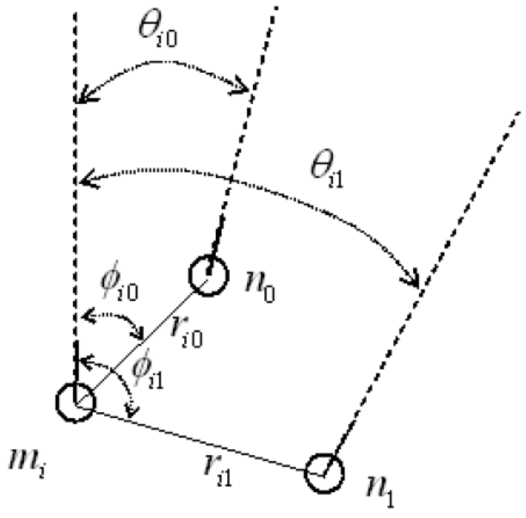


Figure 2.28: Fingerprint encoding based on Minutiae triplet structure proposed by Jiang and Yau [12]

In equation (2.16), r_{i0} and r_{i1} represents the distance between the minutia m_i and its nearest neighboring minutiae n_0 and n_1 in polar coordinates. θ_{i0} and θ_{i1} represents the orientation differences between minutia m_i and its corresponding neighboring minutiae. φ_{i0} and φ_{i1} are the differences of orientations between θ_i and the orientation of the line segment connecting m_i and corresponding neighboring minutia n_0 and n_1. n_{i0} and n_{i1} represents the number of ridges between minutia m_i and n_0 and n_i. Finally, t_i, t_0, t_1 corresponds to the type of each minutia in the minutia-triplet structure, e.g. ending or bifurcation. Jiang and Yau [12], then proposed a local minutiae matching scheme by computing similarity between all possible combinations of minutiae pairs based on weighted Euclidean distance between their corresponding feature vector. The overall matching score between candidate and template fingerprints can be estimated by counting the number of matched minutiae pairs. In order to improve the matching score estimates, a consolidation stage is applied to validate whether the local matching results holds at the global level too. This is performed by aligning all the locally matched minutiae pairs to a minutiae pair that best define the transformation between the candidate and template minutiae pairs. For matching partial fingerprints, Jea et al. [43] employed neural networks to calculate normalized matching score within the overlapping region of candidate and template fingerprints.

Ratha et al. [13], proposed fixed radius based local encoding technique to overcome the problems of spurious or missing minutiae. In this scheme, a feature vector is formed for every minutiae by encoding all the neighboring minutiae that exists within some fixed radius from each minutia. This leads to variable code length for each minutia and can tolerate spurious or missing minutiae at the cost of increasing complexity of local matching algorithm. Tico et al. [44], proposed another variation of minutiae encoding and matching, where each minutia is encoded based on the orientation fields in the close proximity of individual minutia. Feng [45], presented, a hybrid technique that uses fixed-radius local minutiae as well as orientation information for fingerprint matching.

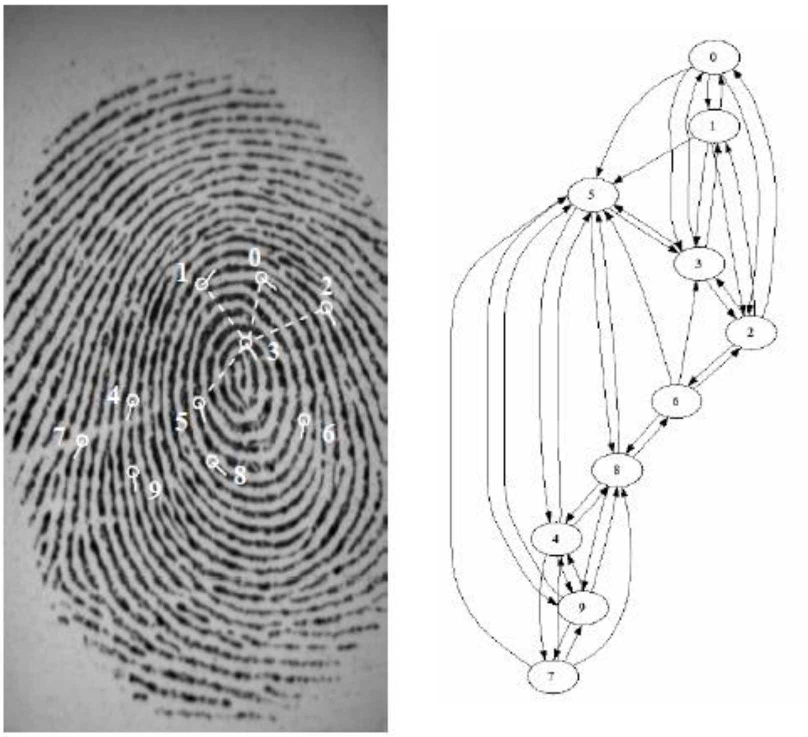


Figure 2.29: Left figure highlights the *k*-plet local structure for minutia 3 and right figure shows the graph representation for the fingerprint based on *k*-plet representation

Chikkerur et al. [41], proposed a graph representation to encode fingerprint based on local structure called *k*-plet. In *k*-plet representation, each minutia is encoded by neighboring *k* minutiae such that the *k* nearest neighbors falls in all the four quadrants around the central minutia. The encoding procedure will be less erroneous in case of presence of spurious minutiae around the central minutia and also preserve more connectivity between minutiae. An example of *k*-plet local structure for minutia number 3 and the graph representation for minutiae points in fingerprint is shown in Figure 2.29 [3]. For the given example *k* is chosen to be 4.

For fingerprint matching, Chikkerur et al. [41], proposed local matching based on of *k*-plet structure using dynamic programming followed by a novel Coupled Breath First Search (CBFS) algorithm for global matching. In CBFS the matching between two graphs is carried out by initializing the matching from an arbitrary pair of node (minutia) of candidate and template fingerprints. In graph traversal all the neighboring nodes that are matched locally are visited and matched in Breath First Search (BFS) manner. The matching score for that pair of node is then obtained by the number of nodes matched at the end of BFS algorithm. As the optimal minutia pair correspondence is not known before the start of BFS algorithm, so the complete graph traversal algorithm is repeated for every node (minutia) pair in graphs. The node pair that gives maximum number of matches is considered as the best minutia pair and the transformation between candidate and template fingerprints can be found by aligning the fingerprints based on this minutia pair. The number of matches for the corresponding node pair is taken as the consolidated matching score for the fingerprints. The implementation of matching algorithm proposed by Chikkerur et al. [41] is online available at: (http://sites.google.com/site/sharatchikkerur/code). This code is used for comparison of the proposed enhancement algorithm with other enhancement techniques in Chapter 3 and Chapter 4.

2.5 Summary

This chapter presents the literature survey on fingerprint based authentication system. A brief survey of existing algorithms for feature extraction and matching is presented along with emphasis on the different state-of-the-art contextual filtering based enhancement techniques. These contextual filtering based enhancement techniques exist either in spatial or in frequency domains with their pros and cons. In next chapter, a novel two-fold contextual filtering based enhancement technique is presented that process the fingerprint image both in spatial and frequency domains to overcome the drawbacks of existing techniques.

Chapter 3

2-FOLD CONTEXTUAL FILTERING FOR ENHANCEMENT

3.1 Proposed Enhancement Scheme

As discussed section 2.2.5, fingerprint enhancement schemes based on contextual filtering are generally carried out either in frequency domain or in spatial domain. However, in the proposed study a novel technique is employed to combine the advantages of both frequency and spatial domains for enhancing fingerprint images.

As ridge patterns approximately follow a sinusoidal wave pattern, therefore, the ridge frequency estimation can be handled efficiently in frequency domain rather than spatial domain. In the proposed enhancement scheme, frequency domain filtering caters for filtering out non-ridge noisy information and background region from valid fingerprint information.

The degradation in fingerprint image because of presence of broken ridges and scars can be resolved by joining ridges in the orientation of ridges. Algorithms for estimation of orientation in spatial domain are usually very accurate, because they are based on local gradients and a smoothing process takes care of noise and discontinuities of ridges. In order to resolve the problem of broken ridges and scars, filtering is carried out in spatial domain.

The proposed enhancement technique is based on the following important observations noted during extensive testing of previous algorithms.

1. Algorithms for frequency calculation in spatial or frequency domain are prone to errors because of the presence of scars, broken and creases running across ridges in fingerprint images. In particular, at high curvature regions and localities consisting of ridge endings and ridge bifurcations, the estimates are usually not very accurate. Therefore, image is not perfectly enhanced at the location of minutiae, thus affecting the minutiae extraction and matching afterwards.

2. Moreover, in previously proposed algorithms, a different filter is applied at every locality tuned based on local frequency and orientation, thus increasing the overall complexity of the algorithm.

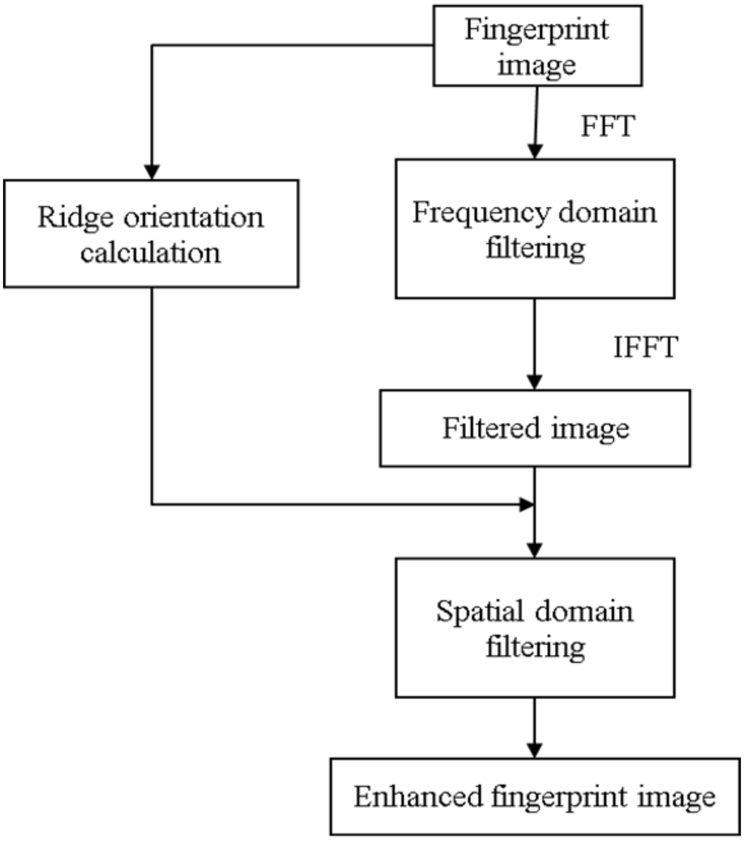

Fingerprint image
FFT
Ridge orientation calculation
Frequency domain filtering
IFFT
Filtered image
Spatial domain filtering
Enhanced fingerprint image

Figure 3.1: Algorithm flow Diagram for the proposed fingerprint enhancement technique

The proposed two-fold enhancement technique avoids dependence on estimation of frequency. Therefore, it is not prone to errors in the frequency estimation process and is effective for enhancing low quality fingerprint images [15]. It also employs same filter for the whole image instead of designing separate filter for each local region in an image. The advantage of this methodology is that the computational complexity remains low, comparable to the most well-known and efficient techniques [31, 32] available in literature as discussed in section 3.3. The overall algorithm flow of the proposed two-fold enhancement technique is shown in Figure 3.1. A detailed explanation of each functional block is given in the following subsections.

3.1.1 Frequency Domain Filtering

Figure 3.2, presents the main algorithmic steps of frequency domain filtering.

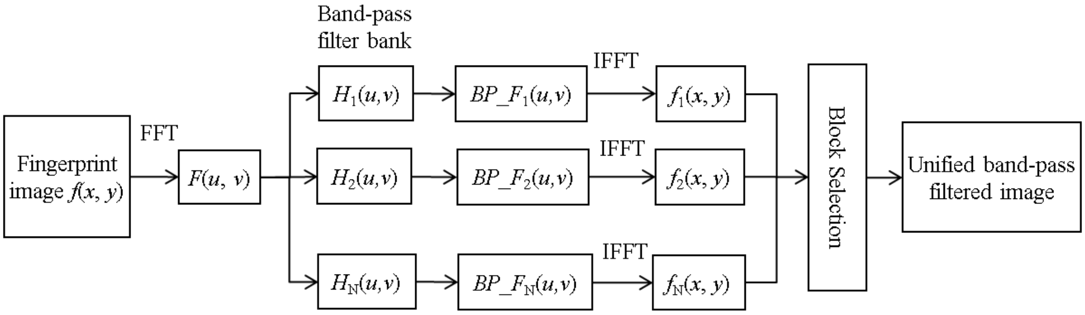


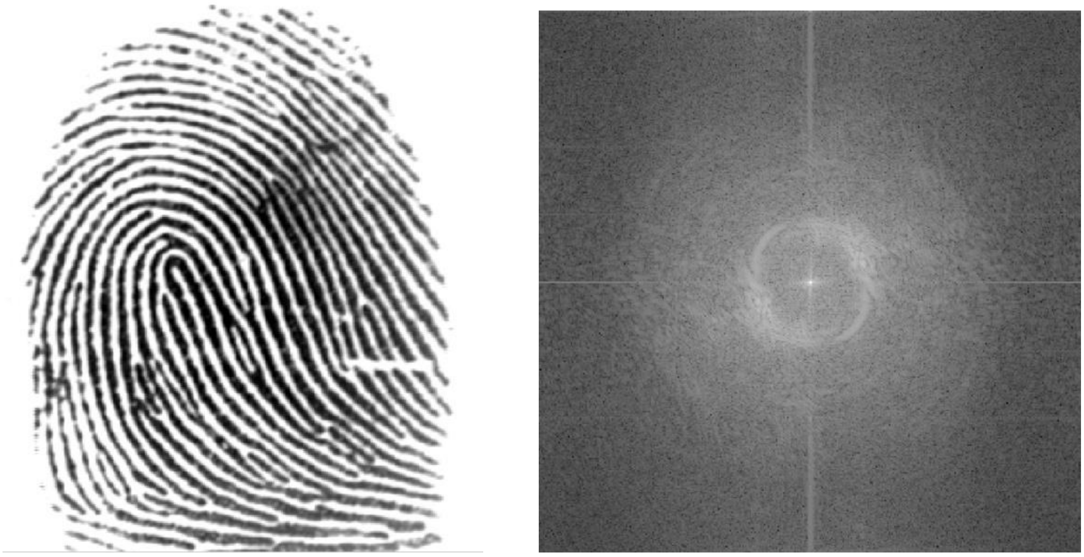
Figure 3.2: Algorithmic steps of frequency domain filtering

Which are:

- Conversion of fingerprint image into frequency domain through Fast Fourier Transform FFT. Here *F* represents the Fourier transform of the fingerprint image.
- The transformed image is then filtered with different frequency band-pass filters and reconstructed in spatial domain using Inverse Fast Fourier Transform IFFT.
- A Unified band-pass filtered image is constructed from individual filtered band-pass images (f_1, f_2, … f_n) by selecting sub-blocks from the set of individual filtered images by choosing a corresponding sub-block having the maximum variance.

3.1.1.1 Band-Pass Filtering

In fingerprint images the ridges can be modeled as a combination of different sinusoidal wave forms having different directions [30], and generate a localized spectrum in the form of a band in frequency domain as shown in Figure 3.3 (b).



(a) (b)

Figure 3.3: Application of Fourier transform on fingerprint image, (a) A fingerprint image, (b) Fourier transform of (a)

The dominant band in frequency domain is dependent on the inter-ridge distance that varies from d_{min} to d_{max} pixels in spatial domain, where d_{min} is generally 5 and d_{max} is 15 for 500-dpi fingerprint images. In order to pass ridge patterns having inter-ridge distances from d_{min} to d_{max} pixels, fingerprint image can be converted into frequency domain, and a band-pass filter can then be applied to filter only the ridge patterns. The dominant band in frequency domain will be thinner if all the ridges have approximately same inter-ridge distances and band will be wider if ridges have varied inter-ridge distances, which is mostly the case in fingerprint images. Therefore, a single band-pass filter will not be beneficial because in order to pass all valid frequencies, a wider band-pass filter is required that will also pass frequency coefficients outside the band of local ridges.

In the proposed algorithm, a fingerprint is filtered with a number of thin band-pass filters and response of each filter is then analyzed locally. The choice of the number of band-pass filters '*n*' in the filter bank is important as it defines the frequency resolution for filtering. For filtering in frequency domain, Fourier Transform of the fingerprint image $F(u,v)$ is computed and a bank of individual band-pass filtered images $BP\_F_i(u,v)$ are then computed by applying '*n*' band-pass filters on $F(u,v)$ as shown in equation (3.1) and (3.2). The value of '*n*' is empirically chosen by calculating error rates as explained in section 3.4 for different values of ‘*n’*. It has been observed that the error rate generally decreases as ‘*n’* increases. However, it does not decrease significantly as ‘*n’* is increased beyond *n* = 5, while the computational complexity increases significantly. Therefore, in this study, we have set *n* = 5.

$$f(x,y) \xrightarrow{FFT} F(u,v), \tag{3.1}$$

$$BP\_F_i(u,v) = H_i\ (u,v)F(u,v), \qquad where\ \ 1 < i < n, \tag{3.2}$$

where $H_\mathrm{i}(u,v)$ is Gaussian band-pass filter and is expressed in equation (3.3).

$$H_i(u,v) = \exp\left[-\frac{1}{2}\left(\frac{D^2(u,v) - D_{0i}^2(u,v)}{D(u,v)W}\right)^2\right], \qquad where\ \ 1 < i < n, \tag{3.3}$$

where, $D(u,v)$ is the distance of each pixel from the center of the Fourier Transform (FT) image. $D_{0\mathrm{i}}(u,v)$ defines the frequency of i^{th} band. W is the width of filter and i has the range $1 < i < n$. The radial distance $D_{0\mathrm{i}}(u,v)$ of i^{th} band-pass filters from center of the FT image is given by equation (3.4).

$$D_{0(i)}(u,v) = (D_{0u(i)} * D_{0v(i)}) / \sqrt{D_{0u(i)}^2 v^2 / (u^2+v^2) + D_{0v(i)}^2 u^2 / (u^2+v^2)}, \tag{3.4}$$

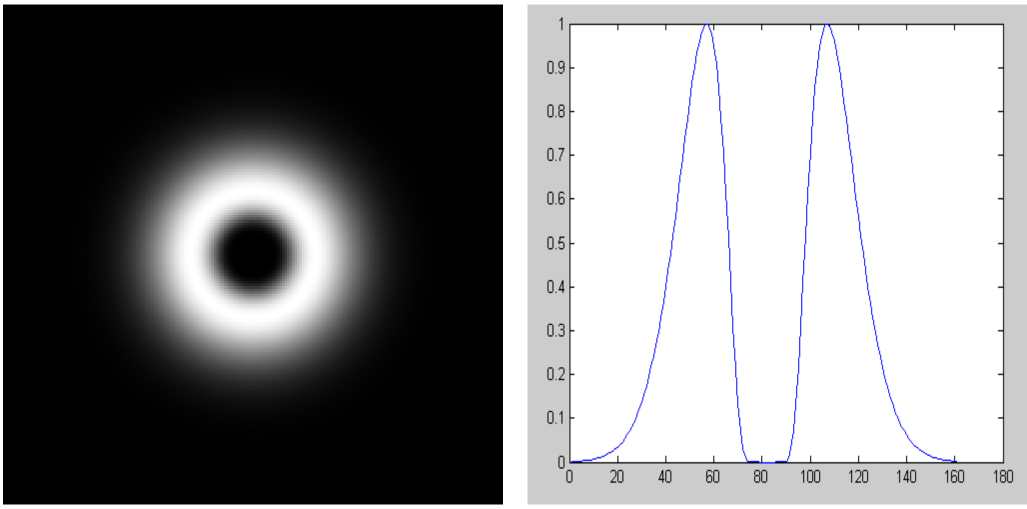
$$D_{0u(i)} = M / \Delta d_i\, ,\ D_{0v(i)} = N / \Delta d_i, \tag{3.5}$$

$$\Delta d_i = 1/2(d_{\max} + d_{\min}) + 1/n * (d_{\max} - d_{\min})(i - (n-1)/2), \tag{3.6}$$

Here, M and N are the size of fingerprint image in horizontal and vertical directions respectively; u and v are the frequency components in horizontal and vertical directions. $D_{0\mathrm{i}}(u,v)$ will be constant for a square image, but it will be different for different directions in a rectangular image. W is the width of the filter; it defines the Gaussian spread of the band-pass filter and is dependent on number of band-pass filters and size of fingerprint image. Width W will become smaller as the number of band-pass filters increases as shown in equation (3.7).

$$W = \frac{(M+N)}{2}\frac{1}{(12+0.3(n-1))}. \tag{3.7}$$

In this case, the maximum width *W*, is empirically set to be (1/12)*(*M*+*N*)/2, and the width gradually reduces as the number of bands '*n*' increases. Since the actual Gaussian spread in equation (3.3), is proportional to *D*(*u*,*v*), the spread is not constant on both sides of the band. The filter has a much smaller spread at the inner side of band and longer spread at the outer side of the band. This is a skewed band-pass filter which is suitable for retaining significant higher frequency details that is important for preserving accurate ridge valley structure. The proposed band-pass filter as shown in Figure 3.4 (a) exhibit properties equivalent to Log-Gabor filter [32], as the DC component of this filter is zero.



