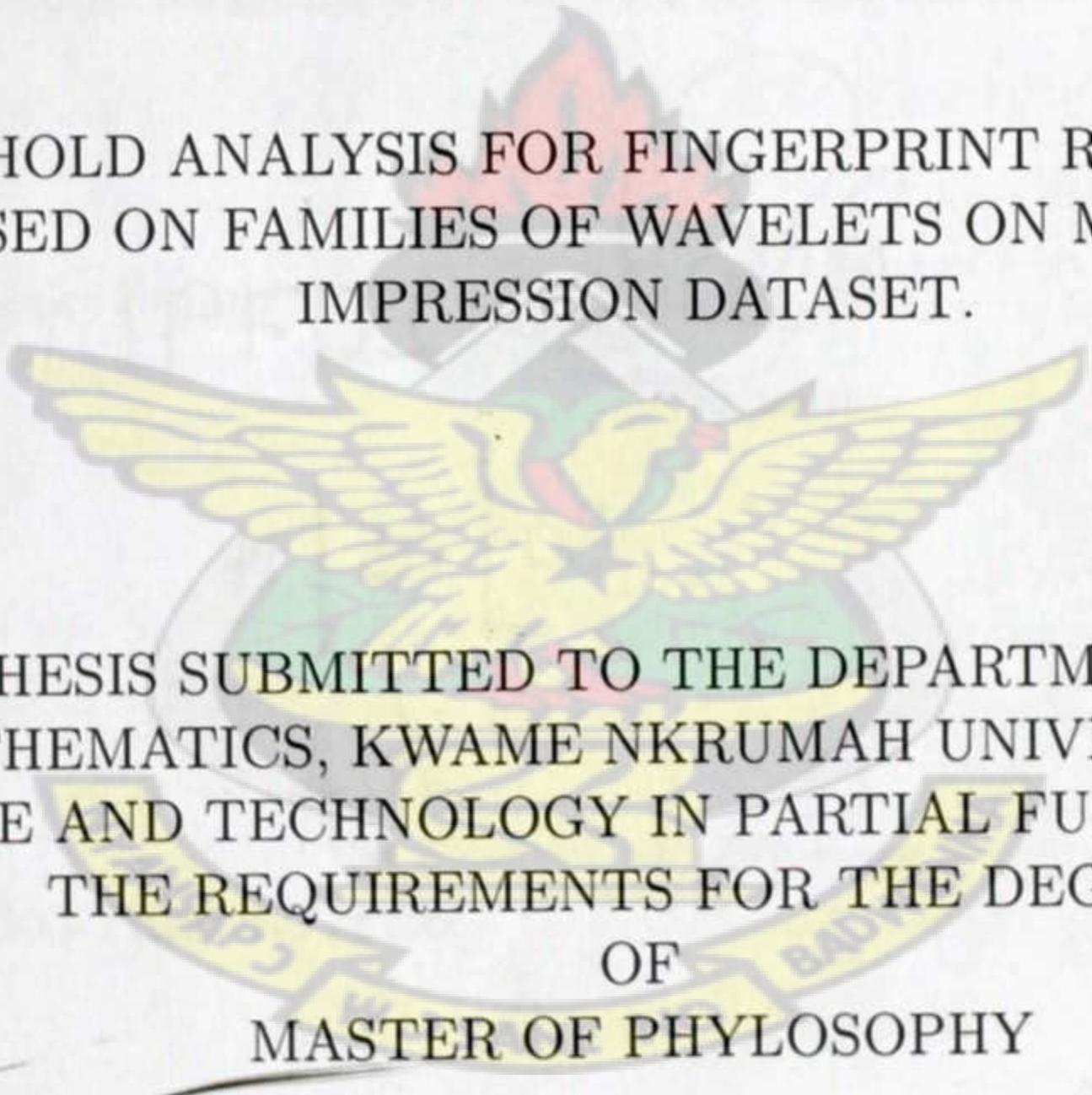


KWAME NKRUMAH UNIVERSITY OF SCIENCE AND  
TECHNOLOGY, KUMASI  
COLLEGE OF SCIENCE

DEPARTMENT OF MATHEMATICS

THRESHOLD ANALYSIS FOR FINGERPRINT RECOGNITION  
BASED ON FAMILIES OF WAVELETS ON MULTIPLE  
IMPRESSION DATASET.



THESIS SUBMITTED TO THE DEPARTMENT OF  
MATHEMATICS, KWAME NKRUMAH UNIVERSITY OF  
SCIENCE AND TECHNOLOGY IN PARTIAL FULFILMENT OF  
THE REQUIREMENTS FOR THE DEGREE  
OF  
MASTER OF PHILOSOPHY

BY  
APPATI JUSTICE KWAME  
MAY, 2013



## DECLARATION

I hereby declare that, this thesis is the result of my own original research and that no part of it has been submitted to any institution or organization anywhere for the award of a degree. All inclusion for the work of others has been dully acknowledged.

APPATI, Justice Kwame

PG6202811



Signature

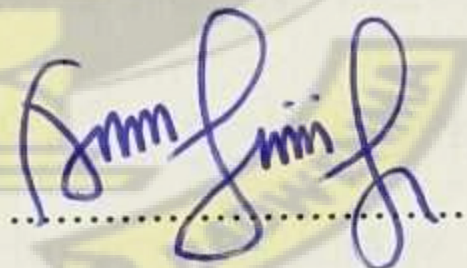
14<sup>th</sup> May, 2013

Date

Certified By:

Dr. AMOAKO-YIRENKYI Peter

Supervisor



Signature

14/05/13

Date

Mr. DARKWAH Kwaku Foukour

Head of Department

Signature

Date

LIBRARY  
KWAME NKRUMAH  
UNIVERSITY OF SCIENCE & TECHNOLOGY  
KUMASI



## ABSTRACT

After many years of using fingerprint as a biometric, a complete and accurate characterization and subsequent recognition of person using fingerprint still remain a problem due to perhaps how fingerprint features are extracted. The way a person's finger is scanned may create different impressions making it difficult to use one type of feature to characterize and identify a person by his finger leading to poor recognition rate. Some of such features are the local dominant orientation and edge parameters. Extraction of unique features for fingerprint recognition requires many algorithms but the effective one still remains a problem. Fingerprint has numerous features and this make extraction difficult. The solution is to use perhaps an efficient algorithm that can robustly combine several feature vectors into a single feature vector for recognition. In this work, the reverse bi-orthogonal wavelet family in 3-levels of resolution was applied to the fingerprint image and information in the vertical, diagonal and horizontal directions extracted and concatenated to form the feature vectors for the recognition. A new Equal Error Rate(EER) of 0.0464 at 95% recognition rate on the respective dataset was recorded as an improvement over the existing methods.



TABLE OF CONTENTS

DECLARATION . . . . . ii

ABSTRACT . . . . . iii

TABLE OF CONTENT . . . . . iv

LIST OF TABLES . . . . . vii

LIST OF FIGURES . . . . . x

DEDICATION . . . . . xiii

ACKNOWLEDGEMENT . . . . . xiv

CHAPTER

I. INTRODUCTION . . . . . 1

1.1 Background . . . . . 1

1.2 Problem Statement . . . . . 3

1.3 Objective . . . . . 4

1.4 Methodology . . . . . 4

1.5 Justification of the Study . . . . . 5

1.6 Organization of Chapters . . . . . 6

II. LITERATURE REVIEW . . . . . 8

2.1 Introduction . . . . . 8

2.2 Related Work . . . . . 8

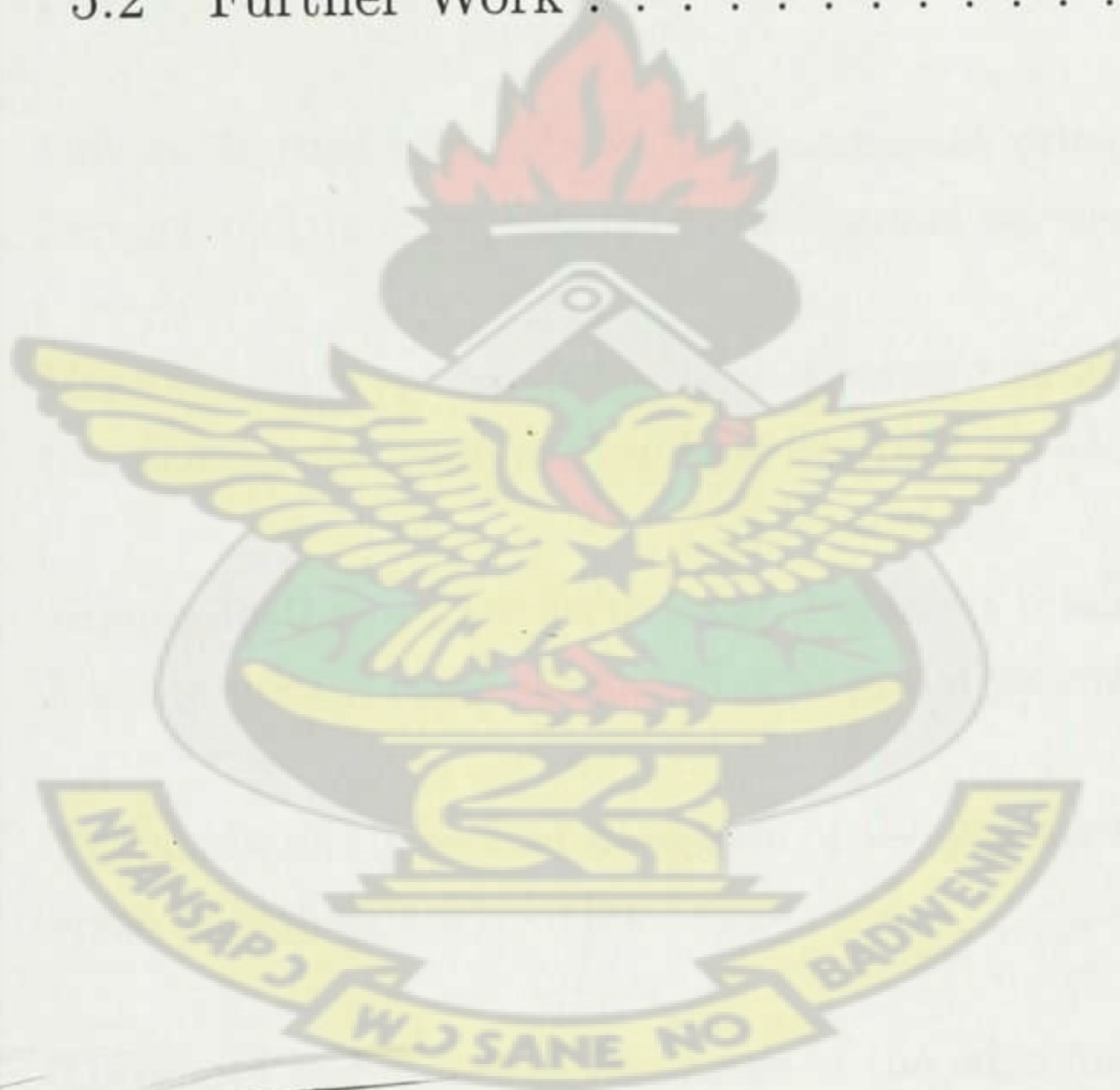
2.3 Definition of Some Mathematical Concepts Used 22



2.3.1	Vector Spaces . . . . .	22
2.3.2	Inner Product Spaces . . . . .	24
2.3.3	Fourier Basis and Transform . . . . .	28
2.3.4	Wavelet Basis and Transforms . . . . .	32
2.3.5	Discrete Wavelet and Multiresolution Analysis . . . . .	35
<b>III. METHODOLOGY . . . . .</b>		<b>39</b>
3.1	Introduction . . . . .	39
3.2	Image Processing . . . . .	40
3.2.1	Introduction . . . . .	40
3.2.2	Image Formation . . . . .	41
3.2.3	Image Processing Techniques . . . . .	42
3.2.4	Point Processing Operators . . . . .	45
3.2.5	Thresholding . . . . .	46
3.2.6	Histogram Equalization . . . . .	47
3.2.7	Intensity Histogram . . . . .	49
3.2.8	Local Enhancement . . . . .	50
3.2.9	Image Analysis . . . . .	51
3.2.10	Mathematical Morphology . . . . .	54
3.2.11	Digital Filters . . . . .	57
3.2.12	Edge Detectors . . . . .	60
3.2.13	Distance Transform . . . . .	69
3.2.14	Image Synthesis . . . . .	71
3.3	Analysis of Fingerprint Data . . . . .	72
3.4	Focus of the Study . . . . .	83
3.4.1	Overview of the Method . . . . .	83
3.5	Fingerprint Feature Extraction and Methods In- volved . . . . .	87
3.6	Fingerprint Recognition . . . . .	90
3.6.1	Performance of the Recognition Systems	91
3.6.2	Matching Approach . . . . .	93
<b>IV. ANALYSIS OF FEATURE EXTRACTION AND RECOGNITION . . . . .</b>		<b>95</b>



4.1	Introduction . . . . .	95
4.2	Source of Data . . . . .	95
4.3	Fingerprint Feature Analysis . . . . .	97
4.4	Analysis of Fingerprint Recognition . . . . .	106
4.5	Analysis of Choice of Threshold . . . . .	117
4.6	Comparison with Related Work . . . . .	118
<b>V. CONCLUSIONS AND RECOMMENDATION</b>		<b>119</b>
5.1	Conclusion . . . . .	119
5.2	Further Work . . . . .	120





## LIST OF TABLES

### Table

4.1	Matrix Representation of Fingerprint 1_1.tif . . . . .	98
4.2	List of all the Wavelet Transform Families Used. . . . .	99
4.3	Unique feature vector from local dominant orientation and co- herence matrix using reverse bi-orthogonal wavelet 3.1 . . . .	102
4.4	Unique feature vector from local dominant orientation and co- herence matrix using reverse bi-orthogonal wavelet 3.9 . . . .	102
4.5	Unique feature vector from local dominant orientation and co- herence matrix using reverse bi-orthogonal wavelet 4.4 . . . .	103
4.6	Threshold and standard deviation of the edge detector using re- verse bi-orthogonal wavelet 3.1 . . . . .	103
4.7	Threshold and standard deviation of the edge detector using re- verse bi-orthogonal wavelet 3.9 . . . . .	104
4.8	Threshold and standard deviation of the edge detector using re- verse bi-orthogonal wavelet 4.4 . . . . .	104
4.9	Feature vector representation of the center area using the reverse bi-orthogonal wavelet 3.1 . . . . .	105
4.10	Feature vector representation of the center area using the reverse bi-orthogonal wavelet 3.9 . . . . .	105



4.11	Feature vector representation of the center area using the reverse bi-orthogonal wavelet 4.4 . . . . .	105
4.12	FAR, FRR and TSR for various threshold values using the reverse bi-orthogonal wavelet 3.1 . . . . .	112
4.13	FAR, FRR and TSR for various threshold values using reverse bi-orthogonal wavelet 3.9 . . . . .	114
4.14	FAR, FRR and TSR for various threshold values using reverse bi-orthogonal 4.4 wavelet. . . . .	114
4.15	Summary of Families of Mother Wavelet used with their corresponding EER values . . . . .	115
4.16	Summary of Families of Mother Wavelet used with their corresponding EER values (Continued) . . . . .	116
4.17	Summary of Families of Mother Wavelet used with their corresponding EER values (Continued) . . . . .	116
4.18	Summary of Families of Mother Wavelet used with their corresponding EER values (Continued) . . . . .	116
4.19	Summary of Families of Mother Wavelet used with their corresponding EER values (Continued) . . . . .	116
4.20	Summary of Families of Mother Wavelet used with their corresponding EER values (Continued) . . . . .	116
4.21	Summary of Families of Mother Wavelet used with their corresponding EER values (Continued) . . . . .	116
4.22	Comparison between the three selected mother wavelet. . . . .	117



4.23	Experimental Results on Database DB3_A of FVC2004 . . . .	118
------	---	-----

KNUST





## LIST OF FIGURES

### Figure

3.1	Sobel convolution kernel . . . . .	63
3.2	Pseudo-convolution kernels for computing approximate gradient magnitude . . . . .	64
3.3	Prewitt gradient edge detector . . . . .	65
3.4	Prewitt compass edge detecting templates sensitive to edges at $0^\circ$ and $45^\circ$ . . . . .	67
3.5	Examples compass edge detecting kernels with each showing two kernels out of the eight . . . . .	68
3.6	Four line detection kernels which respond maximally to horizontal, vertical and oblique single pixel wide lines . . . . .	69
3.7	The distance transform of a simple shape using chessboard distance metric . . . . .	70
3.8	Example of a real $3 \times 3$ fingerprint image block site: paralleled ridge pattern. . . . .	78
3.9	A four overlapping neighborhoods $D_1, D_2, D_3, D_4$ of $3 \times 3$ block with target block $V$ in the center. . . . .	80
3.10	A patched site where the target block is at an edge. . . . .	82



3.11	Fingerprint Wavelet Decompostion. . . . .	87
3.12	Coherence Image of the Approximation image and Horizontal Details. . . . .	88
3.13	Core Point Position in Image. . . . .	90
3.14	Determination of False Acceptance Rate (Impostor Score) . . .	92
3.15	Determination of False Rejection Rate (Genuine Score) . . . .	93
3.16	Estimation of Equal Error Rate (EER) value . . . . .	93
4.1	The eight impressions of the 100th volunteer showing different variance. . . . .	97
4.2	Directional characteristics of the wavelet transform at level one of resolution of fingerprint 1_1.tif Row one is the Approximation and Diagonal Details whiles Row is the Horizontal and Vertical Details . . . . .	99
4.3	Gradient in the direction of x and y respectively. . . . .	100
4.4	Orientation of the fingerprint after extracting the Gradient in the x and y direction . . . . .	100
4.5	Magnitude of Gradient of the fingerprint . . . . .	101
4.6	Coherence image of the fingerprint after applying reverse bi-orthogonal wavelet 3.1 . . . . .	101
4.7	Level one feature extration using reverse bi-orthogonal wavelet 3.1 . . . . .	107
4.8	Level two feature extration using reverse bi-orthogonal wavelet 3.1 . . . . .	107

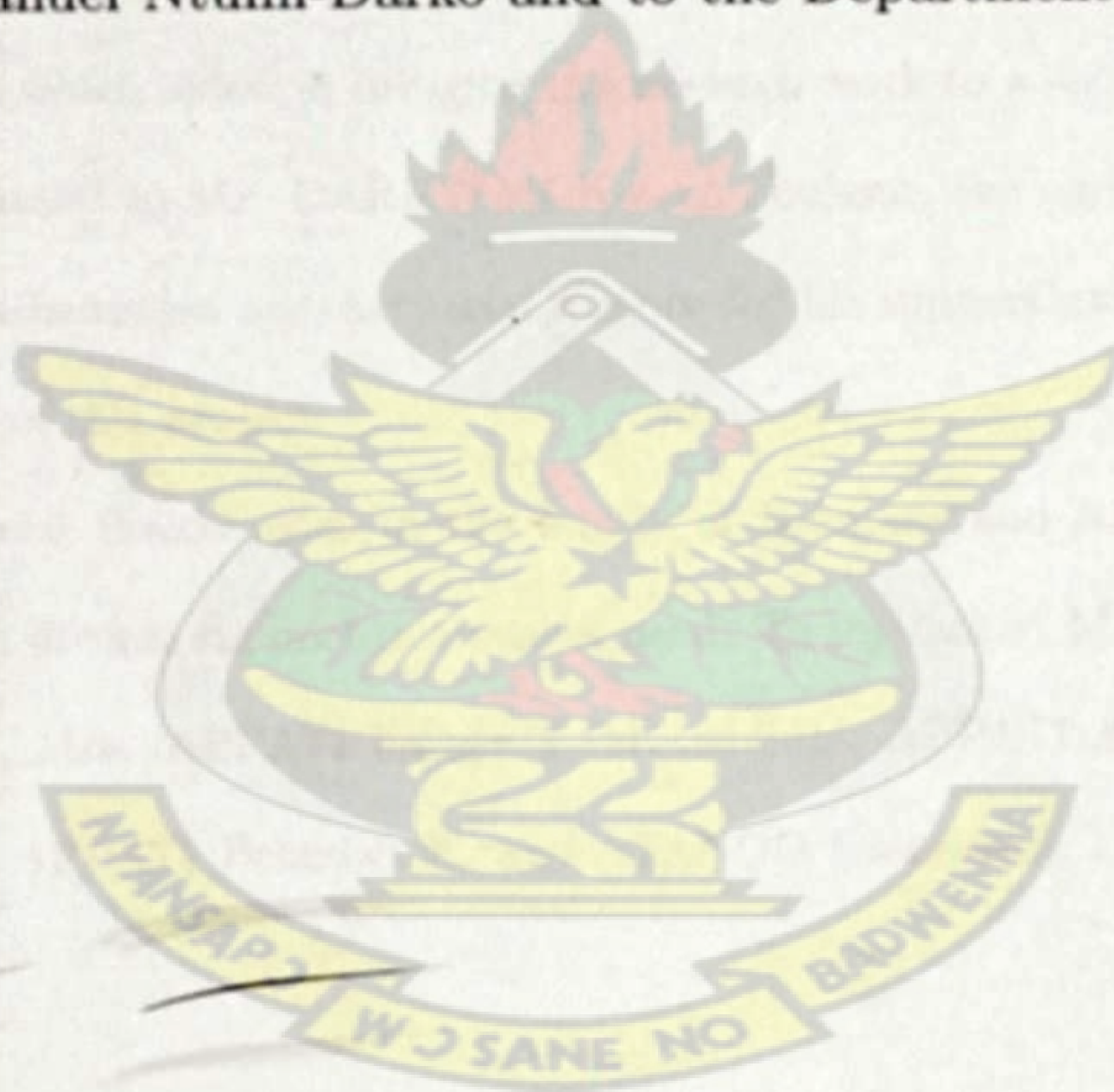


4.9	Level three feature extration using reverse bi-orthogonal wavelet 3.1 . . . . .	108
4.10	Level one feature extration using reverse bi-orthogonal wavelet 3.9 . . . . .	109
4.11	Level two feature extration using reverse bi-orthogonal wavelet 3.9 . . . . .	109
4.12	Level three feature extration using reverse bi-orthogonal wavelet 3.9 . . . . .	110
4.13	Level one feature extration using reverse bi-orthogonal wavelet 4.4 . . . . .	110
4.14	Level two feature extration using reverse bi-orthogonal wavelet 4.4 . . . . .	111
4.15	Level three feature extration using reverse bi-orthogonal wavelet 4.4 . . . . .	111
4.16	Graph of variation of FAR and FAR using rbio3.1 wavelet. . .	113
4.17	Graph of variation of FAR and FAR using reverse bi-orthogonal 3.9 wavelet. . . . .	113
4.18	Graph of variation of FAR and FAR using reverse bi-orthogonal 4.4 wavelet. . . . .	115



## DEDICATION

This thesis is dedicated to my father, Mr. Ebenezer K. Appati, mother, Mrs. Esther Y. M. Appati, my siblings, Adelaide A. Appati, Walter K. Appati, Benedict K. Appati and Emmanuel Atando, my loved one Francisca Nana Adjoa Fordjour Herman Amponsah, Pastor Emmanuel Ntumi-Darko and to the Department of Mathematics, KNUST.





## ACKNOWLEDGEMENT

First and foremost, I give thanks to the Almighty God for giving me the strength and knowledge to go through this studies. My heartfelt gratitude goes to my Supervisor, Dr. AMOAKO-YIRENKYI Peter for his patience, guidance and keen interest in my work which aided in bringing my research work to a successful completion. I am also grateful to Mr. DARKWAH Kwaku Foukour, the current Head of Department of Mathematics and Actuarial Science for his support and advice towards the completion of this project.

My special thanks goes to the family of Hermans and Ntumi-Darko for their prayers and diverse support in times of need. To my father, Mr. APPATI Ebenezer K., mother, Mrs. APPATI Esther Y. M., siblings, APPATI Adelaide A., APPATI Walter K., APPATI Benedict K. and ATANDO Emmanuel, I say God richly bless you.



# CHAPTER I

## INTRODUCTION

KNUST

### 1.1 Background

The authentication or identification of persons based on the measure and analysis of their physiological or behavioral characteristic is termed as biometrics (Prabhakar et al., 2003; Chouhan and Khanna, 2011). Among the most widely used biometric factors are: iris, spatial geometry of the face, fingerprint and voice patterns. Physical access control to secured areas and materials is generally based on locks and keys which can easily get lost or stolen by malicious individuals (Zhang, 2000). Because access is based on locks and keys, the authentication factor is "something you have" which provide no guarantee that the person entering or accessing the material is the actual individual granted the access. The same is for computerized systems where access control is mostly based on passwords hence making the authentication factor "something you know". Unfortunately, the user is given the responsibility to memorize such an amount of password or pin code making them apply password policy practice badly by writing down password, keeping password as simple as possible and even to the extent of using the same password always.

A solution to the above is the biometric, which has its access control factor as "something you are" and provides better security. This is because, the measure



of physiological or behavioral characteristic is often more difficult to forge, steal or imitate as is the case of passwords or keys. Again, users do not need to remember it and they cannot by accident leave it at home. These merit posed by biometric, make its demand higher as a tool for verification and authentication purposes.

In theory, humans can be identify with many features that can be used as a biometric for authentication or identification. However, such features need to be universal (ie. something everybody has), unique (ie. possible to separate individuals from another), easy to collect and not easily imitable (Prabhakar et al., 2003). For these reasons, fingerprint as a biometric tool is mostly considered.

In many forensic research and application, fingerprint is known to be unique and reliable from the biological point of view due to some profound characteristic that it posses (Qinghui and Xiangfei, 2010). These fact has really aided most researchers for many years in person identification and authentication and it still remains one of the oldest and most popular. Although most researchers have dealt a lot in this field, the issue of attaining a higher accuracy still remains a problem due to perhaps how the feature are extracted or how the data were collected.

As a result of the stated issues, many pattern recognition methods which generally involve statistical pattern recognition, feature extraction, cluster analysis and image processing and analysis have been used in many areas including voice recognition, face recognition, fingerprint recognition and even gait recognition (Zhang, 2000).

The motivation for this research actually arise from machine learning and computer vision studies, which provide more evidence that, fingerprint image taken at different impressions contain a signature that is unique to each individual (Pankanti et al., 2002) even to the extent that twins having resembling DeoxyriboNucleic Acid (DNA) are believed to have different fingerprint image. These indeed has lead to an



increase use of automatic fingerprint verification system, which has evolve in its establishment in many disciplines such as: in the military, financial institution, airport and governmental organization where security is a major requirement to overcome possible threats.

## 1.2 Problem Statement

It is well known that, solutions to fingerprint registration, feature extraction and characterization and matching are major problems encountered by almost all previously proposed techniques for fingerprint recognition (Yan et al., 1997). The ability to identify a person by his fingerprint while avoiding impressions made during fingerprint registration especially based on minutia algorithm is also a major issue. For example, people with no or few minutia points such as surgeons who often wash their hands with strong detergents and people with special skin conditions will have low recognition rate. Hence, the number of minutia points becomes a limiting factor for security response using the minutia algorithm. Again, results from minutia algorithm can be confused with false minutia points (i.e areas of fudging that appears due to low-quality enrollment of fingerprint ridge details).