(a) (b)

Figure 3.4 : Frequency response of the proposed Band-pass filter, (a) Band-pass filter, (b) 1-D Frequency response of the proposed Band-pass filter

3.1.1.2 Construction Of Unified Band Pass Filtered Image

The individual band-pass filtered images *BP_F$_i$*(*u*,*v*), after filtering in frequency domain, are brought to spatial domain *f$_i$*(*x*,*y*) through Inverse Fourier Transform IFFT.

$$BPF_i(u,v) \xrightarrow{IFFT} f_i(x,y), \qquad where\, 1 < i < n, \tag{3.8}$$

where *n* is the number of band-pass filters. These filtered images *f$_i$*(*x*,*y*) are then combined to form a unified band-passed filtered image. Unified band-passed filtered image is obtained by selecting sub-blocks from the set of individual filtered images by choosing a corresponding sub-block having the maximum variance. This can be mathematically expressed as:

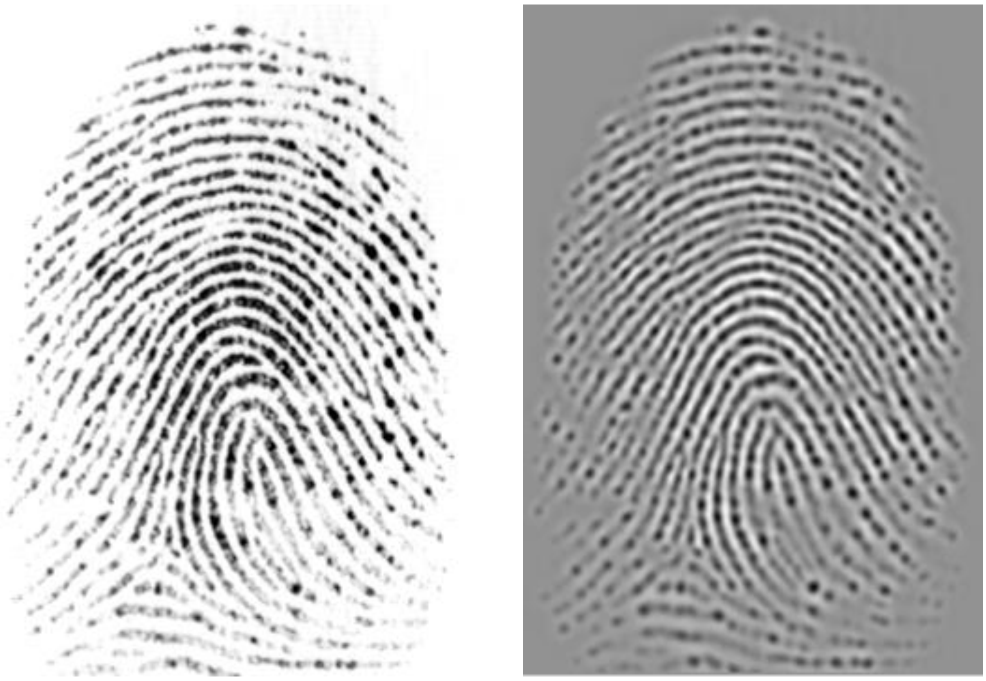
$$f(x,y) = \quad f_h(x,y) \quad L(s-1/2) < x < L(s+1/2), L(t-1/2) < y < L(t+1/2), \tag{3.9}$$

$$where \quad 1 < s < M/(L), 1 < t < N/(L), \tag{3.10}$$

$$h = \arg\max{}_i (\mathrm{var}(f_{i=1,2,..n}(x,y)), \tag{3.11}$$

$$\mathrm{var}(f_i(x,y)) = \sum_{x=L(s-1)}^{L(s+1)} \sum_{y=L(t-1)}^{L(t+1)} (f_i(x,y) - \tilde{f}_i(x,y))^2, \tag{3.12}$$

where *L* x *L* is the size of block which is being filled. The block size should be chosen such that it is not large enough so that the spatial frequency within the block remains approximately constant. Moreover, in order to keep the computational cost low, the block size should not be very small either. For a fingerprint image of resolution 500 dpi, a size of 9*9 is found to be suitable. It should be noted that the variance of the block is calculated over the size of *2L* x *2L*.



(a) (b)

Figure 3.5: Enhanced fingerprint images after proposed directional smoothing method, (a) Fingerprint image 14_7.tif, (b) Unified band-pass enhanced image.

Figure 3.5 shows the results of the proposed frequency domain filtering approach. Figure 3.5 (a) shows the fingerprint image ‘14_7.tif’ taken from FVC 2002 DB1-A [53], and Figure 3.5 (b) shows the unified band-pass filtered image of Figure 3.5 (a).

3.1.2 Spatial Domain Filtering

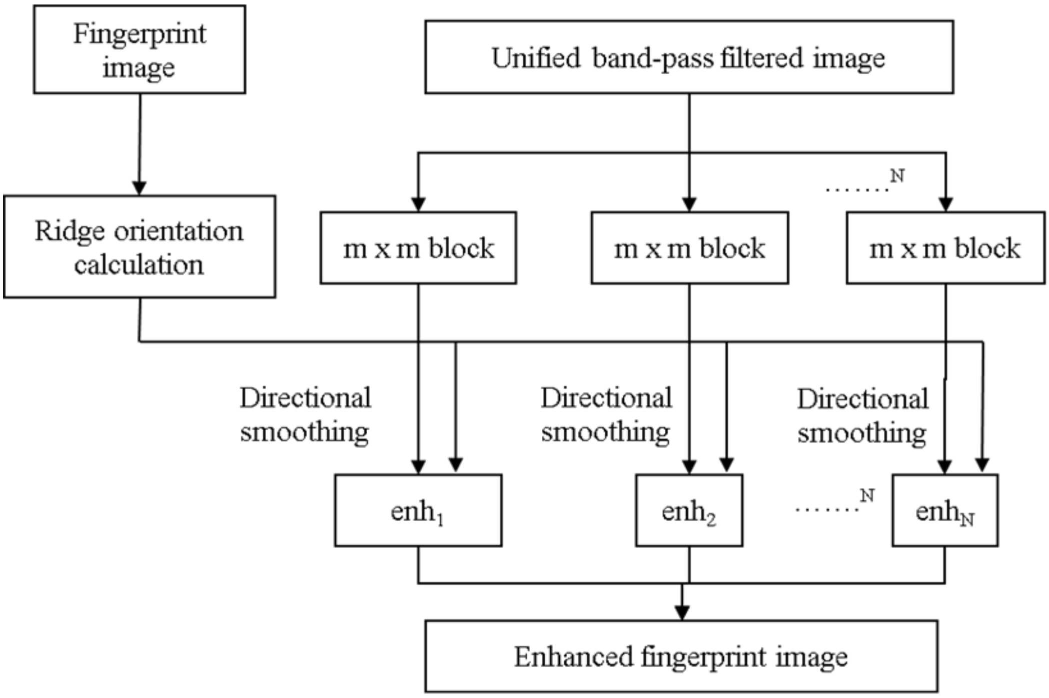


Figure 3.6: Algorithmic steps of spatial domain enhancement

Following the frequency domain filtering, the problem of broken ridges and scars, it can be resolved in spatial domain by smoothing ridges along ridge directions. Figure 3.6 shows the basic algorithmic steps of spatial domain enhancement which are as:

- Compute ridge orientation map in spatial domain.
- Divide unified band-pass filtered image into sub-blocks and apply local directional smoothing on each block.
- Combine all the directionally enhanced sub-blocks to form an enhanced fingerprint image

3.1.2.1 Directional Smoothing

Directional smoothing is performed to enhance the ridge flow structure in the direction of ridges. For this purpose an orientation map is computed that gives dominant orientation of each pixel in the direction of ridges. An orientation image $O(x, y)$ is defined as an M x N image that represents the local ridge orientation at pixel (x, y). Approaches used for estimating local ridge orientations in spatial domain are based on computation of gradients in the fingerprint image. In the proposed study, the algorithm presented by Ratha et al. [27] is used for estimating ridge orientations. The steps involved in directional smoothing are as follows:

- Extract sub-blocks of size L x L from unified band-pass filtered image.

$$\begin{aligned} &f_{s,t}(x,y) = f(x,y) \quad (s-1/2)L < x < (s+1/2)L, (t-1/2)L < y < (t+1/2)L, \\ &f_{s,t}(x,y) = 0 \qquad\qquad \textit{otherwise}, \end{aligned} \tag{3.13}$$

$$\textit{where} \quad 1 < s < M/(L), 1 < t < N/(L). \tag{3.14}$$

- Rotate the sub-block by an angle $-\varphi+90$, where φ is the dominant orientation of sub-block given by the value of $O(x,y)$ at the center of the sub-block. After rotation the ridges in the sub-block become vertical.

$$J_{s,t}(x,y) = f_{s,t}(x_\varphi, y_\varphi), \tag{3.15}$$

$$x_\varphi = x\cos(90-\varphi) + y\sin(90-\varphi), \tag{3.16}$$

$$y_\varphi = -x\sin(90-\varphi) + y\cos(90-\varphi). \tag{3.17}$$

- Apply a 1-D vertical Gaussian smoothing filter on the sub-block.

$$enh_{s,t}(x,y) = J_{s,t}(x,y) * h(y). \tag{3.18}$$

Where $h(y)$ is the 1-D Gaussian smoothing filter for vertical ridges as discussed in section 3.2 and s, t is the center of block in horizontal and vertical directions respectively.

- Directionally smoothed sub-block is then rotated back. Mathematically,

$$enh_{s,t}(x, y) = enh_{s,t}(x_{\varphi}, y_{\varphi}), \tag{3.19}$$

$$x_{\varphi} = x \cos(\varphi \text{ -90}) + y \sin(\varphi \text{ -90}), \tag{3.20}$$

$$y_{\varphi} = -x \sin(\varphi \text{ -90}) + y \cos(\varphi \text{ -90}). \tag{3.21}$$

- Repeat this process on all sub-blocks of unified band-pass filtered image.
- Finally all the processed sub-blocks are tiled together to form an enhanced image.

$$f\ (x, y) = enh_{s,t}(x, y) \quad (s-1/2)L < x < (s+1/2)L, (t-1/2)L < y < (t+1/2)L. \tag{3.22}$$

Figure 3.7 shows the results of proposed enhancement scheme including band-pass filtering and spatial filtering. Figure 3.7 (a) shows the fingerprint image ‘14_7.tif’ taken from FVC 2002 DB1-A [53], and Figure 3.7 (b) shows the unified band-pass filtered image of Figure 3.7 (a). The enhanced image after applying directional smoothing on unified band-pass filtered image is shown in Figure 3.7 (c). It can be observed that the proposed algorithm effectively enhances low contrast regions and also improves the ridge structure by joining the broken ridges.

It can be seen in Figure 3.7 (c) that there are some small regions where the ridges have been blurred. This is an inherent limitation of Contextual filtering based on fixed window size. This effect can be overcome by using adaptive window size during directional smoothing on unified band-pass filtered image. For instance, in high curvature blurred regions, smaller window size in directional smoothing stage will give better results. Therefore, adaptive window size based on variance of local orientations can be used to further improve the results.

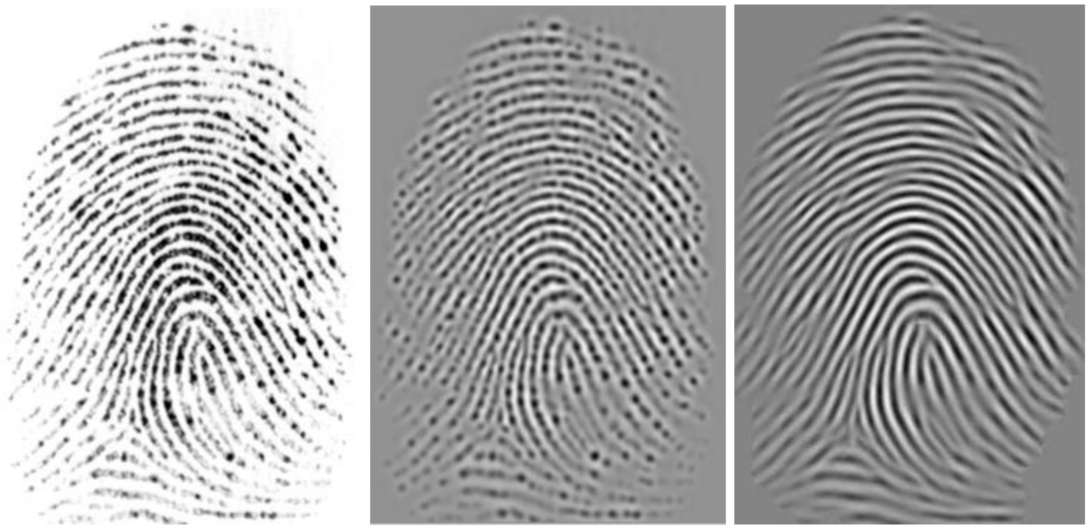

(a) (b) (c)

Figure 3.7: Enhanced fingerprint images after proposed directional smoothing method, (a) Fingerprint image 14_7.tif, (b) Unified band-pass enhanced image, (c) Enhanced images after directional smoothing

3.2 Frequency Response of Proposed Filters

The proposed enhancement technique as demonstrated in sections 3.1.1 and 3.1.2, involves first the band-pass filtering in frequency domain and then directional smoothing in spatial domain. It will be interesting to study the combined filter response and to compare it with variants of Gabor filter. The band-pass filter that is applied in frequency domain is defined in equation (3.3). The description of directional 1-D Gaussian filter applied in spatial domain is as follows.

A 2-D Gaussian filter [54], oriented at an angle θ can be expressed as equation (3.23), and is shown in Figure 3.8 (a),

.

$$h(x,y) = \exp\left[-\frac{1}{2}\left(\frac{x_\theta{}^2}{\sigma_x^2} + \frac{y_\theta{}^2}{\sigma_y^2}\right)\right], \tag{3.23}$$

$$x_\theta = x\cos\theta + y\sin\theta, \tag{3.24}$$

$$y_\theta = -x\sin\theta + y\cos\theta. \tag{3.25}$$

Where σ_x is the standard deviation of Gaussian spread along x_θ-axis and σ_y is the standard deviation of Gaussian spread along y_θ-axis.

A 1-D Gaussian filter oriented at an angle θ can be obtained by taking $\sigma_x \rightarrow 0$, as shown in Figure 3.8 (b). This is because x_θ-axis dependency becomes an impulse function for $\sigma_x \rightarrow 0$.

To show the response of the filter without losing generality, it can be assumed that the ridge direction of a sub-block is vertically upwards. In that case 1-D vertical Gaussian filter is shown in Figure 3.8 (c) and is given by equation (3.26).

$$h(y) = \exp\left[-\frac{1}{2}\left(\frac{y^2}{\sigma_y^2}\right)\right], \quad \text{(for } x_\theta = 0\text{)}, \qquad (3.26)$$

For $x_\theta \neq 0$, $h(y)$ becomes zero.

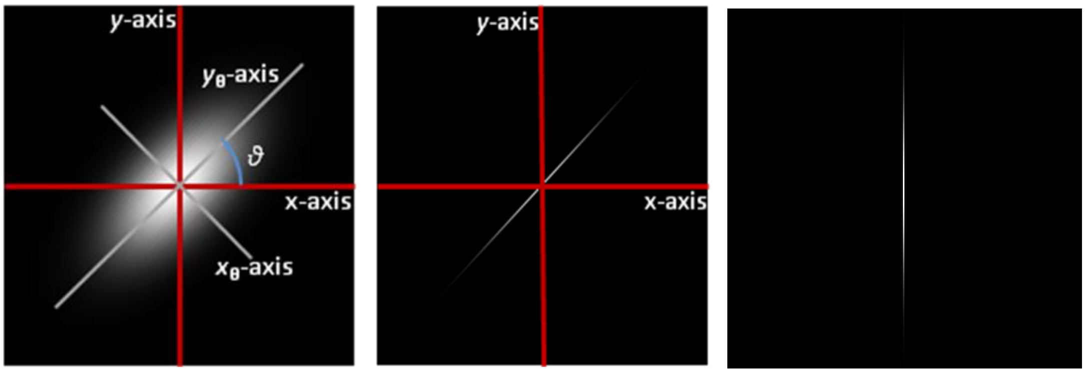
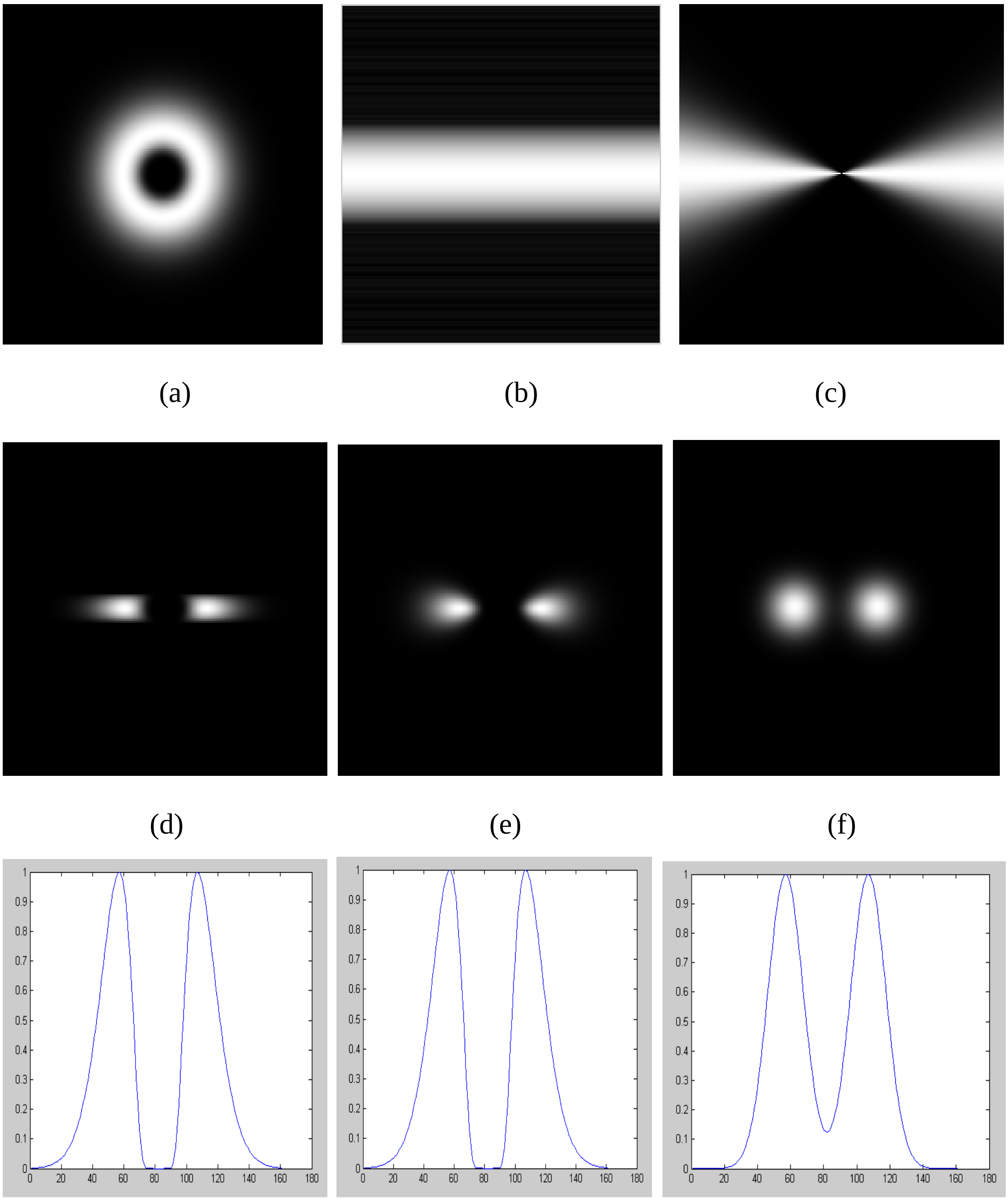


Figure 3.8: Proposed Gaussian filter, (a) 2-D Gaussian filter oriented at angle θ, (b) 1-D Gaussian Filter oriented at angle θ, (c) 1-D Gaussian filter oriented at angle 90^0 for vertical ridges

In Figure 3.9, a comparison of different directional band-pass filters for ridges with vertically upward direction is shown. A band-pass filter with D_0 = 25 and W = 30 is shown in Figure 3.9(a). Frequency response of the proposed 1-D vertical Gaussian filter shown previously in Figure 3.9(c), is shown in Figure 3.9 (b). A 2-D log fan-filter for vertically upward ridges is shown in Figure 3.9 (c). The proposed directional band-pass filter obtained after multiplying frequency response of vertical 1-D Gaussian filter (b) with band-pass filter (a) is shown in Figure 3.9 (d). A 2-D Log-Gabor filter obtained after multiplying directional log fan-filter (c) with band-pass filter (a) as proposed by Wang et al [32] is shown in Figure 3.9 (e). A Traditional 2-D Gabor filter proposed by Hong et al [30] for vertically upward ridges is shown in Figure 3.9 (f).