Generally, during scanning of fingerprints for recognition purposes, the following observations are made (Dechman, 2012):

1. Different scanners for fingerprint registration make different but significant impressions on the fingerprint images and these impressions actually affect the recognition rate.
2. The choice of threshold during matching of fingerprint against known fingerprint database also affect recognition rate.



3. Small fingerprint scanners requires significantly low complexity based methods due to low processing power used by these devices.

The fundamental problem of this thesis, is to find a significantly low complexity based method, that can robustly and accurately extract features, across multiple impressions based dataset of fingerprint in order to increase recognition rate.

### 1.3 Objective

The objectives of this thesis are to:

1. Identify the best wavelet transform among the lot which can robustly extract fingerprint features more uniquely across multiple impression based dataset.
2. Define a common threshold value across the multiple impression dataset in order to identify the best windowing technique.
3. Increase the recognition rate with the chosen wavelet and its corresponding threshold value irrespective of the impressions made by different scanners.

### 1.4 Methodology

In order to address the fundamental problems of this thesis, an efficient mathematical tool for multiple resolution analysis was considered. Wavelet transform, as a method of choice was considered due to the following advantages over its peers as identified by Sifuzzaman et al. (2009). These are:

1. They provide simultaneous localization in time and frequency domain.
2. They are computationally very fast.
3. They can be used to decompose a signal into their component wavelets.



4. Finally, they have the capability of revealing aspects of data like trends, break-down points and discontinuities.

Below are the laid down methodology used to achieve these objectives.

1. Acquisition of both source and test database used for this study was from the standard database named Fingerprint Verification Competition 2004 (FVC2004).
2. Wavelet Transform from different wavelet families was used to extract different but unique features from the fingerprint image.
3. The fingerprint matching between the test and source database against several predefined threshold was performed using the Euclidean Distance Transform algorithm in order to identify the best predefined threshold.
4. The False Acceptance and Rejection Rate (i.e. FAR and FRR) were calculated from the resulting matching algorithm.
5. The performance of the method was analyzed using the Receiver Operating Characteristic (ROC) curve based on the Equal Error Rate (EER) performance indicator.

### **1.5 Justification of the Study**

The design of fingerprint recognition system base on our proposed method will help increase the security of the nation in the following areas;

#### **1. LAW ENFORCEMENT AGENCIES**

Application of the fingerprint recognition system in the law enforcement agencies such as the court or police station can successfully execute an arrest or delivery as soon as the identity of the individual is verified.



## 2. ACCESS CONTROL

Currently, fingerprint based access control devices have certain disadvantages in its usability. Often a user may need to repeatedly scan their finger before they are granted access. These may be caused by inconsistencies between the fingerprint data recorded by the capture device and the data stored within the systems database. This inconsistency increases the systems margin of error which increase false rejections and a lower degree of confidence with every match. In high security access control, an additional measure can be taken to further increase the degree of confidence with every match as resolved in this study.

## 3. FINANCIAL TRANSACTION

In the commercial sector, accurate biometric based authentication can be implemented in electronic commerce. Methods of authentication such cards, passwords and pins are widely being used today. Biometric methods can be supplemented by accurate fingerprint based authentication to obtain a higher degree of user confidence, as well as decrease the presence of fraud in online transaction.

## 1.6 Organization of Chapters

The thesis is divided into five chapters.

The first chapter is an introduction and discussion of some underlying scientific and historical facts about fingerprint recognition. It take into account the problem in the field of fingerprint recognition and a gist of methods to be adopted to solving the problem at hand including the chapter organization.

The second chapter gives the literature review of the techniques and approaches



that describes signal and image analysis as far as fingerprint is concerned. The chapter further reviewed some mathematical concepts used through out the study.

The third chapter, which is the methodology, explains the theory and concepts which form the basis of the thesis. This deals with how digital filters, image algebra, image transform and some discussion on mathematical morphology in image processing and pattern recognition. It also explains how wavelet transforms are formulated and used to achieve higher accuracy in fingerprint image recognition.

The fourth chapter basically discuss the data used in the thesis and analysis as well as experimental results from the Fingerprint Verification Competition (FVC 2004) database was discussed. A study was also made on the current methods used in fingerprint recognition whiles some useful technique found were employed in carrying on with the research. Finally, methods to extracting features relevant to this work was illustrated.

Chapter Five concludes the work with some recommendations made on the relevance of the undertaken project.



## CHAPTER II

# LITERATURE REVIEW

KNUST

### 2.1 Introduction

Every fingerprint possesses structural patterns which have very rich information in them making it possible to be used for fingerprint recognition and these patterns can be categorized into Global and Local features. As discussed in the introductory chapter, there are several methods but the most widely used are the Correlation, Minutiae and Ridge Feature based methods for fingerprint recognition. This chapter first reviews some related studies on the respective subject and try to elaborate on the methods and techniques which have been used throughout this work. It also reviews some related fundamental mathematical concepts that are used in establishing these methods.

### 2.2 Related Work

In pattern recognition, the representation of patterns can be considered as feature extraction and is divided into four groups; statistical pixel features, algebraic features, visual features and transform coefficient features (Hong, 1991). The following provide a short description of some of these techniques:

1. Minutiae based technique: These are the ridge ending and ridge bifurcation



of a fingerprint and are represented by the location  $(x, y)$  and orientation  $\theta$ . However a preprocessing on fingerprint image is required to extract a true minutia which is time consuming.

2. Image based technique: Thus where fingerprint preprocessing at optimal and spatial domain are processed to generate features using mean, standard deviation, variance, energy, gradient and directional features.
3. Transform domain technique: Here, fingerprint image is converted into frequency domain using Fast Fourier Transform (FFT), Discrete Cosine Transform (DCT), Dual Tree Complex Wavelet Transform (DTCWT) and Discrete Wavelet Transform (DWT) to generate the feature vectors.
4. Multiple Classifier Technique: This is where test fingerprint features are compared with features in the database using multiple classifiers such as Euclidean Transform (ED) and Support Vector Machine (SVM).

Khan et al. (2009) in his research presented a new approach which uses wavelet based features fused with minutiae based features for matching purpose. This is because the minutiae based approach requires very lengthy preprocessing during minutiae extraction, however the result still contains false minutiae serving as a drawback. Although previous attempts were made to overcome this challenge by performing post-processing on the fingerprint image yet led to elimination of some valid minutiae along with false ones. Hence they concluded that the strength of this matching algorithm depends highly on the strength of features extracted. This features extraction can either be made directly from the gray-scaled image or from a thinned image and the resulting values matched which also depends on the best matched minutiae pairs from both images. Different kinds of features are extracted



from extracted minutiae which allows us to have rotation and translation invariance and this was the contribution made. It was found that, among the algorithms studied, their proposed work gained significant effects on overall performance which was revealed from their experimental results. It was also realized that using these features make the matching process much more accurate even in the presence of false minutiae.

In the same year, Bhowmik et al. (2009) in this paper, proposed a Euclidean distance based minutia matching algorithm to further improve the matching accuracy in fingerprint verification system by extracting the matched minutia pairs from test and template fingerprints image by using the smallest minimum sum of closest Euclidean distance (SMSCED) corresponding to rotation angle and empirically chosen statistical threshold values. Their proposed algorithm uses only the minutia location instead of using the minutia type and orientation angle as widely employed in existing algorithms to reduce the effect of non-linear distortion. From the experimental results, he concluded that the proposed method has higher accuracy with improved verification rate and rejection rate.

At the year that follows, an article by Qinghui and Xiangfei (2010) made a systematic elaboration on the correlation theories of fingerprint recognition technology and several essential algorithms. The study shows that combining the characteristic of fingerprint image in the pretreatment process is a key technology especially taking into consideration fingerprint image intensification which is based on the gradation standardization, directional diagram and filters that in turns uses the auto-adapted algorithm which is also based on partial smooth threshold value to carry on with binarization. The experimental result indicates that the processing effect is good when it come to fingerprint recognition and verification.



Conti et al. (2010) in their study reckon to the fact that fingerprint classification and matching are the two key issues in automatic fingerprint recognition. They noted that, fingerprint recognition is based on sets of relevant local characteristics such as the ridge endings and bifurcations. It was clear that fingerprint classification is based on fingerprint global features such as the core and delta singularity points however it is unfortunate for these singularity points to be present in all fingerprint image hence the acquisition process is not ideal. This make classification of fingerprints that belong to the arch class very difficult. To this case, they proposed a pseudo singularity points algorithm which will detect and extract all possible singularity point to be used in fingerprint classification and matching. An experimental trials was conducted on the Fingerprint Verification Competition (FVC) databases and results show the effectiveness of the proposed method.

Chengming et al. (2009) acknowledge that, fingerprint matching is very important for automatic fingerprint recognition system (AFIS) but the problem is that, after local structure matching, one minutia in query fingerprint may have multiple candidate matching minutiae in template fingerprint and so getting the right one-to-one matching pairs has great impart on the performance of fingerprint matching algorithm. In the study they proposed a global fingerprint matching algorithm for acquiring a one-to-one matching pairs based on motion coherence which is very useful for fingerprint feature matching. Upon testing the algorithm on a public database FVC2002 DB1\_A and FVC2004 DB1\_A, the results proved that the proposed algorithm performed well compared with other ones.

At a point in time, Xuzhou and Yu (2009) proposed a minutiae matching algorithm that uses minutiae centered circular regions to help ensure the stability of matching and its robustness to non-linear distortion. In this method, a circular region



is constructed around each minutia, which can be regarded as a secondary feature. From the constructed regions, the proposed algorithm can find matched minutiae more rapidly via regional matching. This study shows the algorithm is more tolerant to non-linear distortion when compared to global matching approaches since each minutia region is formed from only a small area of the fingerprint. On the other hand, it was noted that, the algorithm gives a more reliable and distinct features when the area of the constructed region is much larger than that of local neighborhood in local matching approaches. Experimental results show the algorithm has better robustness and stability.

Khalil et al. (2010), in their paper presented a statistically analyzed biometric fingerprint images for personal identification where a sub-image of  $129 \times 129$  was extracted from the original image and transformed into a co-occurrence matrix. Application of four different types of relative positioned distance were used to generate the matrices and the results analyzed by the Program for Rate Estimation and Statistical Summaries (PRESS). The efficiency of this proposed method demonstrated that the further the distance of the relative position the lower the error equal rate.

Seung-Hoon et al. (2009) in their study understood that Fingerprint verification systems are very convenient and secured. However, existing systems can cause errors as a result of the size and quality of fingerprint images that are read only with the use of minutiae. In sorting solution to this problem, they proposed an enhanced security performance algorithm which has the ability to decrease the False Match Rate (FMR). In conclusion they found that, ZeroFMR decreased from 29% to 9.9%, which indicates enhanced security performance.

Zheng et al. (2009) also established from their studies that, fingerprint identity key and the challenge response of authentication based on biological feature requires



fingerprint matching algorithm based on minutiae with high accuracy and robustness. This lead to a proposal to introduce a novel fingerprint matching algorithm based on minutiae which does not rely on fingerprint global feature such as core points and ridge structure but uses a method of similar vector triangle to solve the key problem of point pattern matching algorithm. The experimental results indicated that, the proposed algorithm can be invariant to translation, rotation and deformation and also proves to be robust for fingerprint matching.

In situations where many fingerprint images are of poor quality, then one needs to expect a great damage on the performance of the automatic fingerprint identification system (AFIS). Attempt to ensure that performance is robust with respect to the quality of input fingerprint images requires an enhancement algorithm. Since Gabor filter is a very effective tool for image enhancement, Zhang and Jing (2010), took hold of the advantages of its characteristic in both spatial and spectral domain to dynamically estimate the gabor filter's parameters and bandwidth. Finally, the images were filtered in the frequency domain using the obtained gabor Frequency-domain filter function and experimental result shows that the images were greatly improved.

A new approach for fingerprint verification based on wavelets and pseudo Zernike moments (PZMs) was proposed by Pokhriyal and Sushma (2010) which is robust to noisy images, invariant to rotation and have a good image reconstruction capability. They used this method for global analysis and global feature extraction whiles they used the Wavelets for local analysis and local feature extraction from a fingerprint image. With this hybrid approach they extracted most significant features from the fingerprint images and achieve better verification rate. Different types of wavelets were used for the study but the result shows that Symmlet orthonormal wavelet of



order 8 gives best verification rate.

Ying et al. (n.d.) in their study, propose an effective fingerprint matching algorithm based on error propagation since the traditional methods treat problem as point pattern matching which is essentially an intractable problem due to the various nonlinear deformations commonly observed in fingerprint images. They adopted to the ridge information and Hough transformation to extract several pairs of matching minutiae from which the matched set, which includes the correspondence and its surrounding matched minutiae pairs are established. Here, the subsequent matching process is guided by the concept of error propagation: the matching errors of each unmatched minutiae are estimated according to those of its most relevant neighbor minutiae. In order to prevent the process from being misguided by mismatched minutiae pairs, they adopt a flexible propagation scheme and experimental results demonstrate that their algorithm is robust to non-linear deformations

Karu and Anil (1996) presented a fingerprint image classification algorithm which extracts singular points (cores and deltas) in a fingerprint image and performs classification based on the number and locations of the detected singular points. This make the classifier invariant to rotation, translation and small amount of scale changes. The classifier is rule-based, where the rules are generated independent of a given data set and was tested on 4000 images in the NIST-4 database and on 5400 images in the NIST-9 database. For the NIST-4 database, classification accuracies of 85.4% for the five-class problem and 91.1% for the four-class problem (with arch and tented arch placed in the same category) were achieved. Using a reject option, the four-class classification error can be reduced to less than 6% with 10% fingerprint images rejected.



Roddy and Stosz (1997) presents a model that defines the parameters necessary to estimate the performance of fingerprint authentication systems without going through the rigors of intensive system testing inherent in establishing error rates. The model was developed to predict the performance of the pore based automated fingerprint matching routine developed internally in research and development division at the National Security Agency. In addition they linked together the realms of automated matching and statistical evaluations of fingerprint features and their result provided knowledge of practical performance limits of any automated matching routine that utilizes pores or minutia features.

Alfredo et al. (n.d.) proposed a fast fingerprint authentication method based on the core and minutiae detection of the fingerprint. They also established the relationship between authentication reliability and region size during their experimental study. Application of bank of Gabor filters orientated at different angles was applied to the image to clean it from noises that can result in authentication mistakes. Their approach extracts the core using the flow field and determines the angle to which each vector of the flow field has with respect to the horizontal. From the core extracted, vectors will be trace to the minutiae for the purpose of image alignment and fingerprint matching.

Unlike the conventional minutiae matching algorithms, Ackerman and Ostrovsky (n.d.) propose an algorithm which takes into account region and line structures that exist between minutiae pairs which allows for more structural information of the fingerprint to be accounted for thus resulting in stronger certainty of matching minutiae. Evidence from the testing of the preprocessed images gives stronger assurance that using such data could lead to faster and stronger matches.



One problem besetting fingerprint matching is distortion which changes both geometric position and orientation, leads to the difficulties in establishing a match among multiple impressions acquired from the same fingerprint. Marking all the minutiae accurately as well as rejecting false minutiae is another issue still under research. Manvjeet et al. (2008) in their research combined many methods to build a minutia extractor and a minutia matcher. Also some novel changes like segmentation using morphological operations, improved thinning, false minutiae removal methods, minutia marking with special consideration of the triple branch counting, minutia unification by decomposing a branch into three terminations and matching in the unified  $x - y$  coordinate system after a two-step transformation are used.

In context, Elmir et al. (2009) proposed an algorithm development of the original image processing, minutiae and singular points localization where Gabor filter coding was tested on well known databases such as: FVC2004 databases and FingerCell database. Promising results using radial basis function neural network and support vector machine pushed them to continue the exploitation of new version of neural networks which is spike neural networks and to develop codification and recognition algorithms. Performance Evaluation has proved that, spike neural network achieved a good recognition rate closer to rates achieved by other methods but in a very short time and this make it more useful in online applications.

Singular points (including cores and deltas) not only represent the local ridge pattern characteristics but also determine the topological structure (i.e. fingerprint type) and the work of Jie et al. (2007) tries to analysis performance for singular points in two aspects.

1. Based on the topology theory in 2D manifold, they deduced the relationship between cores and deltas in fingerprints and also proposed a flexible method



to compute the Poincare Index for singular points.

2. Proposal of a novel algorithm for singular point detection using global orientation field after the initial detection with the widely used Poincare Index method. An optimal singular points are selected to minimize the difference between the original orientation field and the model based orientation field reconstructed from the singular points.

The core-delta relation was used as a global constraint for final decision and experimental results shows that their algorithm is accurate and robust.

A fingerprint verification system based on a set of invariant moment features and a nonlinear Back Propagation Neural Network (BPNN) verifier was proposed by Yang and Park (2008) which was used to overcome the demerits of traditional minutiae based methods and other image based methods. The proposed system contains two stages: an off-line stage for template processing and an on-line stage for testing with input fingerprints. Preprocessed fingerprints help detect a reliable unique reference point to determine a Region-of-Interest (ROI). A total of four sets of seven invariant moment features were extracted from four partitioned sub-images of the ROI. Matching between the feature vectors of a test fingerprint and those of a template fingerprint in the database is evaluated by a nonlinear BPNN and its performance is compared with other methods in terms of absolute distance as a similarity measure. The experimental results show that the proposed method with BPNN matching has a higher matching accuracy, while the method with absolute distance has a faster matching speed. Comparing results with other famous methods also show that the proposed method outperforms them in verification accuracy.



Saquib et al. (2011) also proposed a method for reliable detection of singular points which is largely insensitive to the degradation of fingerprint quality. The approach involves two phases, wherein, the first phase detects the singular points which operates on the quadrant change information. The second phase involves the analysis and extraction of the locations having high probability of the existence of singular points which is based on the orientation reliability measure of the filtered fingerprint image. This model tested on some selected noisy images from a publicly available (Cross Match Verifier 300 sensor) fingerprint database scanned at 500dpi shows that the approach effectively eliminates the spurious singular points in the noisy images.

Methods for deciding which part of an image belongs to the foreground and which part to the background was proposed by Xiang et al. (2009). Two kinds of pixel features, it being the coherence of direction and the variance of gray level were used in the segmentation where the fuzzy c-mean clustering algorithm was used to select threshold. Morphology technology is applied at post processing to obtain a smooth contour lines and experimental results demonstrate the effectiveness of the proposed method especially in low quality images in terms of including less background and excluding less foreground. In addition, this robust segmentation algorithm is capable of filtering efficiently spurious boundary.

Kingsbury (2006) describes a technique for using dual-tree complex wavelets to obtain rich feature descriptors of key points in images. The main aim was to develop a method for retaining the full phase and amplitude information from the complex wavelet coefficients at each scale whiles presenting the feature descriptors in a form that allows for arbitrary rotations between the candidate and reference image patches.



Kulwinder et al. (2011) presented a work on a new fingerprint feature detection algorithm to counter the presence of noise in fingerprint images leading to spurious minutiae. The proposed method was used in matching the template for finding bifurcation and termination. The new smoothing algorithm is proposed for the detection of the features of fingerprints. Finally, a method was introduced for finding ridges in the fingerprint image with the help of eight different masks. The results showed the accuracy of the algorithm in terms of genuine acceptance rate, false rejection rate and false acceptance rate.

Patil et al. (2006) describes the fingerprint verification based on wavelet transform and the local dominant orientation where Daubechies wavelet was use to decompose the fingerprint image. The local dominant orientation is computed using the coherence. The train and test fingerprint images are aligned around the core point while the core point is detected as a maximum curvature point. This lead to 85% genuine acceptance rate at 6% false acceptance rate (FAR).

An accurate estimation of fingerprint orientation fields is an essential step in the overall fingerprint recognition process but conventional gradient based approaches are popular but very sensitive to noise. Due to this conditions, Wang et al. (2007) proposed a novel implementation to improve the performance of gradient based methods. The enhanced algorithm chooses the best orientation estimate from four overlapping neighborhoods of every image block, where the voting scheme is based on the reliability measures. The experiment results suggest that the enhanced algorithm achieves better noise resistance with modest computation time in comparison with other gradient based methods.

Lavanya and Raja (2011) present a work on Performance Evaluation of Fingerprint Identification based on DCT (Discrete Cosine Transform) and DWT (Discrete



Wavelet Transform) using Multiple Matching Techniques. The fingerprint is segmented into four cells of size  $150 \times 240$  each. The DCT is applied on each cell while the Harr Wavelet is applied on DCT coefficient of each cells. The directional information features and center area features are computed on LL sub band with final Feature Vector being the concatenation of Directional Information and Center Area Features. The matching techniques viz, ED (Euclidean Distance) and SVM (Support Vector Machine) are used to compare test image feature with database image features. It was observed that the values of TSR (Total Success Rate) and FRR (False Rejection Rate) are better in the case of the proposed algorithm compared to existing algorithm.

In order to ensure that the performance of an automatic fingerprint identification system is robust with respect to the quality of input fingerprint images, it is essential to incorporate a fingerprint enhancement algorithm in the minutiae extraction module. To this, Hong et al. (1998) presented a fast fingerprint enhancement algorithm, which can adaptively improve the clarity of ridge and valley structures of input fingerprint images based on the estimated local ridge orientation and frequency. The performance evaluation of the image enhancement algorithm using the goodness index of the extracted minutiae and the accuracy of an online fingerprint verification system shows that incorporating the enhancement algorithm improves both the goodness index as well as the verification accuracy.

Little are the work done in providing detailed evaluation of methods for ridge distance estimation, in particular, the traditional spectral analysis method applied in the frequency field. Yilong et al. (2004) presented a novel method on non overlap blocks called the statistical method to estimate the ridge distance. Direct Estimation Ratio (DER) and Estimation Accuracy (EA) are defined and used as parameters



along with Time Consumption (TC) to evaluate performance of these two methods for ridge distance estimation. Based on comparison of performances of these two methods, a third hybrid method was developed to combine the merits of both methods. Experimental results indicate that DER was 44.7%, 63.8%, and 80.6%; EA as 84%, 93%, and 91%; and TC as 0.42, 0.31, and 0.34 seconds, with the spectral analysis method, statistical method, and hybrid method, respectively.

Most fingerprint enhancement algorithms rely heavily on local orientation of ridge flows however significant orientation changes occur around the delta and core points in the fingerprint images and this poses a challenge to the enhancement of ridge flows in those high-curvature regions. Instead of identifying the singular points, Chaohong and Govindaraju (2006) calculated an orientation coherence map and determined the minimum coherence regions as high-curvature areas. Gaussian filter window sizes were adaptively chosen to smooth the local orientation map. Because the smoothing operation is applied to local ridge shape structures, it efficiently joins broken ridges without destroying essential singularities and even enforces an increase in continuity of the directional fields. Experimental results demonstrate the effectiveness of the proposed method.

Finally, Mansukhani et al. (2007) says to compensate for the different orientations of two fingerprint images, matching systems should use a reference point and a set of transformation parameters. Fingerprint minutiae are compared with their positions relative to the reference points using a set of thresholds for the various matching features. However a pair of minutiae might have similar values for some of the features compensated by dissimilar values for others and this trade off cannot be modeled by arbitrary thresholds since this might lead to a number of false matches. Instead given a list of potential correspondences of minutiae points, we could use a



static classifier, such as a support vector machine (SVM) to eliminate some of the false matches. A 2-class model was built using sets of minutiae correspondences from fingerprint pairs known to belong to the same and different users. Using this recognizer reduces the number of false minutiae matches by 19%, while only 5% of the minutiae pairs corresponding to fingerprints of the same user are rejected.

## 2.3 Definition of Some Mathematical Concepts Used

In trying to establish an appropriate method that can robustly extract fingerprint features for recognition, it is always important to revise the fundamental mathematical concepts in order to choose the most efficient mathematical tools. Most fingerprint images are formatted as vector graphics and therefore there is the need to review vector spaces, their respective related algebra and some useful transformation that can help extract these features economically.

### 2.3.1 Vector Spaces

**Definition 2.3.1** (Vector Spaces). A vector space  $\mathbb{V}$  over  $\mathbb{C}$  is a set with two operators,  $+$  (vector addition) and  $\cdot$  (scalar multiplication), such that the following properties holds:

1. (Closure of  $+$ ): For any  $u, v \in \mathbb{V}$ ,  $u + v \in \mathbb{V}$
2. (Commutativity of  $+$ ):  $u + v = v + u$  for any  $u, v \in \mathbb{V}$
3. (Associativity of  $+$ ):  $(u + v) + w = u + (v + w)$  for all  $u, v, w \in \mathbb{V}$
4. (Additive Identity): There is an element of  $\mathbb{V}$  denoted by  $0$  such that  $v + 0 = v$  for all  $v \in \mathbb{V}$



5. (Additive Inverse): For every  $v \in \mathbb{V}$ , there is a  $u \in \mathbb{V}$  such that  $v + u = 0$ . We denote such  $u$  by  $-v$ .
6. (Closure of  $\cdot$ ): For every  $\alpha \in \mathbb{C}$  and  $v \in \mathbb{V}$ ,  $\alpha \cdot v \in \mathbb{V}$
7. (Behavior of 1):  $1 \cdot u = u$  for every  $u \in \mathbb{V}$
8. (Associativity of  $\cdot$ ):  $\alpha \cdot (\beta \cdot u) = (\alpha\beta) \cdot u$  for every  $\alpha, \beta \in \mathbb{C}$ ,  $u \in \mathbb{V}$
9. (Distributive Property 1):  $\alpha \cdot (u + v) = \alpha \cdot u + \alpha \cdot v$  for every  $\alpha \in \mathbb{C}$  and  $u, v \in \mathbb{V}$
10. (Distributive Property 2):  $(\alpha + \beta) \cdot u = \alpha \cdot u + \beta \cdot u$  for every  $\alpha, \beta \in \mathbb{C}$  and  $u \in \mathbb{V}$

**Definition 2.3.2** (Linear Combination). Let  $v_1, v_2, \dots, v_n \in \mathbb{V}$  and  $n \in \mathbb{N}$ . A linear combination of  $v_1, v_2, \dots, v_n$  is any vector of the formula

$$\alpha_1 v_1 + \alpha_2 v_2 + \dots + \alpha_n v_n \text{ with } \alpha_1, \alpha_2, \dots, \alpha_n \in \mathbb{C}$$

**Definition 2.3.3** (Linear Independence). Let  $\{v_1, \dots, v_n\} \in \mathbb{V}$ . The vectors,  $v_1, \dots, v_n$  are linearly dependent if there exist  $\alpha_1, \dots, \alpha_n \in \mathbb{C}$  not all zero such that

$$(2.3.1) \quad \alpha_1 v_1 + \alpha_2 v_2 + \dots + \alpha_n v_n = 0$$

otherwise the vectors are said to be linearly independent. Here, equation 2.3.1 holds only when  $\alpha_i = 0$  for all  $i = 1, 2, \dots, n$ .