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 3.9 : Frequency response of the proposed Gaussian filter and its comparison with other contextual filters, (a) Band-pass filter, (b) Frequency response of the proposed 1-D Gaussian filter, (c) 2-D log fan-filter, (d) proposed directional band-pass filter, (e) 2-D Log-Gabor filter, (f) Traditional 2-D Gabor filter, (g) 1-D Frequency response of the proposed directional filter, (h) 1-D Frequency response of Log-Gabor, (i) 1-D Frequency response of Traditional Gabor Filter

In order to compare the response of directional band-pass filters shown in Figure 3.9 (d), (e) and (f) a 1-D vector from center of each Figure is extracted and their plots are shown in Figure 3.9 (g), (h) and (i) respectively. From the plot of Figure 3.9 (i), it is obvious that a DC component is present in the frequency response of Traditional Gabor filter [30] that will degrade the performance during enhancement process. Plots in Figure 3.9 (g) and Figure 3.9 (h) shows the 1-D frequency response of the proposed directional band-pass filter and Log-Gabor directional filter [32], respectively. From the plots it can be concluded that both the responses are identical in the sense that they do not have any DC component. Moreover, the proposed filter has an extended tail at higher frequencies just like log-Gabor filter, which ensures that sudden terminations (endings) and bifurcations of the ridges are preserved.

3.3 Computational cost analysis and comparison

In this section, computational complexity of the proposed two-fold enhancement algorithms is compared with other well-known enhancement techniques. The enhancement techniques considered while analysis are Hong's Gabor filtering [30], Chikkerur's STFT [31] method and Wang's Log-Gabor [32] based enhancement technique. The analysis is performed by computing the computational cost in terms of total number of multiplications (O^{x}), additions (O^{+}) and load operations (O^{l}) required to enhance a fingerprint image. The analysis is performed by assuming that the size of the fingerprint image is M x M, while the size of each partitioned block to be enhanced is $m\times m$ pixels for all the enhancement techniques. In this study, the computational complexity of frequency estimation and filtering algorithms involved in each enhancement technique is considered while analysis. The detailed explanation of computational complexity for each enhancement technique is given in Appendix B, while Table 1 summarizes the computational complexity. Table 2 states the computational complexity in terms of number of multiplications (O^{x}), additions (O^{+}) and load operations (O^{l}) for different values of M and m that is the size of the image and the size of the block, respectively. For the proposed two-fold enhancement, number of band-pass filters required in frequency domain filtering is $n=5$, while the length of 1-D Gaussian filter required for spatial domain enhancement is $g=7$.

Table 1: An analysis of computational complexity for different well-known contextual filtering based enhancement techniques

	Multiplications: (O^x)	**Additions: (O^+)**	**Load /Memory operations: (O^l)**
Hong's Gabor based enhancement	$M^2m^2 + 4M^2(1+1/m(4m+1/2\log_2(m)+1))$	$M^2m^2 + 2M^2(1+2/m(3m+\log_2(m)+2))$	m^2M^2
Chikkerur's STFT based enhancement	$4M^2(8+1/2\log_2(m^2)) + 2M^2(2+\log_2(m^2))$	$4M^2(8+\log_2(m^2)) + 4M^2\log_2(m^2)$	$4M^2$
Wang's Log-Gabor Enhancement	$4M^2(1+1/m(4m+1/2\log_2(m)+1)) + 4M^2(1+\log_2(m^2))$	$2M^2(3+2/m(3m+\log_2(m)+2)) + 8M^2\log_2(m^2)$	$4M^2$
Proposed Two-fold Enhancement Technique	$(M^2)/2\log_2(M^2) + n(M^2) + n(M^2)/2\log_2(M^2) + M^2/m^2(4g) + M^2/m(g+m)$	$(M^2)\log_2(M^2) + n(M^2)\log_2(M^2) + 4M^2 + M^2/m^2(2g) + M^2/m(g+m)$	$n * M^2$

Table 2: An analysis of computational complexity for different values of image sizes (*M*) and block sizes (*m*)

	Hong's Gabor based enhancement	**Chikkerur's STFT based enhancement**	**Wang's Log-Gabor Enhancement**	**Proposed Two-fold Enhancement Technique**
	M=256, m=16			
Multiplications	17874944	4456448	3457024	3574784
Additions	17793024	6291456	5472256	6651392
Load Operations	16777216	262144	262144	327680
	M=512, m=16			
Multiplications	71499776	17825792	13828096	15872000
Additions	71172096	25165824	21889024	29751296
Load Operations	67108864	1048576	1048576	1310720

3.4 Experimental Results

In this section, a detailed performance evaluation has been carried out by comparing major fingerprint enhancement schemes available in literature [30, 31, 32] with the proposed enhancement technique by calculating equal error rates (EER) and comparing detection error tradeoff (DET) graph. In fingerprint recognition systems, the enhancement algorithms are not only evaluated according to their visual perception, but more importantly, by their suitability for fingerprint matching. Therefore, the proposed technique is tested here by evaluating both visually and by error rates on known databases using known matching algorithms.

3.4.1 Databases and Test Protocol

In order to carry out experimentations publically available databases FVC2002 DB1-A, FVC2002 DB2-A [53] and FVC2004 DB1-A [55] are used. For performance evaluation, the FVC2002 [53] testing protocol has been adopted. In the above mentioned FVC databases [53], [55], the total number of subjects is hundred (100) and each subject has eight (8) fingerprints.

In order to evaluate performance of enhancement schemes, performance indicators like equal error rates (EER) and detection error tradeoff (DET) graph are considered.

EER is measured in terms of False Matching Rate (FMR) and False Non-Matching Rate (FNMR).

In order to obtain genuine matching or False Matching Rate (FMR), each sample of a finger is matched against the remaining samples of that finger. Therefore, the total number of genuine matches for a database with hundred (100) fingers and eight (8) samples per finger can be calculated as:

((8*7) /2) * 100 = 2,800.

In order to obtain imposter matching or False Non-Matching Rate (FNMR), the first sample of each finger is matched against the first sample of the remaining fingers in the database. Therefore, the total number of imposter or FNMR matches for a database with hundred (100) fingers can be calculated as:

((100*99) /2) = 4,950.

EER is the error rate given by either False Matching Rate (FMR) or False Non-Matching Rate (FNMR), at a threshold level *t* where both FMR and FNMR are equal as shown in Figure 3.10 [3]. EER is a single error parameter that is accepted worldwide to solely represent the system's performance independent of threshold selection.

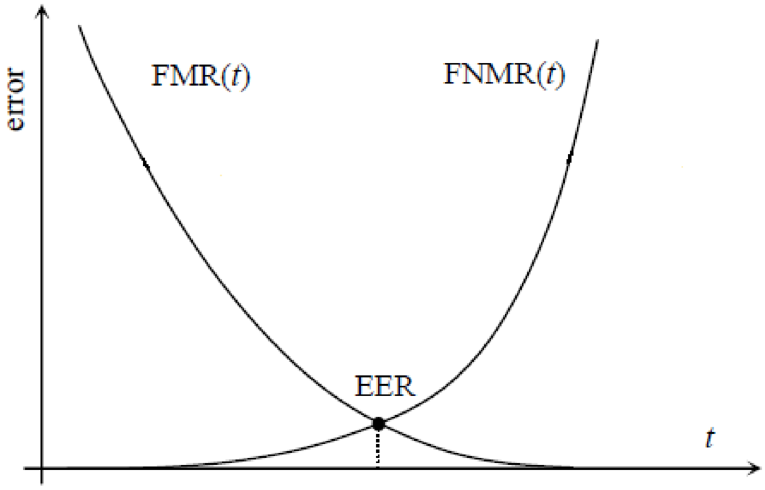


Figure 3.10: An example of FMR(t) and FNMR(t) curves, EER is highlighted at the threshold t where the FMR(t) and FNMR(t) are equal

3.4.2 Matching Algorithm

The matching algorithms used for performance evaluation is proposed by Chikkerur et al. [41]. The source code of this matching algorithm is available at: http://sites.google.com/site/sharatchikkerur/code. This technique employs a k-plet technique to encode each minutia based on its local neighborhood. Matching is then performed based on a dual graph traversal (coupled BFS) algorithm.

3.4.3 Performance Comparison with other enhancement techniques

The proposed enhancement scheme is compared with enhancement performed by Hong's Gabor filtering [30], Chikkerur's STFT method [31] and Wang's Log-Gabor Filtering based enhancement technique [32]. The Matlab implementation of Hong's Gabor enhancement [30] is on-line available at: http://www.csse.uwa.edu.au/~pk/Research/MatlabFns/. The Matlab implementation of Chikkerur's STFT method [31] is on-line available at: http://sites.google.com/site/sharatchikkerur/code. In Figure 3.11 and Figure 3.12, a comparison of enhancement performed by our proposed enhancement technique with other enhancement techniques has been presented. Figure 3.11 (a) is the '14_7.tif' fingerprint image taken from FVC 2002 DB1-A [53]. Figure 3.11 (b) is the output of enhancement performed by our proposed enhancement technique. Figure 3.11 (c), (d) and (e) are outputs of enhancement performed by Hong's Gabor [30], Chikkerur's STFT [31] and Wang's Log-Gabor [32] based enhancements respectively.

Similarly, Figure 3.12 shows output of different enhancement techniques on fingerprint image '33_5.tif' taken from FVC 2002 DB2-A [53]. From the results (Figure 3.10 (c), (e) and Figure 3.12(c), (e)), it can be observed that Hong's Gabor [30] and Wang's Log Gabor [32] based enhancement does not properly enhances fingerprint image in the regions of low contrast and broken ridges. This is mainly because of false frequency estimations in these regions. The enhancement based on Chikkerur's STFT [31] method, produces blocking effects as can be seen in Figure 3.11 (d) and Figure 3.12 (d). From the results of Figure 3.11 (b) and Figure 3.12 (b), it is can be observed, that the proposed enhancement is more effective in enhancing distorted images even with regions having low contrast and broken ridges.

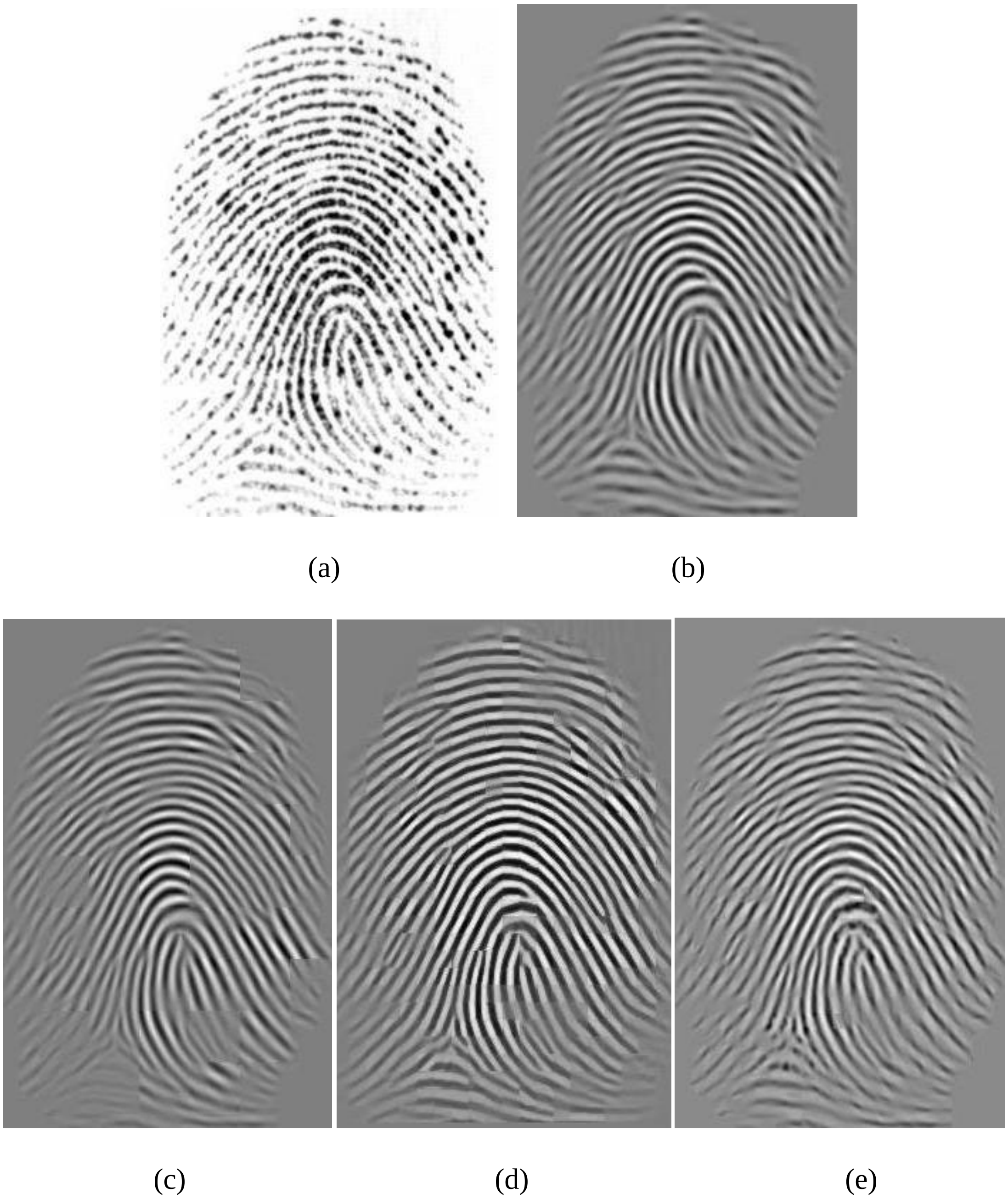
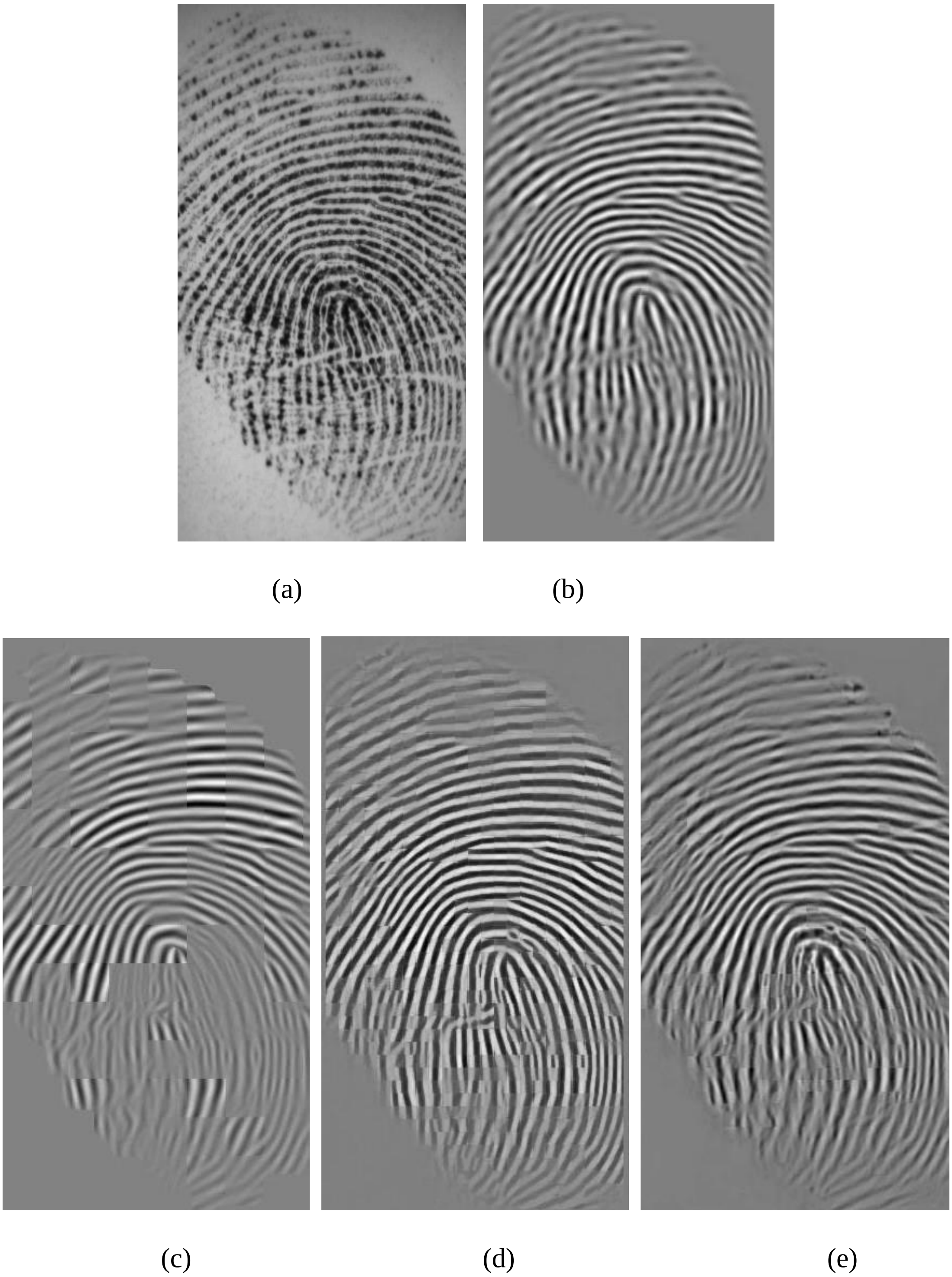


Figure 3.11: Example1: application of proposed two-fold enhancement on fingerprint image and its comparison with other enhancement techniques, (a) Fingerprint image 14_7.tif from FVC 2002 DB1-A [53], (b) Enhanced image after the proposed enhancement technique, (c) Enhanced images after applying Hong's Gabor based enhancement [30], (d) Enhanced images after applying Chikkerur's STFT based enhancement [31], (e) Enhanced images after applying Wang's Log Gabor based enhancement [32].



(a) (b)

(c) (d) (e)

Figure 3.12: Example 2: application of proposed two-fold enhancement on fingerprint image and its comparison with other enhancement techniques, (a) Fingerprint image 33_5.tif from FVC 2002 DB2-A [53], (b) Enhanced image after the proposed enhancement technique, (c) Enhanced images after applying Hong's Gabor based enhancement [30], (d) Enhanced images after applying Chikkerur's STFT based enhancement [31], (e) Enhanced images after applying Wang's Log Gabor based enhancement [32].

In order to evaluate performance of enhancement schemes, performance indicators like equal error rates (EER) and detection error tradeoff (DET) graph are considered. EER is measured in terms of False Matching Rate (FMR) and False Non-Matching Rate (FNMR) as discussed in section 3.4.1.

As explained above, matching algorithm proposed by Chikkerur et al. [41] is used for all enhancement schemes. To extract minutiae points, the enhanced images are binarized followed by thinning process and minutiae points are then extracted [39], for all enhancement techniques. Table 3, reports the error rates of all enhancement techniques for different FVC databases [53, 55]. The EER obtained on FVC 2004 DB1-A [55] is much higher as compared to EER on the other two databases considered. This is because FVC 2004 DB1-A [55] database contains harder cases including distorted and partial fingerprint images. However, it is worth mentioning that the proposed enhancement technique outperforms other enhancement techniques consistently for all the three databases.