**Definition 2.3.4** (Basis). A subset  $U \subset \mathbb{V}$  is said to be basis for  $\mathbb{V}$  if  $U$  is linearly independent and the span  $U = \mathbb{V}$

**Definition 2.3.5** (Standard Basis). The standard basis or Euclidean Basis for  $C^n$  is  $E = \{e_1, e_2, \dots, e_n\}$ , where  $e_j$  is the vector with a 1 in the  $j^{th}$  spot and 0's elsewhere,



i.e

$$e_1 = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, e_2 = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}, e_n = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$$

Hence any  $\alpha_j \in \mathbb{C}$  can be written as:

$$\begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix} = \alpha_1 \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \alpha_2 \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix} + \cdots + \alpha_n \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} = \sum_{j=1}^n \alpha_j e_j$$

**Definition 2.3.6** (Invertible Matrix Theorem). Let  $A$  be an  $n \times n$  matrix. Then the following conditions are equivalent:

1.  $A$  is invertible
2.  $Ax = 0$  has only the trivial solution
3. The columns of  $A$  are linearly independent
4.  $Ax = b$  has a solution for  $x$  for any  $b \in \mathbb{C}^n$
5.  $\det(A) \neq 0$

### 2.3.2 Inner Product Spaces

**Definition 2.3.7** (Inner Product). Let  $\mathbb{V}$  be a vector space over  $\mathbb{C}$ . A (complex) inner product is the map:

$$\langle \cdot, \cdot \rangle: \mathbb{V} \times \mathbb{V} \rightarrow \mathbb{C}$$

such that

1. (Additivity):  $\langle u + v, w \rangle = \langle u, w \rangle + \langle v, w \rangle$  for every  $u, v, w \in \mathbb{V}$ .



2. (Scalar Homogeneity):  $\langle \alpha u, v \rangle = \alpha \langle u, v \rangle$  for all  $\alpha \in \mathbb{C}$  and  $u, v \in \mathbb{V}$
3. (Conjugate Symmetry):  $\langle u, v \rangle = \overline{\langle v, u \rangle}$  for all  $u, v \in \mathbb{V}$
4. (Positive Definiteness):  $\langle u, u \rangle \geq 0$  for all  $u \in \mathbb{V}$  and  $\langle u, u \rangle = 0$  if and only if  $u = 0$

**Definition 2.3.8.** A vector space equipped with a (complex) inner product is called a (complex) inner product space

**Definition 2.3.9.** Define the space of square summable sequence as follows

$$\ell^2 = \left\{ \{z_j\}_{j=1}^{\infty} : z_j \in \mathbb{C} \text{ for all } j \in \mathbb{N} \text{ and } \sum_{j=1}^{\infty} |z_j|^2 < \infty \right\}$$

Then for  $z = z_j, w = w_j \in \ell^2(\mathbb{N})$ , the inner product is defined as:

$$\langle w, z \rangle = \sum_{j=1}^{\infty} w_j \overline{z_j}$$

**Definition 2.3.10.** Define the space of complex valued square integrable functions on  $R$  as:

$$\mathbb{L}^2([-\pi, \pi]) = \left\{ f : [-\pi, \pi) \rightarrow \mathbb{C} : \int_{-\pi}^{\pi} |f(\theta)|^2 d\theta < \infty \right\}$$

Then for  $f, g \in \mathbb{L}^2([-\pi, \pi])$ , the inner product is defined as:

$$\langle f, g \rangle = \int_{-\pi}^{\pi} f(\theta) \overline{g(\theta)} d\theta$$

**Definition 2.3.11 (Normed Linear Spaces).** A vector space  $\mathbb{V}$  over  $\mathbb{C}$  is a normed linear space (pre-Banach) if to every  $u \in \mathbb{V}$  there corresponds a real scalar  $\|u\|$  such that the following holds:

1.  $\|u\| \geq 0$  and  $\|u\| = 0$  if and only if  $u = 0$
2.  $\|\alpha u\| = |\alpha| \|u\|$  for all  $\alpha \in \mathbb{C}$



3.  $\|u + v\| \leq \|u\| + \|v\|$  for all  $u, v \in \mathbb{V}$  (Schwartz Inequality)

**Definition 2.3.12.** Let  $\mathbb{V}$  be a complex inner product space. For  $v \in \mathbb{V}$ , define the norm (associated to the inner product,  $\langle \cdot, \cdot \rangle$ ),  $\|v\|$  as follows:

$$\|v\| = \sqrt{\langle v, v \rangle}, \text{ where } \langle v, v \rangle \geq 0 \text{ and } \langle v, v \rangle \in \mathbb{R} \text{ for every } v \in \mathbb{V}$$

**Theorem 2.3.1** (Cauchy Schwartz). For every  $u, v \in \mathbb{V}$  we have

$$|\langle u, v \rangle| \leq \|u\| \|v\|$$

**Theorem 2.3.2.** For every  $u, v \in \mathbb{V}$  we have:  $\|u + v\| \leq \|u\| + \|v\|$

**Definition 2.3.13** (Orthogonality). Two vectors  $u, v \in \mathbb{V}$  are said to be orthogonal if  $\langle u, v \rangle = 0$

**Definition 2.3.14.** Let  $B \subset \mathbb{V}$  then

1.  $B$  is an orthogonal set if  $\langle u, v \rangle = 0$  for every  $u, v \in B$  with  $u \neq v$ .
2.  $B$  is an orthonormal set if  $B$  is an orthogonal set and  $\langle v, v \rangle = 1$  for every  $v \in B$ .

**Lemma 2.3.1.** Let  $B \subset \mathbb{V}$  be an orthogonal set of vectors with  $0$  not in  $B$ . Then  $B$  is linearly independent set

*Proof.* Let  $u_1, \dots, u_k \in B$  and suppose that

$$\alpha_1 u_1 + \alpha_2 u_2 + \dots + \alpha_k u_k = 0 \text{ for } \alpha_1, \dots, \alpha_k \in \mathbb{C}$$

Next we take the inner product of both sides with  $u_j$  for  $j = 1, 2, \dots, k$ . Then from,

$$\langle \alpha_1 u_1 + \dots + \alpha_k u_k, u_j \rangle = \langle 0, u_j \rangle = 0$$



we see that the left side can be rewritten in the following way:

$$\begin{aligned} \langle \alpha_1 u_1 + \cdots + \alpha_k u_k, u_j \rangle &= \alpha_1 \langle u_1, u_j \rangle + \alpha_j \langle u_j, u_j \rangle + \cdots + \alpha_k \langle u_k, u_j \rangle \\ &= \alpha_j \langle u_j, u_j \rangle. \text{ By Orthogonality} \end{aligned}$$

Since 0 is not in  $B$  and  $u_j \in B$ ,  $\langle u_j, u_j \rangle = 0$ . Thus, combining, we get:  $\alpha_j = 0$ .

which is true for any  $j$

In other words,  $\alpha_1 u_1 + \cdots + \alpha_k u_k = 0$  only when  $\alpha_j = 0$  for every  $j$ . Thus, the set  $B$  must be linearly independent  $\square$

**Definition 2.3.15.** Let  $z \in \mathbb{C}^n$  with a function defined by the sequence,  $z(j) = z_j$  whose domain is the finite set given by  $\{0, 1, \dots, N-1\}$ . We denote this set of functions as  $\ell^2(\mathbb{Z}_N)$ . Here we define inner product as follows:

$$\langle w, z \rangle = \sum_{j=0}^{N-1} w_j \bar{z}_j \text{ for all } w, z \in \mathbb{C}^N = \ell^2(\mathbb{Z}^N)$$

and the associated norm as follows:

$$\|z\|^2 = \left\{ \sum_{k=0}^{N-1} |z(j)|^2 \right\}^{\frac{1}{2}}$$

Again we define periodicity of  $z$  as follows:

$$\text{For any } m \in \mathbb{Z}, z(j + mN) = z(j)$$

**Theorem 2.3.3.** Let  $\mathbb{V}$  be a complex inner product space with finite dimensional basis  $U = \{u_1, \dots, u_n\}$ . Then,

1.  $v = \sum_{j=1}^n \langle v, u_j \rangle u_j$  for every  $v \in \mathbb{V}$
2.  $\langle v, w \rangle = \sum_{j=1}^n \langle v, u_j \rangle \overline{\langle w, u_j \rangle}$  for every  $v, w \in \mathbb{V}$
3.  $\|v\|^2 = \sum_{j=1}^n |\langle v, u_j \rangle|^2$  for every  $v \in \mathbb{V}$



From the standard basis (definition 2.3.5) that  $E = e_0, e_1, \dots, e_{N-1}$  for  $\mathbb{C}^N = \ell^2(\mathbb{Z}_N)$ .

Which implies that

$$e_j(n) = \begin{cases} 1 & \text{if } n = j \\ 0 & \text{if } n \neq j \end{cases}$$

Hence for  $z \in \mathbb{C}^N = \ell^2(\mathbb{Z}_N)$

$$z = \begin{bmatrix} z_0 \\ z_1 \\ \vdots \\ z_{N-1} \end{bmatrix} = \sum_{k=0}^{N-1} z(k) e_k = z_0 \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + z_1 \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix} + \dots + z_{N-1} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$$

Our goal is to transform this and write it with respect to a basis that consists of pure frequencies. So we define  $E_0, E_1, \dots, E_{N-1} \in \mathbb{C}^N = \ell^2 \mathbb{Z}^N$  by

(2.3.2)

$$E_0(n) = \frac{1}{\sqrt{N}}, E_1(n) = \frac{1}{\sqrt{N}} e^{\frac{2\pi i n}{N}}, \dots, E_m(n) = \frac{1}{\sqrt{N}} e^{\frac{2\pi i m n}{N}}, \dots, E_{N-1}(n) = \frac{1}{\sqrt{N}} e^{\frac{2\pi i (N-1)n}{N}}$$

for  $m, n \in \{0, 1, \dots, N-1\}$

### 2.3.3 Fourier Basis and Transform

Let  $z = [z(0)z(1) \dots z(N-1)]^{tr} \in \mathbb{C}^N = \ell^2(\mathbb{Z}_N)$ . For  $m = 0, 1, \dots, N-1$ , define

$$\hat{z}(m) = \sum_{n=0}^{N-1} z(n) e^{-\frac{2\pi i m n}{N}} \text{ and } \hat{z} = [\hat{z}(0)\hat{z}(1) \dots \hat{z}(N-1)]^{tr}$$

The operator  $\wedge : C_N \rightarrow C_N$  is called the Discrete Fourier Transform (DFT)

**Notation 2.3.1.** 1.  $\hat{z}(m) = \sqrt{N} \langle z, E_m \rangle$ . Thus, the discrete Fourier transform (up to a constant of  $\sqrt{N}$ ) is just  $[z]_{E_0, E_1, \dots, E_{N-1}}$

2. The operation,  $\wedge$  (DFT) is a linear transformation. Thus for  $a, b \in \mathbb{C}$  and

$$z, w \in \mathbb{C}^N = \ell^2(\mathbb{Z}_N),$$

$$(az + bw)^\wedge = a\hat{z} + b\hat{w}$$



The application of the above notation is enshrined in the following.

**Theorem 2.3.4.** Let  $z = (z(0)z(1) \cdots z(N-1)), w = (w(0)w(1) \cdots w(N-1)) \in \mathbb{C}^N = \ell^2(\mathbb{Z}_N)$  then the following holds:

1.  $z(n) = \frac{1}{N} \sum_{m=0}^{N-1} \hat{z} e^{\frac{2\pi i m n}{N}}$  (Fourier inversion formula)
2.  $\langle z, w \rangle = \frac{1}{N} \sum_{m=0}^{N-1} \hat{z}(m) \overline{\hat{w}(m)} = \langle \hat{z}, \hat{w} \rangle$  (Parseval's relation)
3.  $\|z\|^2 = \frac{1}{N} \sum_{m=0}^{N-1} |\hat{z}(m)|^2 = \frac{1}{N} \|\hat{z}\|^2$  (Plancherel's formula)

The operator  $\wedge$  (DFT) preserves the norm (size and the inner product which corresponds to the angle between the vectors) and by Parseval's relation and Plancherel's formula, this is true up to a constant which is independent of the function.