Table 3: Summary of comparative analysis of the proposed two-fold enhancement with other well-known techniques based on EER

	EER on FVC Databases		
	FVC2002-DB1-A [53]	FVC2002-DB2-A [53]	FVC2004-DB1-A [55]
Hong's Gabor Enhancement [30]	2.3%	2.06%	17.09%
Chikkerur's STFT Enhancement [31]	1.95%	2.52%	18.4%
Wang's LogGabor Enhancement [32]	1.85%	1.98%	15.36%
Proposed Enhancement	**1.24%**	**1.63%**	**10.84%**

Another performance indicator considered here is DET graph. DET graph is a variation of receiver operating characteristic (ROC) curve. In ROC curve, FMR is plotted against True Matching Rate (TMR) or verification rate. On the other hand, in DET graph, FMR is plotted against FNMR. Therefore, both the axes of DET graph represents error rates and are generally plotted using logarithmic scales. The DET graph for each database using different enhancement schemes is shown separately in Figure 3.13 ((a), (b), and (c)). From these results, it can be observed, that the proposed fingerprint enhancement scheme enhances the fingerprint images reliably and accurately.

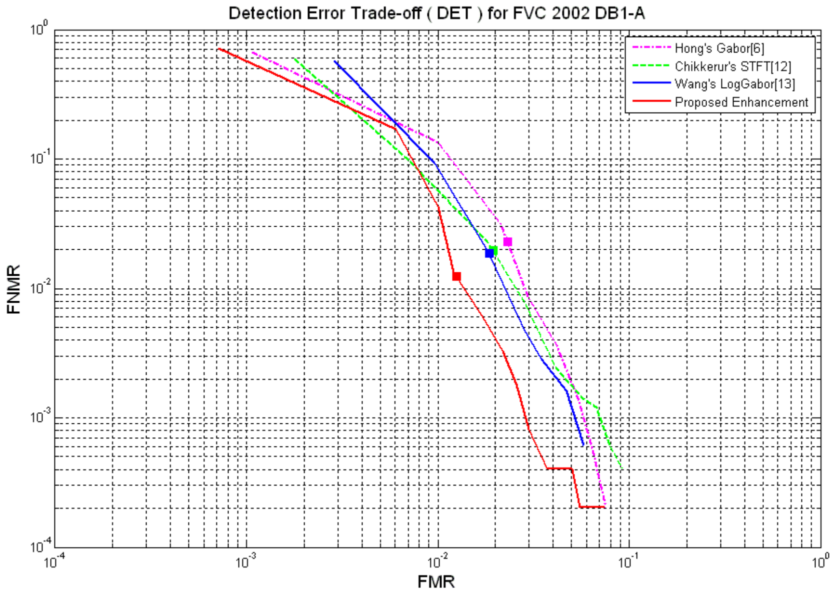


Figure 3.13: DET graph of different enhancement schemes, (a) DET graph of different enhancement schemes on FVC 2002 DB1-A database

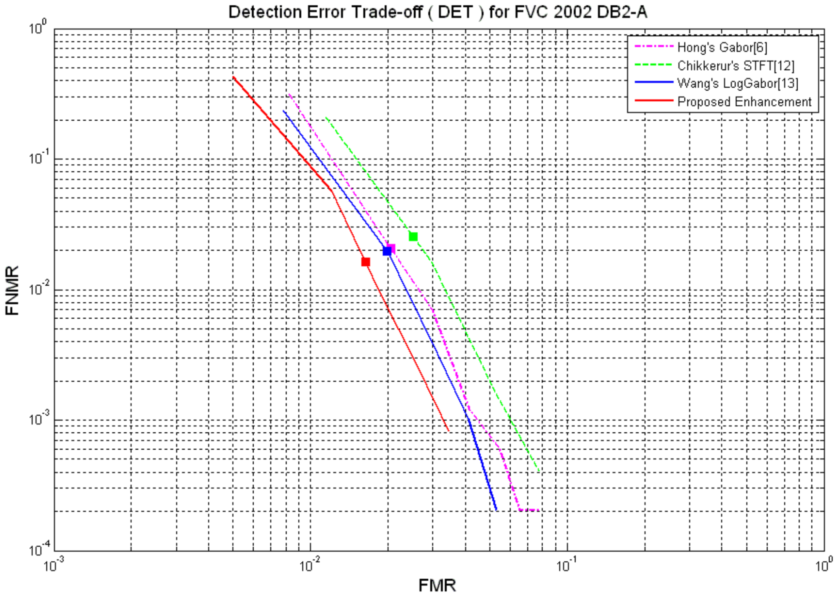


Figure 3.13: (b) DET graph of different enhancement schemes on FVC 2002 DB2-A database

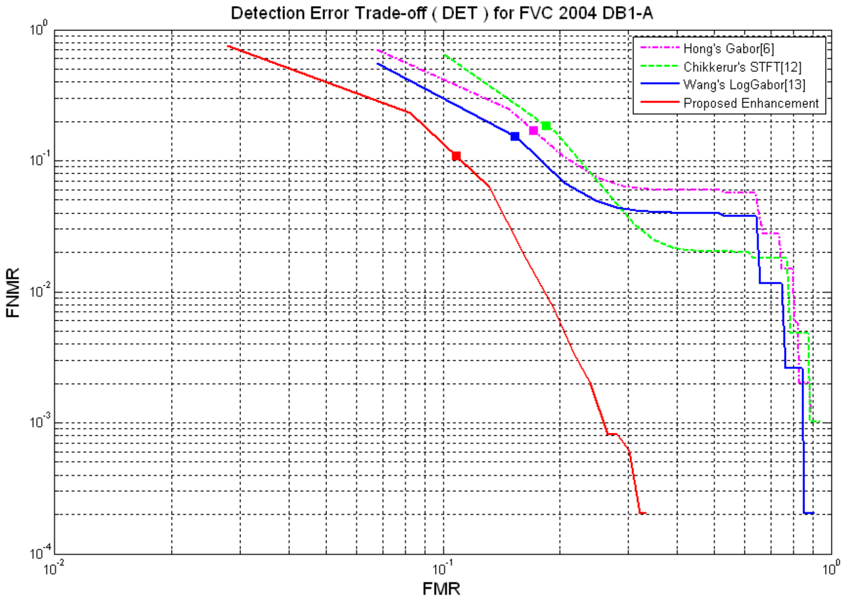


Figure 3.13: (c) DET graph of different enhancement schemes on FVC 2004 DB1-A database

3.5 Summary

An effective fingerprint enhancement technique based on directional contextual filtering has been proposed. The enhancement technique is novel in the sense that it divides the tasks of band pass filtering and directional filtering into frequency and spatial domain respectively, exploiting the strengths of both domains. Two major advantages of the proposed enhancement technique are; avoiding the hazards of frequency estimation errors and making spatial domain filtering quite simple by using a global 1D Gaussian smoothing filter. The enhanced images, as shown in the obtained results, are of good quality having smooth ridge/valley structure. For performance evaluation of the proposed enhancement an extensive testing has been carried out by using three well-known enhancement approaches and three databases. Experimental results demonstrate that proposed enhancement outperforms other well-known enhancement techniques, i.e. Hong's Gabor filtering [30], and Chikkerur's STFT [31] and Wang's Log-Gabor filtering [32].

Chapter 4

ENHANCEMENT AND FREQUENCY DISTORTION REMOVAL

4.1 Non-Uniform Inter-Ridge Separation Distortion

In practice, a fingerprint image suffers from several different kinds of degradations. These degradations include non-uniform illumination, scars or broken ridges due to humidity and imperfect sensors, and non-uniform ridge separations which occurs due to non-uniform contact with sensor. Although contextual filtering techniques work well to enhance the fingerprints ridge structure, but these techniques do not rectify the non-uniform inter-ridge separation which may result in false fingerprint matching. These distortions result in faulty feature extraction thus adversely affecting the recognition performance by complicating the feature matching process and making it time consuming and error prone. The importance of removal of non-uniformity in inter-ridge separation is widely highlighted in the earlier studies. Most of fingerprint recognition systems handle this problem either at image acquisition stage by using additional hardware [56, 57] or by using complicated feature matching algorithms [12, 58, 59, 60]. In [14], a canonical form of image was used for inter-ridge distance normalization. However, this technique can only be applied to high quality thinned binary images. The distorted image is by definition a 2-D dynamic function with varying time frequency plot due to non-uniform ridge separations. Figure 4.1, shows two samples of fingerprint images of the same person, but they appear to be different because of non-uniform inter-ridge separation.

In this study, a novel algorithm is proposed based on local adaptive directional frequency enhancement for noise removal and inter-ridge distance normalization using local short time Fourier transform (STFT) analysis. In the proposed approach, STFT analysis of fingerprint is performed by dividing image into non-overlapping blocks of size *m* x *m*. Where *m* is set to *32*. Then Fourier transform of each block is computed as was done by Chikkerur et al. [31]. The proposed approach is different in the sense that the problem of non-uniform inter-ridge frequency is solved by a novel frequency normalization and nonlinear tiling approach in spatial domain [16].

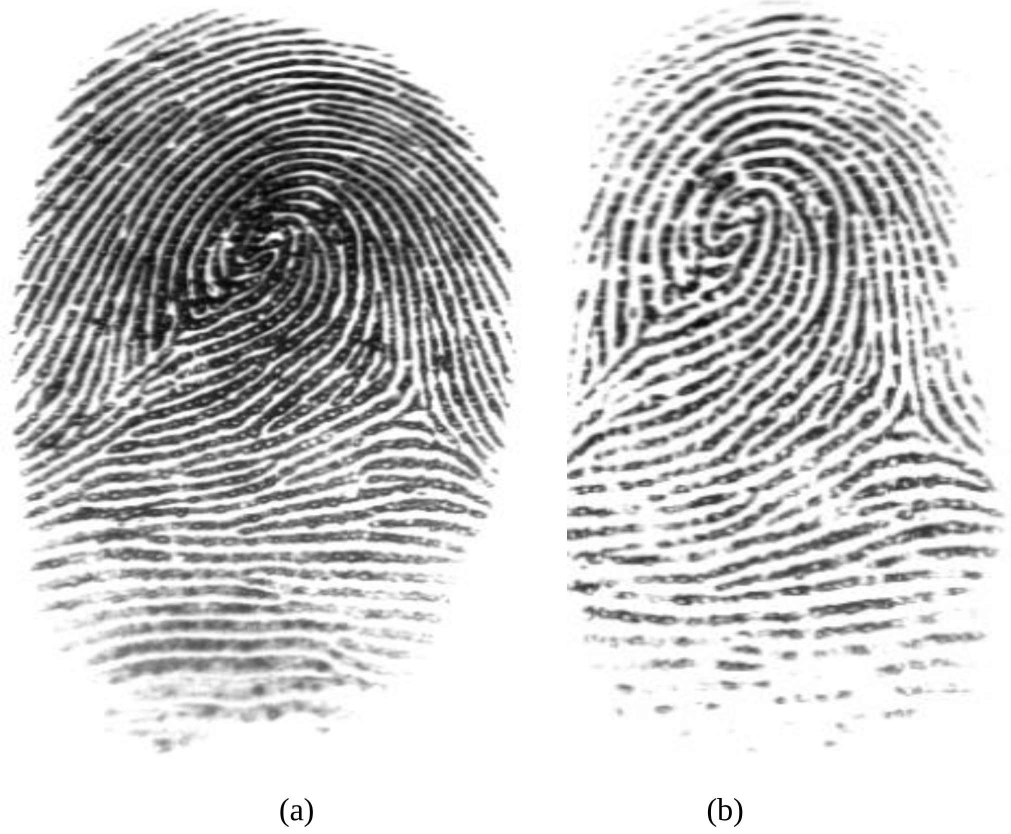


Figure 4.1: Fingerprint images showing the problem of non-uniform inter-ridge separation, (a) and (b) Two samples of fingerprint images of same person

The contents of the rest of the chapter are organized as follows. Section 4.2 formulates our approach by first establishing fingerprint as a dynamic 2-D signal using STFT analysis. The basic filtering approach and frequency normalization approach are then explained. Section 4.3 explains proposed algorithm in detail and section 4.4 presents results of the proposed enhancement technique.

4.2 Problem Formulation

4.2.1 Fingerprint as a non-stationary 2-D signal

An ideal fingerprint image can be imagined as a 2D stationary signal composed of multiple sinusoidal wave patterns of different orientations with approximately same frequency. However, real fingerprint images tend to be non-stationary in nature because of non-uniform inter-ridge distances, this can be observed from Figure 4.2, which shows a fingerprint (Figure 4.2 (a)) with its Fourier transform (Figure 4.2 (b)) and spatial frequency plot shown as gray level image (Figure 4.2 (c)).

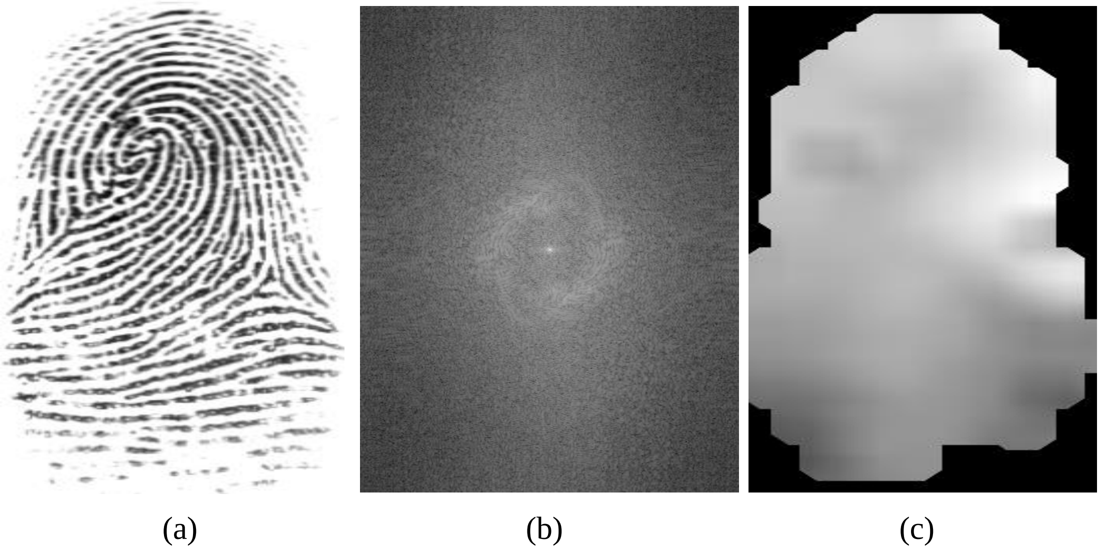


Figure 4.2: Fingerprint image depicting non-stationary nature, a) Fingerprint Image, b) Fourier Transform, c) Frequency Map

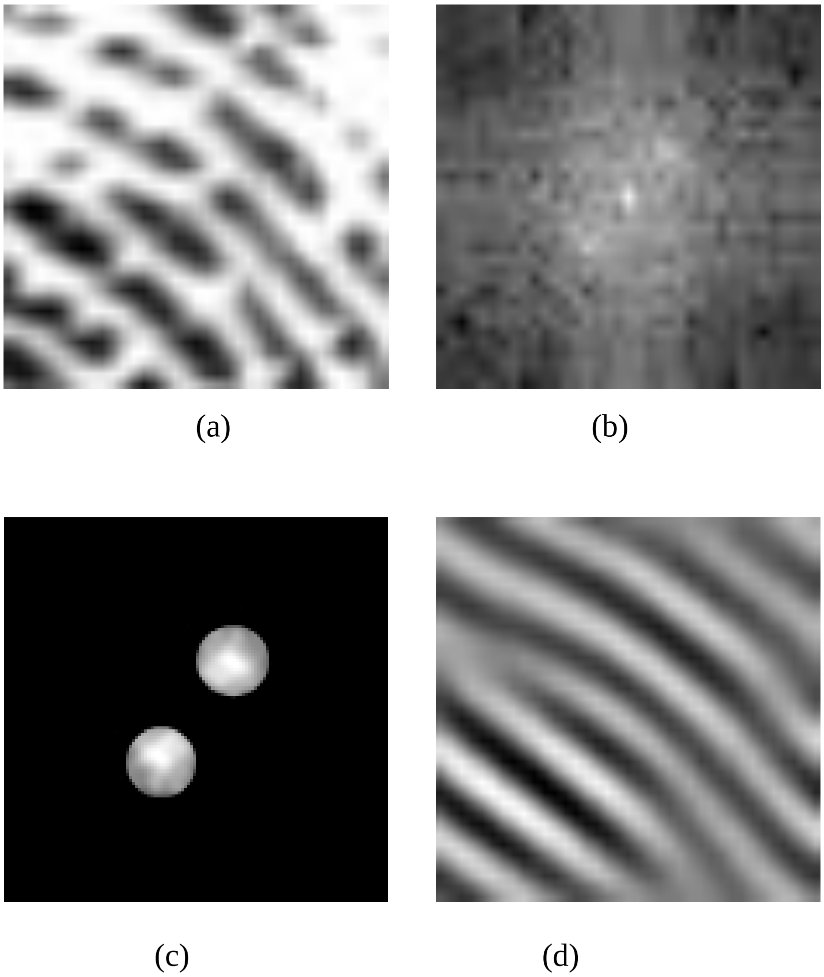
From the Fourier transform of Figure 4.2 (b) it can be seen that a band of frequencies corresponding to ridge frequencies in the fingerprint is clearly dominant and this band should be preserved in an enhancement technique. Frequencies lower than this dominant band correspond to background intensities and slow intensity variations in the fingerprint, while higher frequencies correspond mostly to noise or abrupt intensity variations. However, from the Fourier transform which is a global representation, it is not clear whether the local frequencies are constant throughout the whole image or these are changing.

In the frequency domain plot shown in Figure 4.2 (c), the local frequency of each pixel in fingerprint image is calculated along a line orthogonal to ridge direction and shown against spatial coordinates. It may be mentioned that time-frequency plot of a 2-D signal is a 4-D representation involving 2 frequency components for each spatial location (*x*,*y*). However, due to the nature of a fingerprint image that comprises ridge patterns, it is logical to show only the dominant frequency that is calculated in the direction normal to local ridge orientation. Therefore, this 3-D Spatial-frequency representation of a 2-D image is analogous to 2-D time-frequency representation of a

1-D signal. It is clear from frequency map of the fingerprint that the local frequencies shown by different gray level in Figure 4.2 (c) are not constant throughout the fingerprint region. Hence, fingerprint image can be considered as an orientated 2D dynamic or non-stationary signal. Moreover, since the dominant frequencies are not constant throughout, any enhancement approach assuming constant frequency will not be beneficial.

4.2.2 Filtering In STFT

For non-stationary signal analysis, Short time Fourier transforms (STFT) has been used as an effective analysis tool. An example of filtering using STFT is shown in Figure 4.3. Figure 4.3 (a) shows a scaled sub-image of size 36 x 36 pixels taken from the fingerprint shown in Figure 4.2 (a), and Figure 4.3 (b) shows its Fourier transform.


(a) (b)

(c) (d)

Figure 4.3: Enhancing a sub-image of fingerprint in frequency domain (a) A scaled block of fingerprint, (b) Fourier Transform of sub-image, (c) Dominant Frequency Band, (d) Enhanced sub-image

It can be seen in the transform that there is a dominant sinusoidal component given by two symmetric peaks on either side of the center in a region orthogonal to the dominant ridge orientation of the block. This dominant frequency region corresponds to the desired ridge pattern and the remaining frequencies are unwanted frequencies corresponding to noise and average illumination. In order to enhance the sub-image, this dominant frequency region should be boosted compared to the other frequencies. This can be done by filtering out the undesired frequencies using notch-pass filters. A demonstration of such notch-pass filtering is shown in Figure 4.3 (c) where the filtered Fourier transform image is shown by using approximate notches. Figure 4.3 (d) shows the inverse transformed image in spatial domain which has been enhanced as desired.

4.2.3 Frequency Normalization

For non-stationary signal, inter-ridge frequency varies throughout the signal. In fingerprint image this variation is gradual and assumption of slowly varying frequency is logical. To normalize the frequency distribution in fingerprint, STFT becomes very useful as it divides image into small overlapping blocks and each block is enhanced individually assuming that the frequency within a block is approximately constant. However, frequency variation is quite significant throughout the fingerprint and it is desirable to bring the fingerprint to a nearly constant frequency for favorable fingerprint matching.