**Example 2.3.1.** Let  $N = 4$  and  $z = [2i \ 3 \ 1 - i]$ . We can find  $\hat{z}$ , the Discrete Fourier Transform of  $z$ , by the following: First we write the basis as follows:

$$\{E_0, E_1, E_2, E_3\} = \left\{ \frac{1}{2} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \frac{1}{2} \begin{bmatrix} 1 \\ i \\ -1 \\ -1 \end{bmatrix}, \frac{1}{2} \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix}, \frac{1}{2} \begin{bmatrix} 1 \\ -i \\ -1 \\ i \end{bmatrix} \right\}$$

This gives us:

$$\hat{z}(0) = 2 \cdot 1 + i \cdot 1 + 3 \cdot 1 + (1 - i) \cdot 1 = 6$$

$$\hat{z}(1) = 2 \cdot 1 + i \cdot (-i) + 3 \cdot (-1) + (1 - i) \cdot (i) = i + 1$$

$$\hat{z}(2) = 2 \cdot 1 + i \cdot (-1) + 3 \cdot 1 + (1 - i) \cdot (-1) = 4$$

$$\hat{z}(3) = 2 \cdot 1 + i \cdot i + 3 \cdot (-1) + (1 - i) \cdot (-i) = -3 - i$$

Hence,  $\hat{z} = [6 \ (i+1) \ 4 \ (-3-i)]^T$  and  $\|z\|^2 = 36 + 2 + 16 + 10 = 64$



**Definition 2.3.16** (Fourier Basis and Inversion). . Define  $F_m \in \mathbb{C}^N = \ell^2(Z_N)$  by

$$F_m(n) = \frac{1}{N} e^{\frac{2\pi i m n}{N}} \text{ for } n = 0, 1, 2, \dots, N-1$$

Let  $F = F_0, F_1, \dots, F_{N-1}$  then  $F_m = \frac{1}{N} R_m$ .  $F$  is therefore an orthogonal basis since all  $E_m$ 's are orthogonal basis. Hence

$$z = \sum_{m=0}^{N-1} \hat{z}(m) F_m$$

which makes DFT a linear change of basis, i.e  $\hat{z} = [z]_F$

**Definition 2.3.17.** To take a look at the change of basis in matrix form we define

$$w_N^{mn} = e^{\frac{2\pi i m n}{N}} \text{ and } \hat{z} = \sum_{m=0}^{N-1} \hat{z}(m) w_N^{mn}$$

In matrix form  $\hat{z}$  is given as follows:

$$\hat{z} = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & w_N^1 & w_N^2 & \dots & w_N^{N-1} \\ 1 & w_N^2 & w_N^4 & \dots & w_N^{2(N-1)} \\ \dots & \dots & \dots & \dots & \dots \\ 1 & w_N^{(N-1)} & w_N^{2(N-1)} & \dots & w_N^{(N-1)^2} \end{bmatrix} z = W_N$$

That is  $W_N = [w_N^{mn}]$  hence  $\hat{z} = W_N z$

**Example 2.3.2.** Let

$$W_3 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -\frac{1}{2} - i\frac{\sqrt{3}}{2} & -\frac{1}{2} + i\frac{\sqrt{3}}{2} \\ 1 & -\frac{1}{2} + i\frac{\sqrt{3}}{2} & -\frac{1}{2} - i\frac{\sqrt{3}}{2} \end{bmatrix}$$

and let

$$z = \begin{bmatrix} 0 \\ i\frac{\sqrt{3}}{2} \\ 2 \end{bmatrix} \in \mathbb{C}$$



Hence  $\hat{z}$  will be given by:

$$\hat{z} = W_3 z = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -\frac{1}{2} - i\frac{\sqrt{3}}{2} & -\frac{1}{2} + i\frac{\sqrt{3}}{2} \\ 1 & -\frac{1}{2} + i\frac{\sqrt{3}}{2} & -\frac{1}{2} - i\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} 0 \\ i\frac{\sqrt{3}}{2} \\ 2 \end{bmatrix} = \begin{bmatrix} 2 + i\sqrt{3} \\ \frac{1}{2} + i\frac{\sqrt{3}}{2} \\ -\frac{5}{2} - i\frac{3\sqrt{3}}{2} \end{bmatrix}$$

This shows that the operator  $\Lambda$  is invertible, i.e if  $\hat{z} = \hat{w}$ , then  $z = w$ .

**Definition 2.3.18.** Let  $w = (w(0), w(1), \dots, w(N-1)) \in \mathbb{C}^N = \ell^2(\mathbb{Z}_N)$  and define

$$\check{w}(n) = \frac{1}{N} \sum_{m=0}^{N-1} w(m) e^{\frac{2\pi i m n}{N}}, \text{ for } n = 0, 1, \dots, N-1 \text{ and } \check{w} = (\check{w}(0), \check{w}(1), \dots, \check{w}(N-1))$$

The map  $\vee : \ell^2(\mathbb{Z}_N) \rightarrow \ell^2(\mathbb{Z}_N)$  is called the inverse discrete Fourier transform (IDFT).

That is

$$(\hat{z})^\vee(n) = \sum_{m=0}^{N-1} \hat{z}(m) e^{\frac{2\pi i m n}{N}} = z(n) \text{ and } (\check{w})^\wedge = w$$

**Definition 2.3.19.** Define the space of functions as follows

$$(2.3.3) \quad L^2(\mathbb{R}) = \left\{ f : \mathbb{R} \rightarrow \mathbb{C} : \int_{\mathbb{R}} |f(x)|^2 dx < \infty \right\}$$

$$(2.3.4) \quad \text{with an inner product, } \langle f, g \rangle = \int_{\mathbb{R}} f(x) \overline{g(x)} dx \text{ and the norm as}$$

$$(2.3.5) \quad \|f\| = \left( \int_{\mathbb{R}} |f(x)|^2 dx \right)^{\frac{1}{2}}$$

Then for  $f \in L^2(\mathbb{R})$  and  $\xi \in \mathbb{R}$ , define  $\hat{f}$ , the Fourier transform of  $f$ , to be the function

$$(2.3.6) \quad \hat{f}(\xi) = \int_{\mathbb{R}} f(x) e^{-ix\xi} dx$$

And for  $f \in L^2([-pi, pi])$  and  $\xi \in \mathbb{R}$ , define  $\check{f}$ , the inverse Fourier transform of  $f$  to be the function in  $(L^2([-pi, pi]))$  as follows:

$$(2.3.7) \quad \check{f}(\xi) = \frac{1}{2\pi} \int_{\mathbb{R}} f(\xi) e^{-ix\xi} d\xi \text{ and } (\check{f})^\wedge = (\hat{f})^\vee = f$$



**Theorem 2.3.5** (Properties of the Fourier Transform). For  $f, g \in L^2(\mathbb{R})$ , we have:

1. If  $f * g(x) = \int_{\mathbb{R}} f(x-y)g(y)dy$  then  $(f * g)^\wedge = \hat{f}\hat{g}$

2. If  $R_y f(x) = f(x-y)$ , then  $(R_y f)^\wedge(\xi) = e^{-iy\xi} \hat{f}(\xi)$

3. If  $f_t(x) = f(\frac{x}{t})$ , then  $f_t^\wedge(\xi) = t\hat{f}(t\xi)$

4. If  $f(x) \rightarrow \infty$ , then  $(f')^\wedge(\xi) = (i\xi)\hat{f}(\xi)$

5. If  $xf(x) \in L^2(\mathbb{R})$ , then  $\frac{d}{d\xi}\hat{f}(\xi) = ((ix)f(x))^\wedge(\xi)$

6. If  $f \in L^1(\mathbb{R}) \cup L^2(\mathbb{R})$ , then as  $\xi \rightarrow \infty$ ,  $\hat{f}(\xi) \rightarrow 0$

#### 2.3.4 Wavelet Basis and Transforms

**Definition 2.3.20.**  $\mathbb{Z} \in \mathbb{C}^N = \ell^2(\mathbb{Z}_N)$  is localized in space near  $n_0$  if most of the components  $z(n)$  of  $\mathbb{Z}$  are 0 or at least relatively small except for a few values of  $n$  close to  $n_0$ .

The Euclidean or standard basis is well localized in space but poorly frequency localized. Although the Fourier (Discrete) transform could be frequency localized, it is also not well localized in space. We proceed by discussing wavelets which will help us find basis which are well localized both in space and frequency

**Definition 2.3.21** (Translate). For  $x \in \ell^2(\mathbb{Z}_N) = \mathbb{C}^N$  and  $k \in \mathbb{Z}$  define the translate of  $z$  by the follows:

$$R_k z(n) = z(n-k) \text{ for } k \in \mathbb{Z}$$

**Definition 2.3.22** (Convolution). For  $z, w \in \ell^2(\mathbb{Z}_N) = \mathbb{C}^N$  and  $k \in \mathbb{Z}$  define convolution  $z * w \in \ell^2(\mathbb{Z}_N) = \mathbb{C}^N$  as follows:

$$z * w(m) = \sum_{n=0}^{N-1} z(m-n)w(n) \text{ for } m = 0, 1, \dots, N-1$$



**Definition 2.3.23 (Subspace).** Let  $\mathbb{V}$  be a vector space. A subset  $W \subset \mathbb{V}$  is a subspace of  $\mathbb{V}$  if  $W$  is itself a vector space with the same operation (i.e vector addition  $+$  and scalar multiplication  $\cdot$ ) as  $\mathbb{V}$

**Definition 2.3.24 (The Orthogonal Complement).** Let  $\mathbb{V}$  be an inner product space and  $W \subset \mathbb{V}$  a subspace of  $\mathbb{V}$ . Then, the orthogonal complement of  $W$  is given by:

$$W^\perp = \{v \in \mathbb{V} : \langle v, w \rangle = 0 \text{ for every } w \in W\}$$

$W^\perp$  consist of the vectors in  $\mathbb{V}$  that are orthogonal to every element in  $W$

**Definition 2.3.25.** Let  $\mathbb{V}$  be an inner product space and  $W \subset \mathbb{V}$  a subspace of  $\mathbb{V}$ . Suppose that  $B = \{w_1, w_2, \dots, w_k\}$  is an orthogonal basis for  $W$  and define the projection of  $v$  onto  $W$ , by  $v_B$ , by

$$\sum_{i=1}^k \frac{\langle v, w_i \rangle}{\langle w_i, w_i \rangle} w_i \in W \text{ then } v - v_B \in W^\perp$$

That shows that  $\langle v - v_B, w \rangle = 0$  for every  $w \in W$ . Since  $B$  is a basis for  $W$ , we can write any  $w \in W$  in the form

$$\sum_{j=1}^k \beta_j w_j = \beta_1 w_1 + \beta_2 w_2 + \dots + \beta_k w_k \text{ for some } \beta_1, \beta_2, \dots, \beta_k \in \mathbb{C}$$



Hence

$$\begin{aligned}
\langle v - v_B, w \rangle &= \langle v, w \rangle - \langle v_B, w \rangle \\
&= \langle v, w \rangle - \left\langle v_B, \sum_{j=1}^k \beta_j w_j \right\rangle \\
&= \langle v, w \rangle - \sum_{j=1}^k \overline{\beta_j} \langle v_B, w_j \rangle \\
&= \langle v, w \rangle - \sum_{j=1}^k \overline{\beta_j} \left\langle \sum_{i=1}^k \frac{\langle v, w_i \rangle}{\langle w_i, w_i \rangle} w_i, w_j \right\rangle \\
&= \langle v, w \rangle - \sum_{j=1}^k \overline{\beta_j} \sum_{i=1}^k \frac{\langle v, w_i \rangle}{\langle w_i, w_i \rangle} \langle w_i, w_j \rangle \\
&= \langle v, w \rangle - \sum_{j=1}^k \overline{\beta_j} \frac{\langle v, w_j \rangle}{\langle w_j, w_j \rangle} \langle w_j, w_j \rangle \\
&= \langle v, w \rangle - \sum_{j=1}^k \overline{\beta_j} \langle v, w_j \rangle \\
&= \langle v, w \rangle - \left\langle v, \sum_{j=1}^k \beta_j w_j \right\rangle \\
&= \langle v, w \rangle - \langle v, w \rangle = 0
\end{aligned}$$

The definition above proves the orthogonal decomposition theorem given below where  $w = v_B$  and  $w^\perp = v - v_B$

**Definition 2.3.26** (Orthogonal Decomposition Theorem). Let  $\mathbb{V}$  be an inner product space and let  $W$  be a finite dimensional subspace of  $\mathbb{V}$ . Then, any  $v \in \mathbb{V}$  can be written as  $v = w + w^\perp$  with  $w \in W$  and  $w^\perp \in W^\perp$

**Definition 2.3.27** (First-Stage Wavelet Basis for  $\mathbb{Z}_N$ ). To find a basis which consists of translated and scaled versions of a single function we assume that  $N$  is a power of 2, i.e  $N = 2^n$  for some  $n \in \mathbb{N}$ . Then by the orthogonal decomposition theorem;

$$\mathbb{C}^N \approx \mathbb{V}^n = \mathbb{V}^{n-1} \oplus W^{n-1}$$



If we find two functions,  $u, v \in \ell^2(\mathbb{Z}_N) = \mathbb{C}^N \approx \mathbb{V}^n$  so that  $\{u(x - 2k)\}_{k=0}^{2^{n-1}-1} = \{(R_{2k}u)(x)\}_{k=0}^{2^{n-1}-1}$  is an orthogonal basis for  $\mathbb{V}^{n-1}$  and  $\{v(x - 2k)\}_{k=0}^{2^{n-1}-1} = \{(R_{2k}v)(x)\}_{k=0}^{2^{n-1}-1}$  is an orthogonal basis for  $\mathbb{W}^{n-1}$  then the result of the union,  $\{(R_{2k}u)(x)\}_{k=0}^{2^{n-1}-1} \cup \{(R_{2k}v)(x)\}_{k=0}^{2^{n-1}-1}$  for  $\mathbb{V}^n \approx \mathbb{C}^N$  is called the first wavelet basis with  $u$  and  $v$  being the mother and father wavelets respectively.