In order to clarify the procedure of frequency normalization, consider two different blocks of fingerprint image having significantly different dominant frequencies. Both blocks are first enhanced using the filters corresponding to their individual dominant bands and orientations. In order to bring two blocks on the same frequency, the frequency domain image can be scaled around the center to align the peaks in Fourier transforms of two sub-images. Using the scaling property of Fourier transform, an alternative simpler procedure for such frequency normalization can be inverse scaling in spatial domain. For example, if frequency of the first block is higher than the other, it means that the inter-ridge separation of first block is smaller than the other. In this case normalization can be achieved by either compressing the first block with respect to the second in frequency domain, or by stretching the first block with respect to second in spatial domain.

4.3 Proposed Algorithm

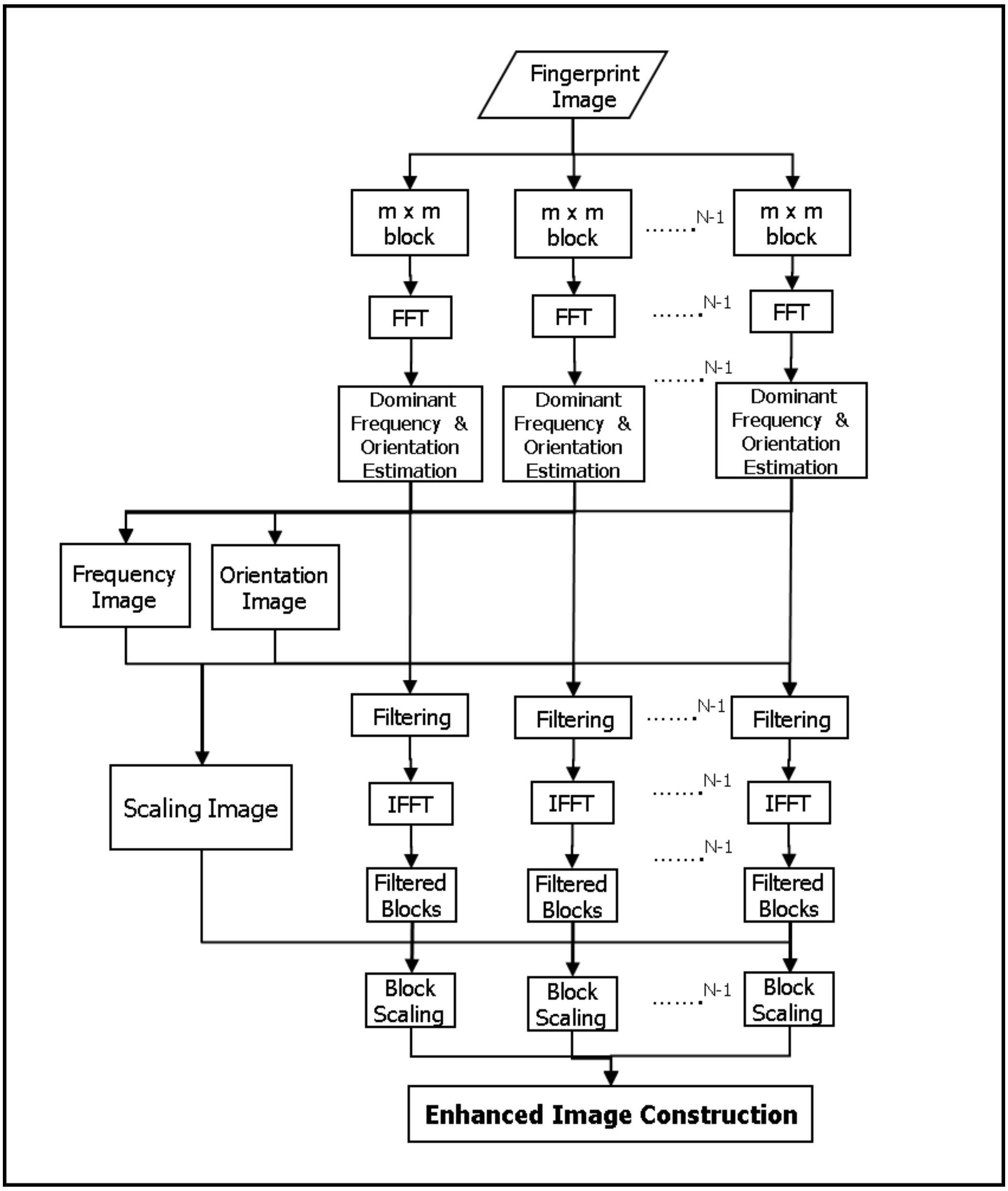


Figure 4.4 Block Diagram of proposed enhancement technique

The overall algorithm flow for the proposed STFT based fingerprint enhancement is shown in Figure 4.4. As shown in the figure, the main components of the proposed algorithm include firstly the division of image into *N* sub-images. Each sub-image is then converted into frequency domain through FFT and dominant band and orientation is estimated. Each sub image is enhanced by using directional band-pass filtering. Each sub-image is then converted back to spatial domain through IFFT and scaling is performed in spatial domain. Finally, sub-images are recombined to obtain full enhanced fingerprint image. A detailed explanation of each component is given in the following subsections.

4.3.1 STFT Computation

Image is first divided into *N* overlapping blocks $I_{a,b}(x,y)$ of 2*m* x 2*m* size, where *a* and *b* are the block indices in *x* and *y* directions respectively. These blocks are larger than the actual blocks, which are to be tiled together later for recombination which are of size *m* x *m* with the same centers and non-overlapping. The larger non-overlapping blocks are more suitable for dominant band estimation and also for non-linear tiling and interpolation in the later stage of recombination. Each block is then converted to its frequency domain representation $F_{a,b}(u,v)$ using FFT.

4.3.2 Frequency and Orientation Map Construction

Normalized frequency and orientation maps for each block $F_{a,b}(u,v)$ are constructed by using dominant band estimation technique described as follows:

1. For each block *a* and *b*,

   a. Firstly, *i* x *j* Gaussian window functions are computed for *i* different frequencies and *j* orientations. Where *i* ranges from $D_{0min}<i< D_{0max}$ and *j* ranges from $0<j<\pi$. The Gaussian window function in frequency domain resembles a directional band-pass filter and is defined by multiplication of two filters.

$$H_{ij}(u,v) = H_{b_i}(u,v) H_{\theta_j}(u,v), \tag{4.1}$$

where $H_{b_i}(u,v)$, is the Gaussian band selection function superimposed by a directional Gaussian function $H_{\theta_j}(u,v)$

The Gaussian band selection function is defined as,

$$H_{b_i}(u,v) = \exp\left[-\frac{1}{2}\left(\frac{D^2(u,v) - D_{0_i}^2}{D(u,v)W}\right)^2\right], \tag{4.2}$$

where, $D(u,v)$ is the distance of each pixel from the center. D_{0_i} is the radial distance of i^{th} band center from the center of the FT image W is the width of the function, it defines the Gaussian spread in both directions. Figure 4.5 (a) shows the 2-D plot of the band selection function.

The directional window function is defined by the following equation:

$$H_{\theta_j}(u,v) = \exp\left[-\frac{1}{2}\left(\frac{(v - D(u,v)\cos\theta_j)^2 + (u - D(u,v)\sin\theta_j)^2}{D(u,v)^2\, W_\theta^2}\right)\right], \tag{4.3}$$

where, W_θ is the total width of function in degrees around the direction of the function. θ_j is orientation of j^{th} directional function. Figure 4.5(b) shows the 2D plot of a directional function and Figure 4.5 (c) shows the final window function.

b. For a given block $F_{a,b}$ (u,v), all directional window functions are multiplied with $F_{a,b}$ (u,v) and absolute energy is computed against each function.

$$E_{a,b}(i,j) = |\, H_{ij}(u,v)\, F_{a,b}(u,v)\,|. \tag{4.4}$$

c. The dominant frequency i and orientation j of the block is chosen to be the one for which $E_{a,b}$ (i,j) gives the maximum value, i.e.

$$\text{Dominant } i,j\big|_{a,b} = \arg_{\max i,j}\left(E_{a,b}(i,j)\right). \tag{4.5}$$

2. This process is repeated for every block and a frequency map f and orientation map O is constructed by storing dominant frequency and direction for every block.

3. Frequency and direction maps are then smoothed using Gaussian smoothing kernel to obtain the final smoothed frequency and orientation maps f_s and O_s.

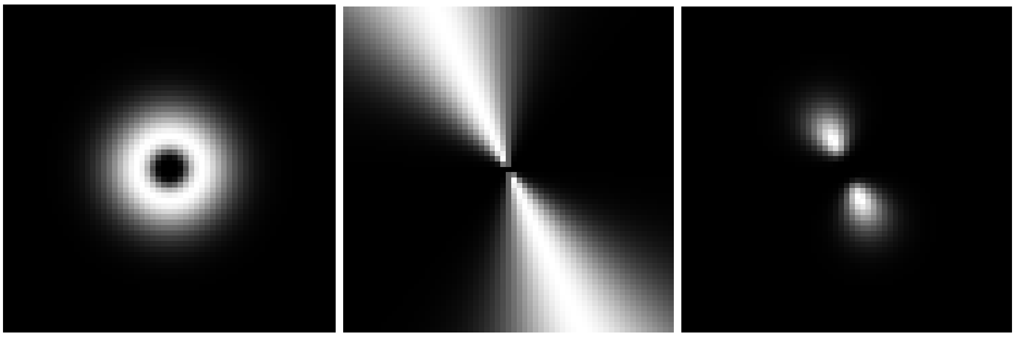


Figure 4.5 Gaussian window function (a) band selection function, (b) directional function, (c) final window function

4.3.3 Block-wise Filtering in Frequency Domain

As discussed in the previous section, Fourier transform $F_{ab}(u,v)$ of each block is enhanced by a filter that suppresses all the frequencies other than those in the dominant band and directions. For this purpose, the notch pass filter defined by the window function in Figure 4.5 (c) is used. $H_{ij}(u,v)$ will act as the Gaussian directional band-pass filter when the i and j are set to values given by the dominant frequency and orientation in the smoothed maps estimated earlier, i.e.

$$i = f_s\,(a,b),\text{ and } j = O_s\,(a,\,b). \tag{4.6}$$

Filtering is performed for each block $a,\,b$ as follows:

$$F_{a,b}{}'(u,v) = H_{ij}(u,v)\,F_{a,b}(u,v), \tag{4.7}$$

where $F_{a,b}{}'$ is the enhanced filtered image in Frequency domain. Each block is then converted back to spatial domain by computing the Inverse Fourier transform.

4.3.4 Frequency normalization

The purpose of frequency normalization is to bring all local frequencies as near as possible to the mean frequency f_m of the fingerprint. To perform the normalization, spatial domain scaling factors $Sx_{a,b}$ and $Sy_{a,b}$ for each block a, b are first calculated, using the following equations:

$$S_{a,b} = \frac{f_s \,|_{ab}}{f_m}, \tag{4.8a}$$

$$S\,x_{a,b} = \left(S_{a,b} - 1\right) * \sin\left(O_s\left(a,b\right)\right)^2 + \left|\left(S_{a,b} - 1\right) * \cos\left(O_s\left(a,b\right)\right) * \sin\left(O_s\left(a,b\right)\right)\right| + 1, \tag{4.8b}$$

$$S\,y_{a,b} = \left(S_{a,b} - 1\right) * \cos\left(O_s\left(a,b\right)\right)^2 + \left|\left(S_{a,b} - 1\right) * \cos\left(O_s\left(a,b\right)\right) * \sin\left(O_s\left(a,b\right)\right)\right| + 1. \tag{4.8c}$$

Each block is then locally normalized by either stretching or contracting it, based on its scaling factor. A block with frequency equal to f_{m} will remain as it is, whereas block with frequency less than f_{m} will be contracted by $S_{a,b}$ and vice versa. In order to resize a block, 2D geometric scale transformation is performed based on bi-cubic interpolation. Notice that for the proposed approach, blocks are square, however scaling factors are different for both x and y directions. The block size tx_{ab} and ty_{ab} for each block is, therefore given as:

$$t\,x_{a,b} = m * S\,x_{a,b}, \tag{4.9a}$$

$$t\,y_{a,b} = m * S\,y_{a,b}. \tag{4.9b}$$

After normalizing each block, a globally normalized image is created with uniform inter-ridge distance. This image can be constructed by intelligently blending all normalized and enhanced blocks. The problem in constructing normalized image can be better explained by the example shown in Figure 4.6. Consider a small portion of image, divided into blocks of size m x m with block centers $A(x_1, y_1)$, $B(x_2, y_2)$, $C(x_3, y_3)$, $D(x_4, y_4)$ etc., as shown in Figure 4.6 (a). Figure 4.6 (b) shows the example when all these blocks are scaled by keeping their centers fixed. As a result of block scaling, few blocks may overlap with each other or some gaps may be left between blocks as shown in Figure 4.6 (b). Both these phenomenon's are not desirable.

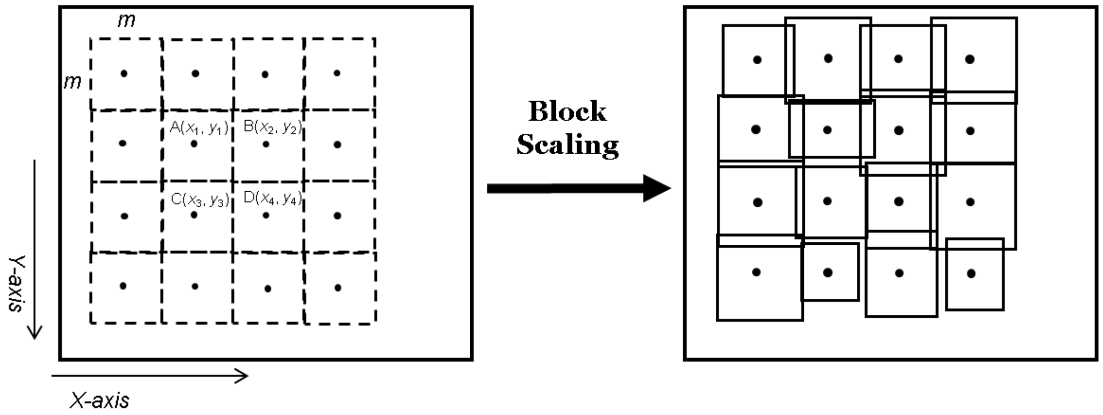


Figure 4.6 Problem in normalized image construction, (a) Image portion with block size *m* x *m*, (b) Image portion after block scaling

In order to avoid this situation, the centers of blocks need to be re-adjusted such that the overlap or gaps between the blocks can be minimized.

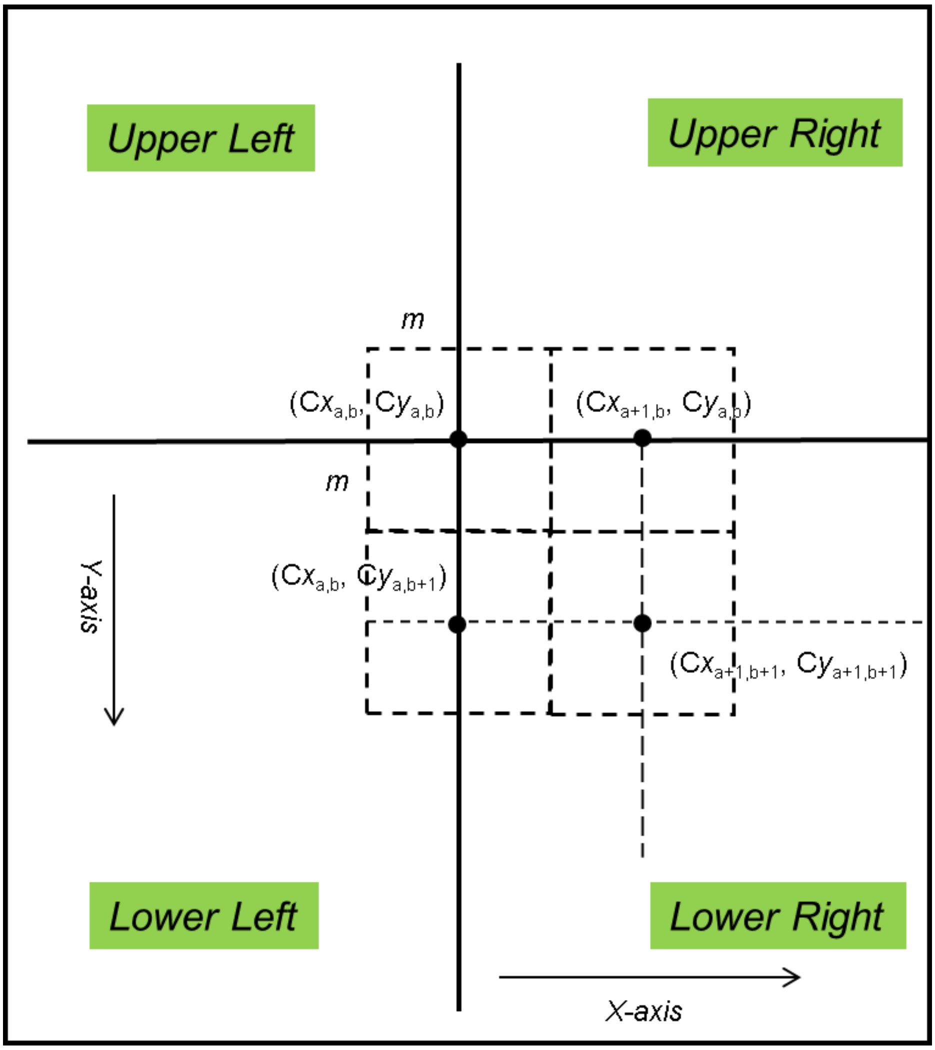

Upper Left
Upper Right
m
($Cx_{a,b}$, $Cy_{a,b}$)
($Cx_{a+1,b}$, $Cy_{a,b}$)
m
($Cx_{a,b}$, $Cy_{a,b+1}$)
($Cx_{a+1,b+1}$, $Cy_{a+1,b+1}$)
Y-axis
Lower Left
Lower Right
X-axis

Figure 4.7 Normalized grid construction map

In the proposed approach, the center of the image is taken as reference and the image is divided into four quadrants, i.e. upper left, upper right, lower left and lower right quadrant as shown in Figure 4.7. The procedure for calculating new block centers for normalized grid of lower right quadrant of an image is as follows:

a. The normalized central block is placed in the output image at its original position, i.e. center of the whole image.

b. For blocks extending horizontally outwards from center of the image new normalized center location is defined as:

$$Cx_{a,b} = \frac{tx_{a-1,b} + tx_{a,b}}{2} + Cx_{a-1,b}, \tag{4.10a}$$

$$Cy_{a,b} = Cy_{a,b}. \tag{4.10b}$$

c. For blocks extending vertically downwards from center of the image new centers of each block is defined as:

$$Cx_{a,b} = Cx_{a,b}, \tag{4.11a}$$

$$Cy_{a,b} = \frac{ty_{a,b-1} + ty_{a,b}}{2} + Cy_{a,b-1}. \tag{4.11b}$$

d. For all the remaining blocks the new central location of each block is defined as:

$$Cx_{a,b} = \frac{1}{2}\left(\left(\frac{tx_{a-1,b} + tx_{a,b}}{2} + Cx_{a-1,b}\right) + Cx_{a,b-1}\right), \tag{4.12a}$$

$$Cy_{a,b} = \frac{1}{2}\left(\left(\frac{ty_{a,b-1} + ty_{a,b}}{2} + Cy_{a,b-1}\right) + Cy_{a-1,b}\right). \tag{4.12b}$$

All the shifted centers for the remaining quadrants can be computed in similar way by just considering the geometrical structure.

4.3.5 Enhanced Image Construction

After computing the normalized grid of block centers, the whole normalized image is constructed. For this purpose, simple tiling approach cannot be used as this approach will introduce blocking effect in the image. In order to avoid blocking effect, a modified version of inter-block bilinear interpolation is used.

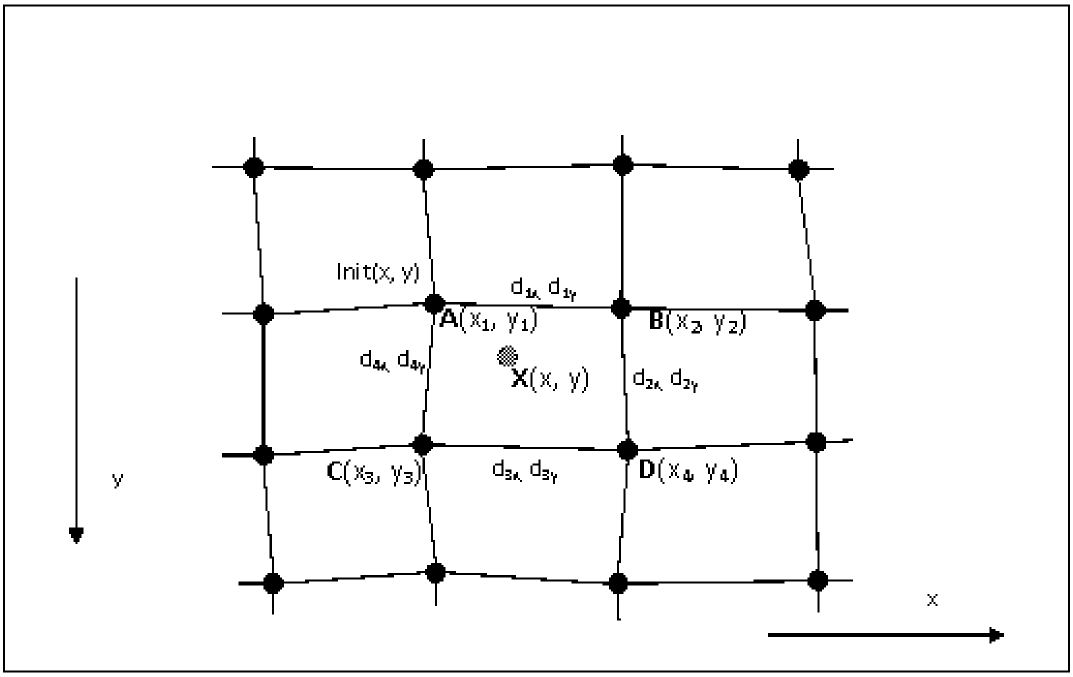


Figure 4.8 Interpolation on non-rigid grid

For each pixel of normalized enhanced image, four nearest centers and the normalized blocks associated with them are selected. Notice that the grid with redefined centers is not regular. Separation between adjacent centers for normalized enhanced image is not constant. The value of each pixel is calculated by bilinear interpolation of the values at that pixel in the four selected normalized blocks. Consider the case of calculating value of pixel X in Figure 4.8. The pixel value is to be calculated using four neighboring enhanced blocks A, B, C, D with centers (x_1, y_1), (x_2, y_2), (x_3, y_3) and (x_4, y_4), respectively. Value of X(x, y) is given as:

$$X = \frac{1}{w}(a\ w_1 + b\ w_2 + c\ w_3 + d\ w_4), \tag{4.13}$$

where a,b,c,d are the pixel intensity values in four selected blocks given by

$$a = A\{\ \nabla_{1x}, \nabla_{1y}\}, \tag{4.14a}$$

$$b = B\{\ \nabla_{2x}, \nabla_{2y}\}, \tag{4.14b}$$

$$c = C\{\, \nabla_{1x} - d_{4x}, \nabla_{1y} - \ d_{4y}\}, \tag{4.14c}$$

$$d = D\{\, \nabla_{2x} - d_{2x}, \nabla_{2y} - d_{2y}\}, \tag{4.14d}$$

Here the distances ∇_{1x}, ∇_{2x}, ∇_{1y} and ∇_{2y} are given as:

$$\nabla_{1x} = x - x_1\ , \tag{4.15a}$$

$$\nabla_{2x} = x - x_2\ , \tag{4.15b}$$

$$\nabla_{1y} = y - y_1, \tag{4.15c}$$

$$\nabla_{2y} = y - y_2, \tag{4.15d}$$

The interpolation weights are given as:

$$w = d_{1x} + \frac{d_{2x}\nabla_{2y}}{d_{2y}} - \frac{d_{4x}\nabla_{1y}}{d_{4y}}\ , \tag{4.16a}$$

$$w_1 = \frac{(d_{4y} - \nabla_{1y})}{d_{4y}}\left(\frac{d_{2x}\nabla_{2y}}{d_{2y}} - \nabla_{2x}\right), \tag{4.16b}$$

$$w_2 = \frac{(d_{2y} - \nabla_{2y})}{d_{2y}}\left(\nabla_{1x} - \frac{d_{4x}\nabla_{1y}}{d_{4y}}\right), \tag{4.16c}$$

$$w_3 = \frac{\nabla_{1y}}{d_{4y}}\left(\frac{d_{2x}\nabla_{2y}}{d_{2y}} - \nabla_{2x}\right), \tag{4.16d}$$

$$w_4 = \frac{\nabla_{2y}}{d_{2y}}\left(\nabla_{1x} - \frac{d_{4x}\nabla_{1y}}{d_{4y}}\right), \tag{4.16e}$$

where $d_{1\mathrm{x}}$, $d_{2\mathrm{x}}$, $d_{2\mathrm{y}}$, $d_{4\mathrm{x}}$, and $d_{4\mathrm{y}}$ are distances between the corner points of the block to be rectified as shown in Figure 4.8. In the above equations, it is assumed that enhanced blocks are centered at coordinates (0,0). The complete derivation of bi-linear interpolation procedure involved in the proposed scheme is presented in Appendix A.

4.4 Results

In order to show the frequency normalization effect, a comparison of the proposed enhancement technique with Gabor based enhancement [30] is presented in Figure 4.9 and Figure 4.10. Figure 4.9 (a) and 4.10 (a), shows two different fingerprint images of the same person from FVC 2004 DB1-A [55], but they appear to be different because of non-uniform inter-ridge separation. Figure 4.9 (b), 4.10 (b) and 4.9 (c), 4.10 (c) shows results of Gabor based enhancement [30] and the proposed enhancement, respectively. For clarity of the frequency normalization effect, enhancement results are shown by using thinned images [3]. The enhanced images have zero mean because the DC component is suppressed in frequency domain filtering. Therefore, the DC value i.e. zero can be used as a global threshold for binarizing enhanced image. After binarization, thinning is performed using morphological operations [3].

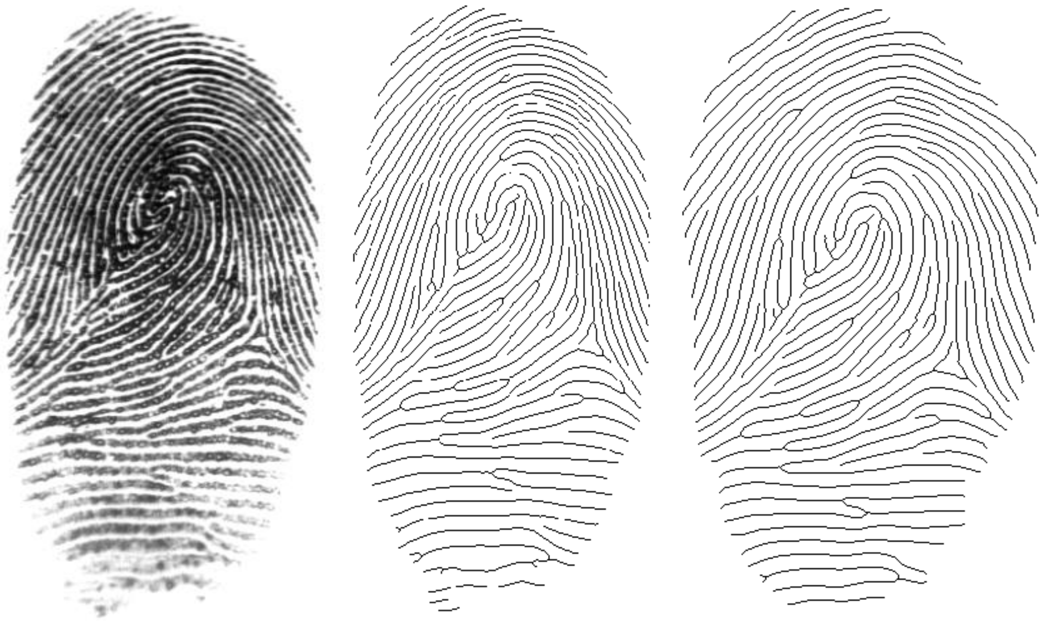


Figure 4.9: Example 1: application of the proposed frequency distortion removal enhancement technique on fingerprint image and its comparison with Gabor Enhancement, (a) Fingerprint 1, (b) Thinned Image based on Gabor Enhancement [30], (c) Thinned Image based on our proposed Enhancement technique

Figure 4.9 (b) and 4.10 (b) shows thinned images after enhancing fingerprint image using Gabor based enhancement [30]. It can be observed that using Gabor based enhancement [30] only the ridge structure is enhanced. However, the inter-ridge separation is still non-uniform in the output images. The enhancement results (thinned images) of the proposed enhancement technique are shown in Figures 4.9 (c) and 4.10 (c). It can be observed that the proposed technique not only improves the ridge structure of fingerprint image but it also effectively reduces non-uniform inter-ridge separation.

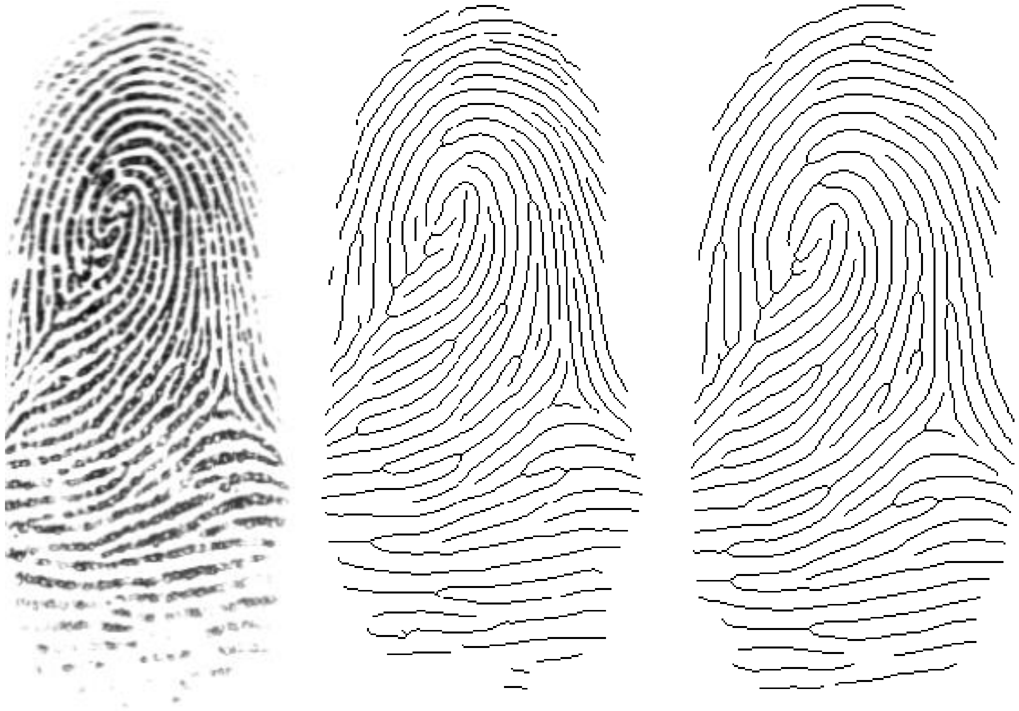


Figure 4.10: Example 2: application of the proposed frequency distortion removal enhancement technique on fingerprint image and its comparison with Gabor Enhancement, (a) Fingerprint 2, (b) Thinned Image based on Gabor Enhancement [30], (c) Thinned Image based on our proposed Enhancement technique

In order to evaluate performance of the proposed frequency distortion removal enhancement technique, similar methodology to the one adopted in section 3.4 is used. Table 4, report the error rates of the proposed enhancement technique, in comparison to other state-of-the-art enhancement techniques for FVC 2002 DB1-A database [53]. The results show that the proposed technique performs better as compared to other enhancement techniques.

Table 4: Comparative analysis of the proposed frequency distortion removal enhancement technique based on EER

Algorithm/method	**EER on FVC2002 DB1-A Databases**
Hong's Gabor Enhancement [30]	2.3%
Chikkerur's STFT Enhancement [31]	1.95%
Wang's LogGabor Enhancement [32]	1.85%
Proposed Enhancement	**1.64%**

4.5 Summary

This chapter discusses a novel contextual filtering based fingerprint enhancement technique to simultaneously overcome the effects of non-uniform inter-ridge frequencies along with broken ridges and scars. The proposed enhancement technique involves processing both in frequency and spatial domains. The problem of broken ridges and scars is resolved in frequency domain, where directional filtering is applied on overlapping sub-blocks of fingerprint image based on the dominant frequency and orientation of each block. From the dominant frequencies of each block a scaling image is computed. In spatial domain, all the overlapping blocks are scaled based on the scaling image such that the inter-ridge frequency of all the blocks become normalized. The scaled sub-blocks are then stitched together by using a modified version of bi-linear interpolation method to obtain the normalized enhanced image. The results shows that the proposed enhancement technique not only improves the problem of broken ridges and scars but also reduces the effect of non-uniform inter-ridge frequency.

Chapter 5

CONCLUSION AND FUTURE WORK

5.1 Conclusion

In this research two fingerprint enhancement approaches are proposed to overcome different distortions including non-uniform illumination, scars and broken ridge and non-uniform inter-ridge frequency.

The first approach discussed in chapter 3, proposes an effective fingerprint enhancement technique based on directional contextual filtering. The enhancement technique is novel in the sense that it divides the tasks of band-pass filtering and directional filtering into frequency and spatial domain respectively, exploiting the strengths of both these domains. The main advantage of the proposed technique is that it avoids calculation of frequency maps before applying contextual filtering thus reducing the chances of errors during frequency estimates. The enhanced images, as shown in the obtained results, are of good quality having smooth ridge/valley structure. In order to determine the performance of the proposed enhancement technique, a comparative analysis of error rates on standard fingerprint databases has also been presented with major contextual filtering based enhancement schemes. Experimental results demonstrate that proposed enhancement outperforms other well-known enhancement techniques, i.e. Hong’s Gabor filtering [30], Chikkerur’s STFT [31] and Wang’s Log-Gabor filtering [32]. A detailed computational complexity analysis of the proposed two-fold enhancement technique is also carried out in this study in section 3.3. The analysis shows that the complexity of the proposed enhancement is comparable to other well-known enhancement techniques like Hong’s Gabor filtering [30], Chikkerur’s STFT [31] and Wang’s Log-Gabor based enhancement [32].

The second approach explained in chapter 4, focuses on overcoming adverse effects of non-linear scaling along with broken ridges and scars by normalizing the inter-ridge spacing of fingerprint images. In the proposed approach, Short time Fourier transform analysis of fingerprint is performed by dividing image into overlapping blocks and then computing Fourier transform of each block as was done by Chikkerur et al. [31]. However, a different and novel approach is proposed for the frequency domain filter design and block tiling approach in spatial domain. Moreover in the same step, non-uniform ridge frequency problem is also solved by a novel frequency normalization and nonlinear tiling approach. Earlier frequency normalization pre-processing technique presented in [14] could only work on thinned images and would fail on severely degraded images. On the other hand, contextual filtering based enhancement techniques presented in [30, 31] could only be used for ridge enhancement and would not cater for frequency normalization. The enhanced image as shown in the obtained results overcomes both kind of distortions and enhanced images are of good quality having uniform frequency and smooth ridge/valley structure.

5.2 Future Work

1) In the proposed two-fold contextual enhancement, fixed window size is used during directional smoothing on unified band-pass filtered image. It has been noted in section 3.1 that due to this fixed window size some blurring along ridges may appears specially in regions with high curvature. The proposed two-fold contextual enhancement may be further improved by using adaptive window sizes during directional smoothing. Therefore, in order to reduce the blurriness an adaptive method can be devised that use larger window for regions with low curvature and smaller window for regions with high curvature.

2) Rigorous Evaluation of the Frequency Normalization Algorithm

a) It has been noted that the results of severely distorted images improve a lot when frequency normalization is used. However, for images with nearly same frequency this method may deteriorate the matching results, because of using interpolation. Therefore, a method for selected use of frequency normalization has to be devised that may check the frequency variance against any threshold before application of frequency normalization.

b) The frequency normalization procedure may be further improved by expectation maximization using some iterative procedure that includes two iterative steps, i.e. locating centres and scaling parameters of blocks accordingly.

c) The proposed enhancement technique reduces the nonlinearity among ridges by making the inter-ridge distance nearly equal. This will enable the matcher process to have more strict thresholds while matching the features. In this scenario, a simple rigid matcher can be devised for feature matching.

3) Enhancement of Palmprint images

High-resolution palmprint images share common features with fingerprint images, like ridges structure and minutiae points [6]. Contextual filtering based enhancement techniques can also be used for palmprint. Palmprint enhancement is more challenging problem as compared to fingerprint because of the large size of palmprint image as compared to fingerprint. For palmprint enhancement, Cappelli et al. [9] used traditional Gabor filtering [30] based enhancement that is proposed for fingerprint enhancement. The effectiveness of the proposed two-fold contextual filtering based enhancement techniques can be further evaluated for enhancement of high-resolution palmprint images.

REFERENCES

[3] A. K. Jain, A. Ross, and S. Prabhakar, “An Introduction to Biometric Recognition,” IEEE Trans. Circuits and Systems for Video Technology, vol. 14, pp. 4-20, 2004.

[4] Biometrics market expected to hit $12 billion in 2015. Homeland Security News Wire. 18 January 2011. webpage currently available at: http://www.homelandsecuritynewswire.com/biometrics-market-expected-hit-12-billion-2015-0 (Date of access: 22-May-2014).

[5] D. Maltoni, D. Maio, A. K. Jain, S. Prabhakar, “Handbook of Fingerprint Recognition”, 2nd Edition, Springer, New York, 2009.

[6] W. Zhao, R. Chellappa, A. Rosenfeld, P. J. Phillips, "Face Recognition: A Literature Survey." ACM Computing Surveys, Vol. 35, No. 4, pp. 399–458, 2003.

[7] L. Masek, “Recognition of Human Iris Patterns for Biometric Identification,” thesis (MS), University of Australia, 2003.

[8] A. K. Jain and J. Feng. “Latent palmprint matching”. IEEE Transactions on Pattern Analysis and Machine Intelligence, June 2009.

[9] J. Daugman, “How iris recognition works”, IEEE Trans. Circuits Syst. Video Technol., vol. 14, no. 1, pp. 21–30, Jan. 2004.

[10] A. Kong, D. Zhang, and M. Kamel, “A Survey of Palmprint Recognition,” Pattern Recognition, Vol. 42, pp. 1408-1418, 2009.

[11] R. Cappelli, M. Ferrara and D. Maio, "A Fast and Accurate Palmprint Recognition System based on Minutiae", IEEE Transactions on Systems, Man and Cybernetics - Part B, vol.42, no.3, pp.956-962, June 2012.

[12] Neurotechnology Inc., VeriFinger, webpage currently available at: http://www.neurotechnology.com (Date of access: 17-Oct-2014).

[13] A. K. Jain, L. Hong, and B. R. “On-line fingerprint verification”. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(4):302–314, 1997.

[14] X. Jiang and W. Y. Yau, “Fingerprint Minutiae Matching Based on the Local and Global Structures,” in Proc. Int. Conf. on Pattern Recognition (15th), vol. 2, pp. 1042–1045, 2000.

[15] N. K. Ratha, V. D. Pandit, R. M. Bolle and V. Vaish, “Robust Fingerprint Authentication Using Local Structural Similarity,” in Proc. Workshop on Applications of Computer Vision, pp. 29–34, 2000.

[16] A. W. Senior and R. Bolle, “Improved fingerprint matching by distortion removal,” IEICE Transactions on Information and Systems (special issue on biometrics), vol. E84–D, no. 7, pp. 825–832, 2001.

[17] Mubeen Ghafoor, Imtiaz A. Taj, Waqas Ahmad, M Noman Jafri, “Efficient 2-fold Contextual Filtering Approach for Fingerprint Enhancement”, IET Image Processing, Volume 8, Issue 7, p. 417 – 425, July 2014.

[18] Mubeen Ghafoor, Imtiaz A. Taj, M Noman Jafri, “Fingerprint Frequency Normalization and Enhancement using 2-D STFT Analysis”, submitted to IET Image processing, (under review).

[19] D. Maio and D. Maltoni, “Direct Gray-Scale Minutiae Detection in Fingerprints”. IEEE Transactions on Pattern Analysis Machine Intelligence, 1997, vol.19, no.1, pp.27-40, 1997.

[20] C. I. Watson, M. D. Garris, E. Tabassi, C. L. Wilson, McCabe R. M., S. Janet and K. Ko, “User's Guide to NIST Biometric Image Software (NBIS). National institute of standards and technology”, 2002.

[21] Fingerprint Verification Competition Ongoing. Available at: https://biolab.csr.unibo.it/fvcongoing/UI/Form/Home.aspx (Date of access: 17-Oct-2014).

[22] B. G. Kim and D. J. Park, “Adaptive image normalization based on block processing for enhancement of fingerprint image,” Electronics Letters, vol. 38, no. 14, pp. 696–698, 2002.