**Theorem 2.3.6.** Suppose  $M \in N$  and  $N = 2M$ . Let  $u, v \in \ell^2(\mathbb{Z}_N) = \mathbb{C}^N$ . then

$$B = \{(R_{2k}u)\}_{k=0}^{M-1} \cup \{(R_{2k}v)\}_{k=0}^{M-1}$$

is an orthogonal basis for  $\ell^2(\mathbb{Z}_N) = \mathbb{C}^N$  if and only if

1.  $|\hat{u}(n)|^2 + |\hat{u}(n + M)|^2 = 2$
2.  $|\hat{v}(n)|^2 + |\hat{v}(n + M)|^2 = 2$
3.  $\hat{u}(n)\overline{\hat{v}(n)} + \hat{u}(n + M)\overline{\hat{v}(n + M)} = 0$

for all  $n = 0, 1, \dots, M - 1$

**Definition 2.3.28 (Mother Wavelet).** A function  $\psi \in L^2(\mathbb{R})$  is a mother wavelet if the following holds:

$$\|\psi\| = 1 \text{ and } \pi \int_{\mathbb{R}} \frac{|\hat{\psi}(\xi)|^2}{\xi} d\xi < \infty$$

### 2.3.5 Discrete Wavelet and Multiresolution Analysis

**Definition 2.3.29.** For  $\varphi, \psi \in L^2(\mathbb{R})$  and  $j, k \in \mathbb{Z}$ , define  $\varphi_{j,k}, \psi_{j,k} \in L^2(\mathbb{R})$  by:

$$\varphi_{j,k}(x) = 2^{\frac{j}{2}} \varphi(2^j x - k) \text{ and } \psi_{j,k}(x) = 2^{\frac{j}{2}} \psi(2^j x - k) \text{ where } \|\psi_{j,k}\| = \|\psi\|$$

**Definition 2.3.30 (Wavelet System).** A system of wavelet is a complete orthogonal set in  $L^2(\mathbb{R})$  of the form  $\{\psi_{j,k}\}_{j,k \in \mathbb{Z}}$  for some  $\psi \in L^2(\mathbb{R})$ . That is the set for which  $\langle f, \psi_{j,k} \rangle = 0$  for all  $j, k \in \mathbb{Z}$  only if  $f = 0$ . We note that the function  $\psi_{j,k}$  are called wavelets and the  $\psi$  is the mother wavelet



**Definition 2.3.31.** If  $\{\psi_{j,k}\}_{j,k \in \mathbb{Z}}$  is a wavelet system, then any  $f \in L^2(\mathbb{R})$  can be expanded in terms of the  $\psi_{j,k}$  and this gives the identity below:

$$f = \sum_{j \in \mathbb{Z}} \sum_{k \in \mathbb{Z}} \langle f, \psi_{j,k} \rangle \psi_{j,k}$$

The function  $Tf(j, k) = \langle f, \psi_{j,k} \rangle$  is called the discrete wavelet transform (DWT) where  $Tf(j, k) = Wf(2^{-j}, k)$

**Definition 2.3.32** (Multiresolution Analysis). The multiresolution analysis (MRA) with scaling function  $\varphi$  is sequence  $\{V_j\}_{j \in \mathbb{Z}}$  of the subspace of  $L^2(\mathbb{R})$  with the following properties:

1.  $V_j \subset V_{j+1}$  for all  $j \in \mathbb{Z}$  (Nesting or Monotonicity)
2. There exist a function  $\varphi \in V_0$  such that  $\{\varphi_{0,k}\}_{k \in \mathbb{Z}}$  is orthogonal and

$$V_0 = \left\{ \sum_{k \in \mathbb{Z}} z(k) \varphi_{0,k} \mid z = \{z(k)\} \in \ell^2(\mathbb{Z}) \right\} \text{ Existence of Scaling function}$$

Which implies that  $\{\varphi_{0,k}\}$  is an orthogonal basis for  $V_0$

3. For any  $j \in \mathbb{Z}$ ,  $f(x) \in V_0$  if and only if  $f(x) \in V_j$ , (Dilation or Scaling)
4.  $\cup_{j \in \mathbb{Z}} V_j = \{0\}$  (Separation)
5.  $\cup_{j \in \mathbb{Z}} V_j$  is dense in  $L^2(\mathbb{R})$  (Density)

**Lemma 2.3.2.** Let  $\{V_j\}_{j \in \mathbb{Z}}$  be a MRA with scaling function  $\varphi$  and scaling sequence  $u$ . Then  $\{R_{2k}u\}_{k \in \mathbb{Z}}$  is an orthogonal set in  $\ell_2(\mathbb{Z})$

**Proposition 2.3.1.** Suppose  $u = (u(j))_{j \in \mathbb{Z}} \in \ell_2(\mathbb{Z})$  is such that  $\{R_{2k}u\}_{k \in \mathbb{Z}}$  is orthogonal in  $\ell_2(\mathbb{Z})$ . Define  $v = (v(j))_{j \in \mathbb{Z}} \in \ell_2(\mathbb{Z})$  by

$$v(j) = (-1)^{j-1} \overline{u(1-j)}$$

Then  $\{R_{2k}u\}_{k \in \mathbb{Z}}$  is an orthogonal



**Lemma 2.3.3.** Let  $\{V_j\}_{j \in \mathbb{Z}}$  be a MRA with scaling function  $\varphi$  and scaling sequence  $u = (u(k))_{k \in \mathbb{Z}}$ . Define  $v = (v(k))_{k \in \mathbb{Z}}$  by:

$$v(k) = (-1)^{k-1} \overline{u(1-k)} \text{ for all } k \in \mathbb{Z}, \text{ and define } \psi \text{ by}$$

$$\psi(x) = \sum_{k \in \mathbb{Z}} v(k) \varphi_{1,k}(x) = \sum_{k \in \mathbb{Z}} v(k) \sqrt{2} \varphi(2x - k)$$

Then  $\{\psi_{0,k}\}$  is an orthogonal set in  $L^2(\mathbb{R})$

$$\text{Define } W_0 = \left\{ \sum_{k \in \mathbb{Z}} z(k) \psi_{0,k} \mid z = (z(k))_{k \in \mathbb{Z}} \in \ell^2(\mathbb{Z}) \right\} \text{ then, } V_1 = V_0 \oplus W_0$$

**Theorem 2.3.7** (Mallet's Theorem). Let  $\{V_j\}_{j \in \mathbb{Z}}$  be a MRA with scaling function  $\varphi$  and scaling sequence  $u = (u(k))_{k \in \mathbb{Z}}$ . Define  $v = (v(k))_{k \in \mathbb{Z}}$  by

$$v(k) = (-1)^{k-1} \overline{u(1-k)} \text{ and } \psi(x) = \sum_{k \in \mathbb{Z}} v(k) \psi_{1,k}(x)$$

Then  $\{\psi_{j,k}\}_{k \in \mathbb{Z}}$  is a wavelet system in  $L^2(\mathbb{R})$

**Example 2.3.3.** In this example we demonstrate how to construct wavelet using the Haar MRA. Given Haar MRA as

$$V_j = \{f \in L^2(\mathbb{R}) \mid f \text{ is constant on } [2^j k, 2^j(k+1)) \text{ for all } k \in \mathbb{Z}\} \text{ and}$$

$$\begin{cases} 1 & \text{if } x \in [0, 1) \\ 0 & \text{otherwise} \end{cases}$$

The key is to find the scaling sequence  $u = (u(k))_{k \in \mathbb{Z}}$ . For example,

$$\varphi_{1,k}(x) = \sqrt{2} \varphi(2x - k) = \begin{cases} \sqrt{2} & \text{if } x \in [\frac{k}{2}, \frac{k+1}{2}) \\ 0 & \text{otherwise} \end{cases}$$

Then we have

$$\begin{aligned} u(k) &= \langle \varphi, \varphi_{1,k} \rangle \\ &= \int_{\mathbb{R}} \varphi(x) \overline{\varphi_{1,k}(x)} dx \\ &= \int_{[0,1) \cap [\frac{k}{2}, \frac{k+1}{2})} \sqrt{2} dx = \begin{cases} \frac{\sqrt{2}}{2} & \text{if } k = 0, 1 \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$



Hence the scaling equation becomes;  $\varphi(x) = \varphi(2x) + \varphi(2x - 1)$ . Define  $v$  according to Lemma (2.3.3),  $v(j) = (-1)^{j-u} \overline{u(i-j)}$ , we get

$$v(0) = -\frac{1}{\sqrt{2}}; v(1) = \frac{1}{\sqrt{2}}; v(k) = 0 \text{ for } k \neq 0, 1$$

Next, we define  $\psi$  using the same definition in Lemma (2.3.3) as follows:

$$\psi(x) = \sum_{k \in \mathbb{Z}} v(k) \sqrt{2} \varphi(2x - k)$$

which gives us

$$\psi(x) = -\psi(2x) + \psi(2x - 1) = \begin{cases} -1 & \text{if } x \in [0, \frac{1}{2}) \\ 1 & \text{if } x \in [\frac{1}{2}, 1) \\ 0 & \text{otherwise} \end{cases}$$

And hence evaluating  $\psi_{j,k}$ , we get a wavelet system,  $\{\psi\}_{j,k} \in \mathbb{Z}$  as follows

$$\psi_{j,k}(x) = \begin{cases} -2^{\frac{j}{2}} & \text{if } x \in [\frac{k}{2^j}, \frac{k}{2^j} + \frac{1}{2^{j+1}}) \\ 2^{\frac{j}{2}} & \text{if } x \in [\frac{k}{2^j} + \frac{1}{2^{j+1}}, \frac{k}{2^j}) \\ 0 & \text{otherwise} \end{cases}$$



## CHAPTER III

# METHODOLOGY

# KNUST

### 3.1 Introduction

The extraction and analysis of fingerprint features is not new in the area of biomedical engineering however research in this area tries to understand how features of a fingerprint can be extracted fully and used to identify person's uniquely with no commission of error.

All human being posses fingerprint and these fingerprints are result of unique ridge and valley structure formed by skin over the fingers (Woodward et al., 2003). Ridges and valleys often run in parallel and these structures have bifurcation and ridge endings called termination. This ridge structure as a whole takes different shapes characterized by high curvature, crossover or bufucation and are called singularity.

These singularities may be categorize into three main topologies thus, the loop, delta and the whorl but what makes fingerprint distinct is the dispersion of such structures at local level and these are called minutiae (Kekre and Bhatnagar, 2007). Minutiae simply means small details and this refers to the various ways that the ridges can be irregular. An unexpected ridge end is termed termination while when it divides into two is called bifurcations.

By this insight, computer vision methods related to fingerprint analysis are en-



gaged to detect, track and extract these features automatically to recognize peoples identity. But these features can be efficiently extracted when the appropriate fundamental mathematical tools are engaged. The following discusses the appropriate tools that was considered in order to efficiently extract features for recognition.

## 3.2 Image Processing

### 3.2.1 Introduction

Let  $f$  be a real-valued function defined on the real line  $\mathbb{R}$  and square integrable as follows:

$$(3.2.1) \quad \int_{-\infty}^{\infty} f^2(t) dt < \infty$$

If  $f(t)$  is considered as a value of a signal at some time  $t$ , then the signal  $f$  can be analyzed in ways other than the time-value form, i.e  $t \rightarrow f(t)$ . Signals are analyzed in terms of frequency components and various combinations of time and frequency components. These component parts of the signal are at times altered to remove unwanted features or compressed to get more efficient storage however it is essential for this signal to be reconstructed from its component parts and this constitute the Analysis, Processing and Synthesis in Image Processing.

Analysis of signal in to other words mean to decompose a signal into it basic components. Consider a signal space as a vector space then this signal can be broken up into series of sums of subspace where each subspace captures a special feature of the signal. Processing of signal also involves the modification of some basic components of the signal that are obtained from the analysis. Reconstruction of a signal from its component parts after been altered is what we call synthesis. All these helps to understand signals when it comes to comparing the altered signals with respect to the original signal in test for convergence.



### 3.2.2 Image Formation

An image may be viewed as being derived from a continuous image function,  $f : \mathbb{R}^+ \cup \{0\}$  by taking a finite number of samples where the functional value  $f(x)$  represents the intensity and some desired parameters of the physical image at the point  $x \in \mathbb{R}^2$ . Digital image may be defined as a matrix (a two-dimensional array) of pixels Nixon and Aguado (2002) where the value of each pixel is proportional to the brightness of the corresponding point and its value is often derived from the output of an Analog/Digital (A/D) converter.

A matrix of pixels, which is the image can be describe as an  $M \times N$ ,  $p$ -bit pixels where  $N$  and  $M$  are the number of points along the horizontal and vertical axis respectively while  $p$  is the brightness value and gives a range of  $2^p$  values, ranging from 0 to  $2^p - 1$  value of 8 giving the brightness levels between 0 and 255 which is usually displayed as black and white respectively with a shades of gray in between them. Smaller values of  $p$  gives fewer levels which reduce the contrast of the image

Color images follow a similar strategy in specifying it pixels intensity but instead of one image plane, it uses  $q$  intensity components where  $q$  is the number of colors corresponding to the image plane. Example, the  $(R - G - B)$  image space has  $q = 3$  corresponding to red, green and blue colors components.

In any colors mode, the pixel's color can be specified in two ways. First, you can associate an integer value, with each pixel, that can be used as an index to a table that stores the intensity of each color component. The index is used to recover the actual color from the table when the pixel is going to be displayed or processed. In this method, the table is know as the image palette and the display is performed by color mapping.

The alternative is to use several image planes to store the color components of



each pixel. This method is known as true color and it represents an image more accurately, essentially by considering more color. The most common format uses 8 bits for each of the three  $R - G - B$  components. These images are known as 24-bit true color and they can contain  $2^{24} = 16777216$  different colors simultaneously.

### 3.2.3 Image Processing Techniques

#### Image Arithmetic

In image arithmetic, one of the standard arithmetic operations or logical operations is applied to two or more images. The operators are applied in a pixel-by-pixel fashion, which means that the value of a pixel in the output image depends only on the values of the corresponding pixels in the input image. Hence, the images normally have to be of the same size. Image arithmetic is very important because of its application and beside are very simple and fast.

#### Pixel Addition

The Pixel Addition operator or offset take as input, two identical sized images and produces as output a third image of the same size as the first two. Each pixel value is the sum of the values of the corresponding pixel from each of the two input images. This property can however be expanded to allow more than two images to be combined by using various additive law Davies (1990). The addition of two images is performed in a single pass. The output pixel values are given by:

$$(3.2.2) \quad I(i, j) = P_1(i, j) + P_2(i, j)$$

A constant value  $K$  can also be added to a single image by using the following:

$$(3.2.3) \quad I(i, j) = P_1(i, j) + K$$



If the pixel values in the input images are actually vectors rather than scalar values thus for example a color image then the individual components red, blue and green are simply added separately to produce the output value. In situations where the pixel addition is greater than the maximum allowed pixel value, the overflowing pixel values could be set to the maximum allowed value which then result in saturation effect.