[23] S. Greenberg, M. Aladjem, D. Kogan and I. Dimitrov, “Fingerprint Image Enhancement Using Filtering Techniques,” in Proc. Int. Conf. on Pattern Recognition (15th), vol. 3, pp. 326–329, 2000.

[24] A. J. Willis and L. Myers, “A cost–-effective fingerprint recognition system for use with low-quality prints and damaged fingertips,” Pattern Recognition, vol. 34, no. 2, pp. 255–270, 2001.

[25] A. M. Bazen and S. H. Gerez, “Segmentation of Fingerprint Images,” in Proc. Workshop on Circuits Systems and Signal Processing (ProRISC 2001), 2001.

[26] E. Chandra and K. Kanagalakshmi, "Noise Elimination in Fingerprint Images using Median Filter", Int. Journal of Advanced Networking and Applications, (2011), Vol. 02, Issue:06, pp: 950-955, 2011.

[27] A. Grasselli “On the automatic classification of fingerprints”. Methodologies of Pattern Recognition, S. Watanabe (Ed.), Academic, New York, 1969.

[28] M. Kass and A. Witkin, “Analyzing oriented patterns,” Computer Vision Graphics and Image Processing, vol. 37, no. 3, pp. 362–385, 1987.

[29] N. K. Ratha, S. Y. Chen and A. K. Jain, “Adaptive flow orientation-based feature extraction in fingerprint images,” Pattern Recognition, vol. 28, no. 11, pp. 1657–1672, 1995.

[30] A. M. Bazen and S. H. Gerez, “Systematic methods for the computation of the directional fields and singular points of fingerprints,” IEEE Transactions on Pattern Analysis Machine Intelligence, vol. 24, no. 7, pp. 905–919, 2002.

[31] F. Turroni, D. Maltoni, R. Cappelli, and D. Maio, “Improving fingerprint orientation extraction,” IEEE Trans. Inf. Forensics Security, vol. 6, no. 3, pp. 1002–1013, Sep. 2011.

[32] L. Hong, Y. Wan and A. K. Jain, “Fingerprint image enhancement: Algorithms and performance evaluation”, IEEE Transactions on Pattern Analysis Machine Intelligence, vol. 20, no. 8, pp. 777–789, 1998.

[33] S. Chikkerur, A. N. Cartwright and V. Govindaraju, “Fingerprint enhancement using STFT analysis. Pattern Recognition”, vol. 40, no. 1, pp. 198–211, 2007.

[34] W. Wang, J. W. Li, F. F. Huang, H. L. Feng, “Design and implementation of Log-Gabor filter in fingerprint image enhancement”, Pattern Recognition Letters, vol. 29, pp. 301-308, 2008.

[35] R. C. Gonzalez, R. E. Woods, “Digital Image Processing (3rd Edition)”, Prentice Hall, 3 edition, August 2007.

[36] R. C. Gonzalez, R. E. Woods, and S. L. Eddins. Digital Image Processing Using MATLAB, 2nd ed. Gatesmark Publishing, 2nd edition, 2009.

[37] A. K. Jain, S. Prabhakar, L. Hong, and S. Pankanti, “Filterbank - based fingerprint matching”, IEEE Transactions on image processing, vol. 9 no.5, pp. 846-859, 2000.

[38] Y. Jie, Y. Y. Fang, Z. Renjie and S. Qifa, “Fingerprint minutiae matching algorithm for real time system,” Pattern Recognition, vol. 39, no. 1, pp. 143–146, 2006.

[39] S. Chikkerur and N. Ratha, “Impact of Singular Point Detection on Fingerprint Matching Performance,” in Proc. Workshop on Automatic Identification Advanced Technologies, pp. 207–212, 2005.

[40] J. Zhou and J. Gu, “A Model-Based Method for the Computation of Fingerprints’ Orientation Field,” IEEE Trans. Image Processing, vol. 13, no. 6, pp. 821-835, June 2004.

[41] M. Tico, and P. Kuosmanen, “An Algorithm for Fingerprint Image Post processing” in Thirty-Fourth Asilomar Conference on Signals, Systems, and Computers, October 29, 2000 - November 1, 2000.

[42] A. K. Jain, J. Feng, A. Nagar, and K. Nandakumar, “On Matching Latent Fingerprints,” Proc. CVPR Workshop Biometrics, June 2008.

[43] S . Chikkerur, A. N. Cartwright and V. Govindaraju, “K-plet and CBFS: A Graph based Fingerprint Representation and Matching Algorithm”, Proceedings of the International Conference on Biometrics, Hong Kong, 2006.

[44] X. Jiang, W.-Y. Yau, “Fingerprint minutiae matching based on the local and global structures”, Internat. Conf. Pattern Recognition 2 (2000), pp. 1038–1041, 2000.

[45] T. Y. Jea and V. Govindaraju, "A minutia-based partial fingerprint recognition system," Pattern Recognition, vol. 38, pp. 1672-1684, 2005.

[46] M. Tico, P. Kuosmanen, Fingerprint matching using an orientation-based minutia descriptor, IEEE Trans. Pattern Anal. Mach. Intell. 25 (8) (2003), pp. 1009–1014, 2003.

[47] J. Feng, "Combining minutiae descriptors for fingerprint matching," Pattern Recognition, vol. 41, pp. 342 - 352, 2008.

[48] L. O’Gorman and J. V. Nickerson, “An approach to fingerprint filter design. Pattern Recognition”, vol. 22, no. 1, pp. 29–38, 1989.

[49] B. G. Sherlock, D.M. Monro and K. Millard, “Algorithm for enhancing fingerprint images,” Electronics Letters, vol. 28, no. 18, pp. 1720, 1992.

[50] B. G. Sherlock, D.M. Monro and K. Millard, “Fingerprint enhancement by directional Fourier filtering”, IEE Proceedings Vision Image and Signal Processing, vol. 141, no. 2, pp. 87–94, 1994.

[51] Carsten Gottschlich, “Curved-Region-Based Ridge Frequency Estimation and Curved Gabor Filters for Fingerprint Image Enhancement”, IEEE Transactions on Image Processing 21 (4), 2220-2227, 2012.

[52] C. Tang, N. Yang, Q. Mi, “Fingerprint enhancement using the second-order oriented partial differential equation”, Image and Signal Processing (CISP), 2012 5th International Congress on , vol., no., pp.307,311, 16-18 Oct. 2012.

[53] F. Turroni, R. Cappelli, D. Maltoni, "Fingerprint enhancement using contextual iterative filtering," Biometrics (ICB), 2012 5th IAPR International Conference, vol., no., pp.152, 157, 2012.

[54] P. Sutthiwichaiporn, V. Areekul, "Adaptive boosted spectral filtering for progressive fingerprint enhancement, Pattern Recognition", Volume 46, Issue 9, Pages 2465-2486, ISSN 0031-3203, 2013.

[55] Fingerprint verification competition 2002. Available at: http://bias.csr.unibo.it/fvc2002/ (Date of access: 22-Aug-2014).

[56] Movellan, R. Javier, Retrieved 2008-05-14. Tutorial on Gabor Filters. Available at: http://mplab.ucsd.edu/tutorials/gabor.pdf (Date of access: 22-April-2014).

[57] Fingerprint verification competition 2004. Available at: http://bias.csr.unibo.it/fvc2004/ (Date of access: 17-Oct-2014).

[58] N. K. Ratha and R. M. Bolle, “Effect of controlled image acquisition on fingerprint matching”, in Proceedings of the International Conference on Pattern Recognition, 1998, vol. II, pp. 1659–1661, 1998.

[59] C. Dorai, N. Ratha, and R. Bolle, “Detecting dynamic behaviour in compressed fingerprint videos: Distortion”, in Proceedings of Computer Vision and Pattern Recognition, June 2000, pp. 320–326, 2000.

[60] Z. M. Kov´acs-Vajna, “A fingerprint verification system based on triangular matching and dynamic time warping”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 11, pp. 1266–1276, November 2000.

[61] D. Lee, K. Choi and J. Kim, “A Robust Fingerprint Matching Algorithm Using Local Alignment,” in Proc. Int. Conf. on Pattern Recognition (16th), vol. 3, pp. 803–806, 2002.

[62] P. Bhowmick and B.B. Bhattacharya, “Approximate Fingerprint Matching Using kd-tree,” in Proc. Int. Conf. on Pattern Recognition (17th), vol. 1, pp. 544–547, 2004.

[63] E. Henry, Classification and uses of fingerprints, Routledge, London, 1900.

[64] S. Jirachaweng, T. Leelasawassuk, and V. Areekul, “Performance and Computational Complexity Comparison of Block-Based Fingerprint Enhancement”, Advances in Biometrics, vol. 5558, pp. 656-665, 2009.

[65] National ICT R&D Fund, Ministry of IT and Telecom, Pakistan. http://www.ictrdf.org.pk/ (Date of access: 07-Sep-2014).

APPENDICES

Appendix A – Bi-linear interpolation for non-regular 2D grid

Bilinear interpolation is performed to find the values at random positions on a regular 2-D grid. Bilinear interpolation extends from linear interpolation by first performing linear interpolation in one direction followed by interpolation in the other direction. An example of finding intensity value at position *X*(*x*, *y*) using bilinear interpolation on a regular 2-D grid is shown in Figure A.1.

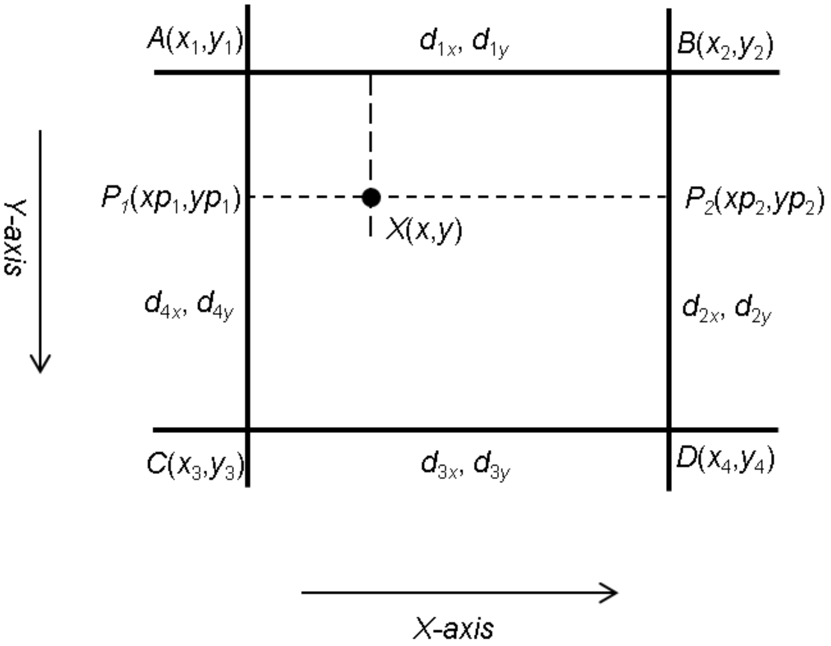


Figure A.1: Bilinear Interpolation on 2-D regular grid

The intensity value at position $P_1(xp_1, yp_1)$ can be estimated by linearly interpolating between pixel vales at $A(x_1, y_1)$ and $C(x_3,y_3)$ as given in following equation:

$$P_1(xp_1, yp_1) = C(x_3, y_3) * \frac{(\nabla_{1y})}{d_{4y}} + A(x_1, y_1) * \frac{(d_{4y} - \nabla_{1y})}{d_{4y}}, \qquad \text{(A.1)}$$

Here the distances ∇_{1y} and d_{4y} are given as:

$$\nabla_{1y} = y - y_1, \tag{A.2}$$

$$d_{4y} = y_3 - y_1, \tag{A.3}$$

Similarly, the intensity value at $P_2(xp_2, yp_2)$ is estimated by linearly interpolating between pixel vales $B(x_2, y_2)$ and $D(x_4,y_4)$ as given in following equation:

$$P_2(xp_2, yp_2) = D(x_4, y_4) * \frac{(\nabla_{2y})}{d_{2y}} + B(x_2, y_2) * \frac{(d_{2y} - \nabla_{2y})}{d_{2y}}, \tag{A.4}$$

$$\nabla_{2y} = y - y_2, \tag{A.5}$$

$$d_{2y} = y_4 - y_2, \tag{A.6}$$

The intensity value at position $X(x, y)$ is finally estimated by linearly interpolating between intensity vales at $P_1(xp_1, yp_1)$ and $P_2(xp_2, yp_2)$ as given in following equation:

$$X(x, y) = P_2(xp_2, yp_2) * \frac{(x - xp_1)}{(xp_2 - xp_1)} + P_1(xp_1, yp_1) * \frac{(xp_2 - x)}{(xp_2 - xp_1)} \tag{A.7}$$

In case of regular grid shown in Figure A.1, the value of xp_1 is equal to x_1 and x_3, similarly xp_2 is equal to x_2 and x_4, so we can write the above equation as:

$$X(x, y) = P_2(xp_2, yp_2) * \frac{(x - x_1)}{(x_2 - x_1)} + P_1(xp_1, yp_1) * \frac{(x_2 - x)}{(x_2 - x_1)} \tag{A.8}$$

$$X(x, y) = P_2(xp_2, yp_2) * \frac{(\nabla_{1x})}{d_{1x}} + P_1(xp_1, yp_1) * \frac{(d_{1x} - \nabla_{1x})}{d_{1x}} \tag{A.9}$$

where the distances ∇_{1y} and d_{1y} are given as:

$$\nabla_{1x} = x - x_1, \tag{A.10}$$

$$d_{1x} = x_2 - x_1, \tag{A.11}$$

Coming back to the proposed problem of bilinear interpolation as discussed in section 4.3.4, it can be noted that the grid used for interpolation is non-regular that makes the interpolation problem more complicated. A closer look of nor-regular grid that can exist in the proposed algorithm can take the form as shown in Figure A.2.

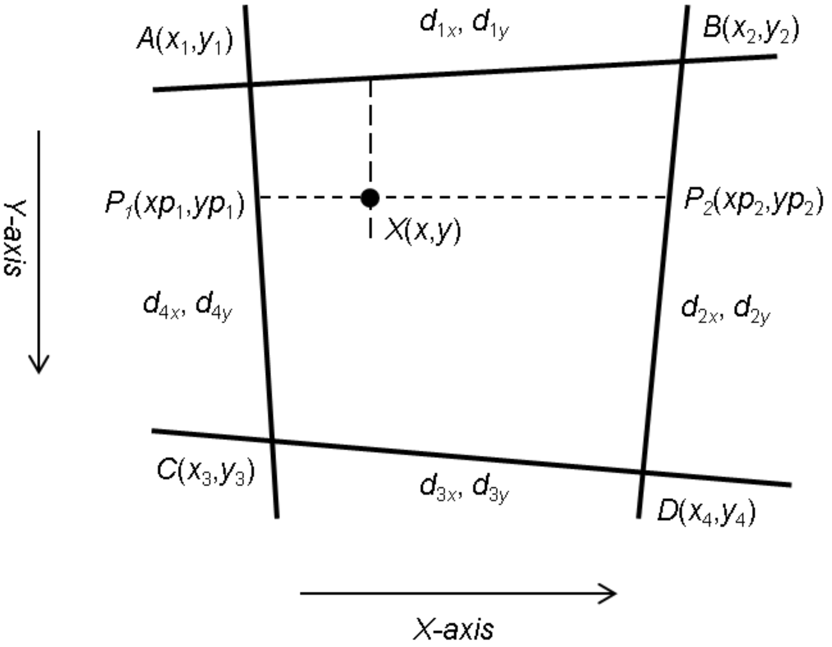


Figure A.2: Bilinear Interpolation on non-regular 2-D grid

In order to find bilinear interpolation of intensity value at position *X*(*x*,*y*) for non-regular 2-D grid in Figure 2, equation (7) can used. However, in case of non-regular grid the *x* components of P_1 and P_2 can be misaligned.

So it is needed to calculate *x*-components (xp_1, xp_2) of points P_1 and P_2 before using standard bilinear interpolation procedure as explained above. In order to estimate the *x*-components of P_1 and P_2, equation of straight line for two points is used as given in following equations:

$$xp_1 = x_1 + \frac{(\nabla_{1y}) * (d_{4x})}{d_{4y}} \tag{A.12}$$

$$xp_2 = x_2 + \frac{(\nabla_{2y}) * (d_{2x})}{d_{2y}} \tag{A.13}$$

where distances ∇_{1y}, ∇_{2y} and d_{4y} are given in equation (2), equation (5) and equation (3) respectively. Distances d_{4x}, ∇_{2y}, d_{2x} and d_{2y} are given as:

$$d_{4x} = x_3 - x_1, \tag{A.14}$$

$$\nabla_{2y} = y - y_2, \tag{A.15}$$

$$d_{2x} = x_4 - x_2, \tag{A.16}$$

The interpolated intensity value at $X(x,y)$, equation (A.7) can be re-written as:

$$X(x,y) = \frac{1}{w}\{P_2(xp_2, yp_2) * (x - xp_1) + P_1(xp_1, yp_1) * (xp_2 - x)\} \tag{A.17}$$

Here w can be written as:

$$w = xp_2 - xp_1$$

Or $$w = \left(x_2 + \frac{(\nabla_{2y}) * (d_{2x})}{d_{2y}}\right) - \left(x_1 + \frac{(\nabla_{1y}) * (d_{4x})}{d_{4y}}\right)$$

Or $$w = x_2 - x_1 + \left(\frac{(\nabla_{2y}) * (d_{2x})}{d_{2y}}\right) - \left(\frac{(\nabla_{1y}) * (d_{4x})}{d_{4y}}\right)$$

Or $$w = d_1x + \left(\frac{(\nabla_{2y}) * (d_{2x})}{d_{2y}}\right) - \left(\frac{(\nabla_{1y}) * (d_{4x})}{d_{4y}}\right)$$

In standard bilinear interpolation, value of P_1 and P_2 are calculated using weighted (averaged) intensity values of pixels $A(x_1,y_1)$, $B(x_2,y_2)$, $C(x_3,y_3)$ and $D(x_4,y_4)$. However, in problem discussed in section 4.4.1, A, B, C *and* D are the four nearest blocks around $X(x, y)$ with centers at (x_1,y_1), (x_2,y_2), (x_3,y_3) and (x_4,y_4), respectively. The intensity values at (x, y) for each block is selected based on following equations:

$$a = A\{\ \nabla_{1x}, \nabla_{1y}\}, \tag{A.18}$$

$$b = B\{\ \nabla_{2x}, \nabla_{2y}\}, \tag{A.19}$$

$$c = C\{\ \nabla_{1x} - d_{4x}, \nabla_{1y} -\ d_{4y}\}, \tag{A.20}$$

$$d = D\{\ \nabla_{2x} - d_{2x}, \nabla_{2y} - d_{2y}\}, \tag{A.21}$$

Here a, b, c, d are the pixel intensity values in four selected blocks. The distance ∇_{2x} is given as:

$$\nabla_{2x} = x - x_2 \,, \tag{A.22}$$

So, for the scenario discussed in section 4.4.1, the equation for P_1 and P_2 (equation (A.1) and equation (A.4)) can further be derived as:

$$P_2(xp_2, yp_2) * (x - xp_1) = \left(d * \frac{(\nabla_{2y})}{d_{2y}} + b * \frac{(d_{2y} - \nabla_{2y})}{d_{2y}}\right) * \left(x - \left(x_1 + \frac{(\nabla_{1y}) * (d_{4x})}{d_{4y}}\right)\right) \tag{A.23}$$

$$P_2(xp_2, yp_2) * (x - xp_1) = \left(d * \frac{(\nabla_{2y})}{d_{2y}} + b * \frac{(d_{2y} - \nabla_{2y})}{d_{2y}}\right) * \left(\nabla_{1x} x - \frac{(\nabla_{1y}) * (d_{4x})}{d_{4y}}\right) \tag{A.24}$$

$$P_2(xp_2, yp_2) * (x - xp_1) = b * \left(\frac{d_{2y} - \nabla_{2y}}{d_{2y}}\right) * \left(\nabla_{1x} x - \frac{\nabla_{1y} * d_{4x}}{d_{4y}}\right) + d * \frac{\nabla_{2y}}{d_{2y}} * \left(\nabla_{1x} x - \frac{\nabla_{1y} * d_{4x}}{d_{4y}}\right) \tag{A.25}$$

$$P_2(xp_2, yp_2) * (x - xp_1) = b * w_2 + d * w_4 \tag{A.26}$$

where w_2 and w_4 are given by following equations:

$$w_2 = \left(\frac{d_{2y} - \nabla_{2y}}{d_{2y}}\right) * \left(\nabla_{1x} x - \frac{\nabla_{1y} * d_{4x}}{d_{4y}}\right) \tag{A.27}$$

$$w_4 = \frac{\nabla_{2y}}{d_{2y}} * \left(\nabla_{1x} x - \frac{\nabla_{1y} * d_{4x}}{d_{4y}}\right) \tag{A.28}$$

$$P_1(xp_1, yp_1)*(xp_2 - x) = \left(c*\frac{(\nabla_{1y})}{d_{4y}} + a*\frac{(d_{4y} - \nabla_{1y})}{d_{4y}}\right)*\left(\left(x_2 + \frac{(\nabla_{2y})*(d_{2x})}{d_{2y}}\right) - x\right) \tag{A.29}$$

$$P_1(xp_1, yp_1)*(xp_2 - x) = \left(c*\frac{(\nabla_{1y})}{d_{4y}} + a*\frac{(d_{4y} - \nabla_{1y})}{d_{4y}}\right)*\left(\frac{(\nabla_{2y})*(d_{2x})}{d_{2y}} - \nabla_{2x}\right) \tag{A.30}$$

$$P_1(xp_1, yp_1)*(xp_2 - x) = a*\left(\frac{(d_{4y} - \nabla_{1y})}{d_{4y}}\right)*\left(\frac{\nabla_{2y} * d_{2x}}{d_{2y}} - \nabla_{2x}\right) + c\left(\frac{\nabla_{1y}}{d_{4y}}\right)*\left(\frac{\nabla_{2y} * d_{2x}}{d_{2y}} - \nabla_{2x}\right) \tag{A.31}$$

$$P_1(xp_1, yp_1)*(xp_2 - x) = a*w_1 + c*w_3 \tag{A.32}$$

where w_1 and w_3 are given by following equations:

$$w_1 = \left(\frac{(d_{4y} - \nabla_{1y})}{d_{4y}}\right)*\left(\frac{\nabla_{2y} * d_{2x}}{d_{2y}} - \nabla_{2x}\right) \tag{A.33}$$

$$w_3 = \left(\frac{\nabla_{1y}}{d_{4y}}\right)*\left(\frac{\nabla_{2y} * d_{2x}}{d_{2y}} - \nabla_{2x}\right) \tag{A.34}$$

Finally, equation for estimating interpolated values at $X(x, y)$ (equation (A.17)) can be rewritten as:

$$X(x, y) = \frac{1}{w}\{a*w_1 + b*w_2 + c*w_3 + d*w_4\} \tag{A.35}$$

Appendix B – Comparison of computational complexity for contextual filtering based enhancement techniques

In order to compare computational complexity of different enhancement algorithms i.e. Hong's Gabor filtering [30], Chikkerur's STFT [31] method, Wang's Log-Gabor [32] enhancement and the proposed two-fold enhancement technique, computational cost is measured in terms of total number of multiplications (O^x), additions (O^+) and load operations (O^l) required to enhance a fingerprint image. The computational cost of only frequency estimation and filtering algorithms of each enhancement technique will be catered. In analysis below, the size of the fingerprint image is taken as *M* x *M*, while the size of each block to be filtered is $m \times m$ for all the enhancement techniques.

B.1 Computational complexity of Hong's Gabor filtering [30]

Computational complexity for Hong's Gabor filtering based enhancement [30] is divided into two steps. In first step computational complexity of frequency estimation for fingerprint image is evaluated. Then in second step computational complexity of filtering is presented, followed by the total number of computations required for enhancement.

Step 1: Computation complexity for calculating frequency of fingerprint image

Hong et al [30] computed frequency by using x-signature algorithm. In this algorithm fingerprint image is initially divided into blocks of size *m* x *m* followed by frequency estimation within each block. Computational complexity of each step is discussed below by initially calculating the number of operations ([O^x, O^+, O^l] number of multiplications, additions and load operations respectively) required for estimating frequency within a block followed by computational complexity of frequency estimation for the complete fingerprint image.

A. Computations for estimating frequency of a block of size *m* x *m:*

a) Rotating each block according to the orientation of the local ridges such that the ridges within the block becomes vertical.

   Multiplications required: 4 m^2

   Addition required: 2 m^2

Total Operations can be given as: [4 m^2, 2 m^2, 0]

b) Computation of x-signature that requires projection of oriented window into 1D sequence.

Multiplications: 0

Addition: m^2

Total Operations can be given as: [0, m^2, 0]

c) Peak estimation from x-signature for frequency estimation

There are different methods through which frequency can be estimated from x-signature both in frequency domain as well as spatial domain. In this study, computational complexity of a frequency domain method is considered [62], where x-signature (projected sequence) is transformed to frequency domain and then peak frequency is estimated.

i) Computing Fourier transform of x-signature:

Multiplication: $m/2 \log_2 (m)$

Addition: $m \log_2 (m)$

Total Operations: [$m/2 \log_2 (m)$, $m \log_2 (m)$, 0]

ii) In order to find frequency, distance between the origin and highest peak is found from frequency spectrum of x-signature [62].

Multiplication: m

Addition: $2m$

Total Operations: [m, $2\ m$, 0]

Computations for estimating frequency of a block of size *m* x *m:*

Multiplications: $m(4\ m + 1/2 \log_2 (m) + 1)$

Additions: $m\ (3\ m + \ \log_2 (m) + 2)$

Total Operations: [$m(4\ m + 1/2 \log_2 (m) + 1)$, $m\ (3\ m + \log_2 (m) + 2)$,0]

B. **Computations for estimating frequency of an image of size *M* x *M* with block size *m* x *m:***

This will be equal to total operations required to calculate frequency within a block multiplied by the total number of block in the fingerprint image.

Multiplications: $4M^{2}/m(4m + 1/2\log_{2}(m) + 1)$

Additions: $4M^{2}/m(3m + \log_{2}(m) + 2)$

Total Operations can be written as:

$$\left\langle \begin{array}{l} \left\{ \frac{4M^{2}}{m}\left(4m + 1/2\log_{2}(m) + 1\right) \right\}, \\ \left\{ \frac{4M^{2}}{m}\left(3m + \log_{2}(m) + 2\right) \right\}, \\ \{0\} \end{array} \right\rangle \tag{B.1}$$

C. **Computations for interpolation involved in frequency estimation for an image of size *M* x *M:***

Multiplications: $4M^{2}$

Additions: $6M^{2}$

Total operations required for frequency estimation

Total Operations are given as:

$$\left\langle \begin{array}{l} \left\{ 4M^{2}\left(1 + \frac{1}{m}\left(4m + 1/2\log_{2}(m) + 1\right)\right) \right\}, \\ \left\{ 2M^{2}\left(3 + \frac{2}{m}\left(3m + \log_{2}(m) + 2\right)\right) \right\}, \\ \{0\} \end{array} \right\rangle \tag{B.2}$$

Step 2: Computational complexity for Gabor filtering proposed by Hong [30]:

i) Computations for filtering:

The filtering process of Gabor based enhancement [30] is computationally expensive as output intensity for every pixel in the fingerprint image is calculated by calculating its sum of product with a filtered. The total number of multiplications and additions in this case are:

Multiplications: M^2m^2

Additions: M^2m^2

ii) Computations for loading:

It should also be noted that for every pixel in the fingerprint image, a new filter may require to be loaded into the memory. Therefore, the total number of load operations required for filtering an image of size M x M with block size m x m can be given as:

Load / Memory operations: $M^2\ (m^2)$

Total computational complexity:

Multiplications: $M^2m^2 + 4\,M^{\,2}\,(1+ 1/\,m(4\,m + 1/2\,\log_2(m) + 1))$

Additions: $M^2m^2 + 2\,M^{\,2}(3+ 2\,/\,m\,(3\,m + \ \log_2(m) +2)\,)$

Load / Memory operations: M^2m^2

Total Operations for Hong's based enhancement is given as:

$$\left\langle \begin{array}{l} \left\{M^2m^2 + 4M^2\left(1+\frac{1}{m}\left(4m + 1/2\log_2(m)+1\right)\right)\right\}, \\ \left\{M^2m^2 + 2M^2\left(3+\frac{2}{m}\left(3m + \log_2(m)+2\right)\right)\right\}, \\ \left\{M^2m^2\right\} \end{array} \right\rangle \tag{B.3}$$

B.2 Computational complexity of Chikkerur's STFT [31] method

This section lists the computational complexity using Chikkerur's STFT [31] enhancement in terms of number of operations ([O^x , O^+, O^l] number of multiplications, additions and load operations respectively).

Step 1: Computation complexity for calculating frequency of fingerprint image

The computational complexity using Chikkerur's STFT [31] frequency estimation is given as:

1) Fingerprint image is divided into overlapping blocks of size m x m.

   a) For each block DC component is removed that requires following number of operations:

      Multiplication: *0*

      Addition: $2\ m^2$

   b) The block is then multiplied by a spectral window that requires following number of operations:

      Multiplication: m^2

      Addition: *0*

   c) Computations for FFT for a block of size m x m:

      Multiplication: $m^2/2 \log_2 (m^2)$

      Addition: $m^2 \log_2 (m^2)$

   d) Estimate frequency for each block:

      In order to compute frequency of a block, Chikkerur et al. [31] converted the Fourier spectrum of the block into polar form and then computed the probability density function and marginal density function. Followed by calculation of expected value of ridge frequency within the block. This procedure requires following computations:

      Multiplications: $7\ m^2$

      Addition: $6\ m^2$

      Total Multiplications for each block: $m^2 (8 + 1/2 \log_2 (m^2))$

      Total Additions for each block: $m^2 (8 + \log_2 (m^2))$

Computations for frequency estimation for the complete fingerprint image for a total of 2*M/m* x 2*M/m* block*:*

Multiplication: $4\ M^2\ (8 + 1/2 \log_2 (m^2))$

Addition: $4\ M^2\ (8 + \log_2 (m^2))$

Total operations are given as:

$$\left\langle \left\{ 4M^2 \left(8 + 1/2 \log_2 (m^2) \right) \right\},\ \left\{ 4M^2 \left(8 + \log_2 (m^2) \right) \right\},\ \{0\} \right\rangle \tag{B.4}$$

Step 2: Computational complexity for filtering:

The filtering process using Chikkerur et al. [31] based enhancement technique is performed by multiplying each block with a directional filter followed by computing inverse Fourier transform of the enhanced block. The computation complexity is given below:

a) Computations for filtering a block in Frequency domain:

Multiplications: (m^2)

Addition: 0

Computations for filtering for a total of 2*M/m* and 2*M/m* blocks:

Multiplications: $4M^2$

Addition: 0

Computations for load operations:

For every block in the fingerprint image, a new filter may require to be loaded into the memory. Therefore, the total number of load operations required for filtering an image of size M x M with block size m x m can be given as:

Load / Memory operations: $4\ M^2$

Total Operations: [$4M^2$, 0, $4M^2$]

b) Computations for IFFT for a block of size *m* x *n:*

Multiplication: $m^2/2 \log_2 (m^2)$

Addition: $m^2 \log_2 (m^2)$

Computations for IFFT for a total of 2*M/m* and 2*M/m* block*:*

Multiplication: $2M^2 \log_2 (m^2)$

Addition: $4\ M^2 \log_2 (m^2)$

Total Operations: [$2M^2 \log_2 (m^2)$, $4\ M^2 \log_2 (m^2)$, 0]

Total Computation required for filtering:

Multiplication: $2\,M^2\,(2 + \log_2 (m^2))$

Addition: $4\,M^2 \log_2 (m^2)$

Load / Memory operations: $4\,M^2$

$$\left\langle \left\{2M^2\left(2+\log_2(m^2)\right)\right\},\ \left\{4M^2\log_2(m^2)\right\},\ \left\{4M^2\right\}\right\rangle \tag{B.5}$$

Total Computation required for Chikkerur et al. [31] based enhancement:

Multiplication: = Multiplication operations for frequency estimation + Multiplication operations for filtering

Multiplication: = $4\,M^2\,(8 + 1/2 \log_2 (m^2)) + 2\,M^2\,(2 + \log_2 (m^2))$

Addition: = Addition operations for frequency estimation + Addition operations for filtering

Addition: = $4\,M^2\,(8 + \log_2 (m^2)) + 4\,M^2 \log_2 (m^2)$

Load / Memory operations: $4\,M^2$

Total Operations for Chikkerur's based enhancement is given as:

$$\left\langle \begin{array}{l} \left\{4M^2\left(8+1/2\log_2(m^2)\right)+2M^2\left(2+\log_2(m^2)\right)\right\}, \\ \left\{4M^2\left(8+\log_2(m^2)\right)+4M^2\log_2(m^2)\right\}, \\ \left\{4M^2\right\} \end{array} \right\rangle \tag{B.6}$$

B.3 Computational complexity of Wang's Log-Gabor [32] filtering

Wang et al. [32], proposed fingerprint enhancement in frequency domain based on Log-Gabor filters. In this technique, frequency is estimated based on the x-signature method proposed by Hong et al. [30]. Whereas, the filtering technique is similar to that of Chikkerur et al. [31], that includes application of frequency domain filtering. As mentioned, Wang et al. [32] estimated frequency based on x-signature method, whose computational complexity is already mentioned in section 3.3.1. The total computational complexity of Wang's Log-Gabor [32] is be equal to the computational complexity of frequency estimation algorithm using x-signature method, plus the computational complexity of frequency domain filtering that is discussed below.

Step 1: Computation complexity for calculating frequency of fingerprint image

The computational complexity using Hong's x-signature frequency estimation method [30] is already calculated in section B.1 in equation (B.2), that is given in terms of number of operations ([O^x , O^+, O^l] number of multiplications, additions and load operations respectively), as:

Multiplications: $4\,M^2\,(1+1/\,m(4\,m+1/2\,\log_2(m)+1))$

Additions: $2\,M^2(3+2\,/\,m\,(3\,m+\ \log_2(m)+2)\,)$

Step 2: Computational complexity for filtering:

The filtering process using Log-Gabor based enhancement technique [32] has two phases. In first phase, fingerprint image is divided into overlapping blocks of size m x m and then Fourier transform is computed for each block. In second phase, each block is multiplied with a Log-Gabor filter and then its inverse Fourier transform is computed to obtain spatial domain enhanced block. The computation complexity of each step is given below.

a) Computations for FFT for a block of size *m* x *m:*

Multiplication: $m^2/2\,\log_2(m^2)$

Addition: $m^2\,\log_2(m^2)$

Computations for FFT for a total of 2*M/m* and 2*M/m* block*:*

Multiplication: $2M^2\,\log_2(m^2)$

Addition: $4\,M^2\,\log_2(m^2)$

b) Computations for filtering a block in Frequency domain:

Multiplications: (m^2)

Addition: 0

Computations for filtering for a total of 2*M/m* and 2*M/m* blocks:

Multiplications: $4M^2$

Addition: 0

Computations for load operations:

It should be noted that for every block in the fingerprint image, a new filter may require to be loaded into the memory. Therefore, the total number of load operations required for filtering an image of size M x M with block size m x m can be given as:

Load / Memory operations: $4M^2$

c) **Computations for IFFT for a block of size *m* x *n:***

Multiplication: $m^2/2 \log_2 (m^2)$

Addition: $m^2 \log_2 (m^2)$

Computations for IFFT for a total of 2*M/m* and 2*M/m* block*:*

Multiplication: $2M^2 \log_2 (m^2)$

Addition: $4M^2 \log_2 (m^2)$

Total Computation required for filtering:

Multiplication: $4M^2 (1+\log_2 (m^2))$

Addition: $8M^2 \log_2 (m^2)$

Load / Memory operations: $4M^2$

Total Operations: $[4M^2 (1+\log_2 (m^2)), 8M^2 \log_2 (m^2), 4M^2]$

Total Computation required for Log-Gabor based enhancement [32]:

Multiplication: Multiplication operations for frequency estimation + Multiplication operations for filtering

Multiplication: $4M^2(1+ 1/m(4m + 1/2 \log_2 (m) + 1)) + 4M^2 (1+\log_2 (m^2))$

Addition: Addition operations for frequency estimation + Addition operations for filtering

Addition: $2M^2(3+ 2/m(3m + \log_2 (m) +2)) + 8M^2 \log_2 (m^2)$

Load / Memory operations: $4M^2$

Total Operations for Wang's Log-Gabor based enhancement is given as:

$$\left\langle \begin{array}{l} \left\{4M^2\left(1+1/m\left(4m+1/2\log_2(m)+1\right)\right)+4M^2\left(1+\log_2(m^2)\right)\right\}, \\ \left\{2M^2\left(3+2/m\left(3m+\log_2(m)+2\right)\right)+8M^2\log_2(m^2)\right\}, \\ \left\{4M^2\right\} \end{array} \right\rangle \tag{B.7}$$

B.4 Computational complexity of the proposed two-fold enhancement technique

In this subsection, the computational complexity of the proposed two-fold enhancement technique is calculated. As discussed in the proposed enhancement technique in section 3.1 the algorithm involves processing in spatial as well as frequency domains. Therefore, the total computational complexity is the summation of operations involved both in spatial and frequency domain.

Step 1: Computational complexity in frequency domain:

i) Computations for FFT for image of size *M* x *M:*

Multiplication: $(M^2)/2 \log_2 (M^2)$
Addition: $(M^2) \log_2 (M^2)$

ii) Computations for filtering in frequency domain:

Since in the proposed technique *n* band pass filters are used. Therefore, the total operations required for filtering will be:

Multiplications: $n\,(M^2)$

Addition: 0

Computations for loading:

It should also be noted that for one fingerprint image, *n* different band-pass filter images are loaded into the memory. Therefore, the total number of load operations required for filtering an image of size *M* x *M* can be given as:

Load / Memory operations: $n\,M^2$

iii) Computations for IFFT for *n* filtered images of size *M* x *M:*

Since there are n band pass filtered images. Therefore, the total computations for taking inverse Fourier transform for n band pass filtered images will be:

Multiplication: $n\ (M^{2})/2 \log_{2} (M^{2})$

Addition: $n\ (M^{2}) \log_{2} (M^{2})$

iv) Computations of unified band-pass filtered image of size *M* x *M:*

Computations of energy for a block of size m x m:

Multiplications: 0

Addition: m^2

Computations of unified band-pass filtered image of size *M* x *M*:

Multiplications: 0

Addition: $4M^2/m^2\ (m^{2}) = 4M^2$

Total computations in frequency domain:

Multiplications: $(M^2)/2 \log_{2} (M^2) + n\ (M^2) + n\ (M^{2})/2 \log_{2} (M^{2})$

Addition: $(M^2) \log_{2} (M^2) + n\ (M^{2}) \log_{2} (M^{2}) + 4M^2$

Total Operations in frequency domain are given as:

$$\left\langle \begin{array}{l} \{M^2/2\log_2(M^2) + nM^2 + nM^2/2\log_2(M^2)\}, \\ \{M^2\log_2(M^2) + nM^2\log_2(M^2) + 4M^2\}, \\ \{nM^2\} \end{array} \right\rangle \tag{B.8}$$

Step 2: Computations of spatial domain filtering:

i) Computations for rotation of 1D Gaussian filter of size *g* for image of size *M* x *M*:

Multiplications: $M^2/m^2\ (4g)$

Addition: $M^2/m^2\ (2\ g\ )$

ii) Computations for convolution of 1D Gaussian filter *g* with block of size *m* x *m* for image of size *M* x *M* :

Multiplications: $M^2/m\ (g + m)$

Addition: $M^2/m\ (g + m)$

Total computations in spatial domain:

Multiplications: $M^2/m^2\ (4g) + M^2/m\ (g + m)$

Addition: $M^2/m^2\ (2g) + M^2/m\ (g + m)$

Total Operations: $[M^2/m^2\ (4g) + M^2/m\ (g + m)\ ,\ M^2/m^2\ (2g) + M^2/m\ (g + m)\ , 0]$

Total Computational complexity of the proposed two-fold enhancement

Multiplications: $(M^2)/2 \log_2 (M^2) + n\,(M^2) + n\,(M^2)/2 \log_2 (M^2) + M^2/m^2\ (4g) + M^2/m\ (g + m)$

Additions: $(M^2) \log_2 (M^2) + n\,(M^2) \log_2 (M^2) + 4M^2 + M^2/m^2\ (2g) + M^2/m\ (g + m)$

Load Operations: $n\,M^2$

Total Operations required for the proposed two-fold enhancement:

$$\left\langle \begin{matrix} \left\{M^2/2\log_2(M^2) + nM^2 + nM^2/2\log_2(M^2) + M^2/m^2(4g) + M^2/m(g+m)\right\}, \\ \left\{M^2\log_2(M^2) + nM^2\log_2(M^2) + 4M^2 + M^2/m^2(2g) + M^2/m(g+m)\right\}, \\ \left\{nM^2\right\} \end{matrix} \right\rangle \quad \text{(B.9)}$$