### Pixel Subtraction

The pixel subtraction or difference takes two images as input and produces as output a third image whose pixel values are those of the first image minus the corresponding pixel values from the second image. It is also often possible to just use a single image as input and subtract a constant value from all the pixels. Some versions, Marion (1991); Davies (1990) of the operator will output the absolute difference between pixel values rather than the signed output. The output pixel values are given by:

$$(3.2.4) \quad I(i, j) = P_1(i, j) - P_2(i, j)$$

If the operator computes absolute difference between the two input image then:

$$(3.2.5) \quad I(i, j) = |P_1(i, j) - P_2(i, j)|$$

or if desired to subtract a constant value  $K$  from a single image then:

$$(3.2.6) \quad I(i, j) = P_1(i, j) - K$$

If the operator calculates absolute differences and the two input images use the same pixel value type, then it is impossible for the output pixel values to be outside the range that may be represented by the input pixel type and so this problem does not arise and hence the advantage of using absolute differences.



## Pixel Multiplication and Scaling

Pixel multiplication usually comes in two main forms thus the first one takes two input images and produces an output image in which the pixel values are just those of the first image multiply by the values of the corresponding values in the second image while the second one is termed the scaling method where each pixel value is multiplied by a specified constant.

The multiplication of two images is performed in a single pass using the formula:

$$(3.2.7) \quad I(i, j) = P_1(i, j) \times P_2(i, j)$$

Scaling by a constant is performed using:

$$(3.2.8) \quad I(i, j) = P_1(i, j) \times K$$

The constant  $K$  is often a floating point number and may be less than one which will reduce the image intensities. It may even be negative at times if the image format supports that. If the pixel value is actually vectors rather than scalar value thus color image then the individual components (i.e Red, Blue and Green components) are multiplied separately to produce the output value. Again if the output values are calculated to be larger than the maximum allowed pixel value then they may either be truncated at that maximum value or wrap-around and continue upwards from the minimum allowed number again (Boyle and Thomas, 1988). A scaling factor greater than one will brighten an image and when less than one will darken the image. Scaling generally produces a much more natural brightening/darkening effect rather than simply adding an offset to the pixels (Boyle and Thomas, 1988), since it preserves the relative contrast of the image better. Scaling is also useful prior to other image arithmetic in order to prevent pixel values going out of the range.



Using pixel-by-pixel multiplication, a binary image can be used to multiply another image in order to act as a mask which is achieved by multiplying 1 (one) to pixels that are to be preserved 0 (zero) to those that are not.

### Pixel Division

The pixel division operator takes two images as input and produces a third pixel values that are pixel values of the first divided by the corresponding pixel values of the second image. This can also be implemented by dividing with a specified constant the single input image. The division of two images,  $P_1$  and  $P_2$  is given by the formula:

$$(3.2.9) \quad I(i, j) = P_1(i, j) \div P_2(i, j)$$

Division by a constant  $K$  is also given by:

$$(3.2.10) \quad I(i, j) = P_1(i, j) \div K$$

The division operator may only handle integer or floating-point division. If integer division is performed, then the resulting image is rounded down to the next lowest integer for output. Unlike in pixel subtraction which gives absolute change for each pixel, division gives the fractional change between the corresponding pixel value

#### 3.2.4 Point Processing Operators

Single point processing is an image enhancing technique which determines a pixel value in the enhanced image depending only on the value of the corresponding pixel in the image. The process can be described with the mapping function as follows (Marion, 1991):

$$(3.2.11) \quad S = M(R)$$

Where  $R$  and  $S$  are the point values in the input and output images respectively. The form of the mapping function,  $M$  determines the effect of the operation. It can



also be define in an ad-hoc manner using the thresholding or gamma correction or can be computed from the input image using histogram equalization[section 3.2.6].

Point operators are also know as Look-Up Table (LUT) Transformations, since the mapping function in the case of a discrete image can be implemented in a look-up table. A subgroup of the point processors is the set of anamorphosis operators. The describe all point operators with a strictly increasing or decreasing mapping function. Examples of such include the logarithm operator, exponential operators and many among others.

### 3.2.5 Thresholding

Thresholding is used to separate out the regions of the image corresponding to the objects of interest from the regions of the image that correspond to th background. Thresholding, (Marion, 1991; Nixon and Aguado, 2002) often provides an easy and convenient way to perform this segmentation based on the different intensities or colors in the foreground and background regions of an image. Thresholding also enables one to see what areas of an image consist of pixels whose values lie within a specified range of intensities.

The input to a thresholding operation is typically a gray-scale or color image. In its implementation the output is a binary representing the segmentation. Black pixels correspond to background and white pixels correspond to foreground. This segmentation is determined by a single parameter known as the intensity threshold. In a single pass, each pixel in the image is compared with this threshold. If the pixel's intensity is higher than the threshold, the pixel is set to white in the output or otherwise it is set to black. In other implementations, multiple thresholds can be specified so that an intensity value can be set to white whiles everything else is set



to black. For example in color image or multi-spectral image, it may be possible to set different thresholds for each color channel and that makes it possible to select just those pixels within a specified RGB space.

Not all images can be clearly segmented into foreground and background using simple thresholding and as to whether an image can be correctly segmented this way can be determined using the intensity histogram of the image. If it is possible to separate out the foreground of an image on the basis of pixel intensity, then the pixels within the foreground objects must be distinctly different from the intensity of pixels within the background. In this case distinct peak value is expected in the histogram corresponding to the foreground such that the thresholds can be chosen to isolate this peak accordingly. If such a peak does not exist, then it is unlikely that simple thresholding will produce a good segmentation. In this case adaptive thresholding may be a better option to be considered.

### 3.2.6 Histogram Equalization

Histogram equalization techniques provides a method for modifying the contrast of an image by altering that image such that its intensity has a desired shape. Histogram equalization employs a monotonic, non-linear mapping which re-assigns the intensity values of pixels in the image such that the output image contains a uniform distribution of intensities.

This technique is usually introduced by using continuous rather than discrete process function (Semmler, 2004). Therefore, supposing that the images of interest contain continuous intensity level on the interval  $[0, 1]$  and that the transformation function  $f$  which maps an input image  $A(x, y)$  onto an output image  $B(x, y)$  is continuous within this interval. Assuming that the transfer law  $D_B = f(D_A)$  where  $D$



is the intensity level, single-valued and monotonic increasing then the corresponding inverse law from Boyle and Thomas (1988) can be defined as:

$$D_A = f^{-1}(D_B)$$

All pixels in the input image with densities in the region  $D_A$  to  $D_A + dD_A$  will have their pixel values re-assigned such that they assume an output pixel density value in the range from  $D_B$  to  $D_B + dD_B$ . The surface areas  $h_A(D_A)dD_A$  and  $h_B(D_B)dD_B$  will therefore be equal giving us:

$$(3.2.12) \quad h_B(D_B) = h_A(D_A) \div d(D_A)$$

where

$$d(D_A) = \frac{df(D_A)}{dD_A}$$

This result in the language of probability theory can be written if the histogram  $h$  is regarded as a continuous probability density function  $P$  describing the distribution of the intensity levels as follow (Marion, 1991; Davies, 1990):

$$(3.2.13) \quad P_B(D_B) = P_A(D_A) \div d(D_A)$$

In the case of histogram equalization thus from section 3.2.6, the output probability densities should all be an equal fraction of the maximum number of intensity levels in the input image  $D_M$  where the maximum level considered is 0 (zero). The transfer function necessary to archive this result is:

$$(3.2.14) \quad h_B(D_A) = D_M \times P_A(D_A)$$

therefore,

$$(3.2.15) \quad F_A(D_A) = D_M \int_0^{D_A} P_A(D_A)$$



where  $F_A(D_A)$  is the cumulative probability distribution or the cumulative histogram of the original image.

Thus, an image which is transformed using its cumulative histogram produces an output histogram which is flat. A digital implementation of histogram equalization is usually performed by defining a transfer function of the form:

$$(3.2.16) \quad F_A(D_A) = \max \left( 0, \text{round} \left[ D_M \times \frac{n_k}{N^2} \right] - 1 \right)$$

where  $N$  is the number of image pixels and  $n_k$  is the number of pixels at intensity level  $k$  or less. In Marion (1991)'s implementation, the output image will not be necessarily be fully equalized and there may be 'holes' in the histogram which are unused intensity levels. These effects are likely to decrease as the number of pixels and intensity levels in the output image increase and this is effective for detailed enhancement of images and in the correction of non-linear effects introduced by digitizers and the likes.

### 3.2.7 Intensity Histogram

In the context of image processing, histogram of an image is normally referred to a histogram of the pixel intensity values. This histogram is a graph showing the number of pixels in an image at each different intensity value found in that image.

For an 8-bit gray-scale image, there are 256 different possible intensities and so the histogram will graphically display 256 numbers showing the distribution of pixels amongst those gray-scale values.

Histogram can also be taken of color images, individual histogram or red, green and blue channels can be taken or a 3-D histogram can be produced with three axis representing the channels and brightness at each point representing the pixel count, however the exact output from the operation depends heavily on its implementation.



One common use of histogram is to decide what value of threshold to use when converting a gray-scale image to a binary one by thresholding. If the image is suitable for thresholding, then the histogram will be bimodal thus the pixel intensities will be clustered around two well distinct values. A suitable threshold for separating these two groups will then be found somewhere in between the two peaks in the histogram otherwise it will be unlikely that a good segmentation can be produced by thresholding.

KNUST

### 3.2.8 Local Enhancement

The histogram processing techniques discussed are global in the sense that they apply a transformation function whose form is based on the intensity level distribution of an entire image. Although this method can enhance the overall contrast of an image thereby making certain details more visible but there are cases (Stollnitz et al., 2006; Oberst, 2007) where enhancement of details over small areas may be desired. The solution in these cases is to derive a transformation based on the intensity distribution in the local neighborhood of every pixel in the image.

The procedure involves defining a neighborhood around each pixel and using the histogram characteristics of this neighborhood to derive a transfer function which maps that pixel into an output intensity level. This is performed for each pixel in the image, this is because moving across rows and down columns only adds one new pixel to the local histogram hence updating the histogram from the previous calculation with new data introduced at each motion is possible. Local enhancement may also be defined on transforms based on pixel on pixel attributes other than histogram like intensity mean or intensity variance.



### 3.2.9 Image Analysis

Many categories of image processing operators transform an input image to form a new image without attempting to extract usable global information from the image. The following operators do extract globally usable information such as:

1. Pixel Value Distribution (the number of pixels having each value)
2. Classified Pixels (the category of scene entity that the corresponding pixel is taken from)
3. Connected Components (the groups of pixels all of which have the same label or classification)

It is possible to represent some of this information as an image thus an image whose pixels are the category or index number of the scene structure but the information need not always be so represented nor even necessarily be represented as an image. This category of operation is often considered part of the middle level image interpretation (i.e a signal to symbol transformation or feature extraction) and whose results might ultimately be used in higher level image interpretation (i.e symbol to symbol transformations such as scene description or object location)

#### Classification

Classification includes a broad range of decision theoretic approaches to the identification of images or part of images. All classification algorithms are based on the assumption, Boyle and Thomas (1988); Marion (1991); Davies (1990) that the image in question depicts one or more features and that each of these features belongs to one of the several distinct and exclusive classes. The classes may be specified a priori by an analyst (i.e supervised classification) or automatically clustered (i.e unsuper-



vised classification) into sets of prototype classes where the analyst merely specifies the number of the desired categories.

Image classification analyzes the numerical properties of various image features and organizes data into categories. Classification algorithms typically employ two phases of processing: training and testing. In the initial training phase, characteristics properties of typical image features are isolated based on these unique description of each classification category thus training class is created. In the subsequent testing phase, these feature space partitions are used to classify image features.

The description of training classes is an extremely important component of the classification process. In supervised classification, statistical processes based on an a priori knowledge of probability distribution functions or distribution free processes can be used to extract class descriptors. Unsupervised classification relies on clustering algorithms to automatically segment the training data into prototype classes.

In either case the motivating criteria for constructing training classes are that they are independent thus a change in the description of one training class should not change the value of another, discriminatory thus different image features should have significantly different description and reliable thus all image features within a training group should share the common definitive descriptions of that group.

The convenient way of building a parametric description of this sort is through a feature vector  $(v_1, v_2, \dots, v_n)$  where  $n$  is the number of attribute which describe each image feature and training class. This representation allows us to consider each image feature as occupying a point and each training class as occupying a sub space within the  $n$ -dimensional classification space. Viewed as such, the classification problem is that of determining to which sub-space class each feature vectors belongs.

Considering an application where we must distinguish two different types of ob-



jects (e.g. book and pen) based upon a set of two attribute classes. If we assume that we have a vision system capable of extracting these features from a set of training images, we can plot the result in the 2-D feature space.

To be able to distinguish two different types of objects base on a set of two attribute classes, one may need to decide how to numerically partition the feature space so that if we are given the feature vector of the a test object. It can be determined quantitatively to which of the two classes it belongs. On e of the most simple technique is to employ a supervised, distribution free approach know as the mean (minimum) distance classifier.

### Mean Distance Classifier

Suppose that each training class is represented by a prototype thus in our case mean vector:

$$(3.2.17) \quad m_j = \frac{1}{N_j} \sum_{x=\omega_j}^M x$$

where  $N_j$  is the number of training pattern vectors from class  $\omega_j$ . Based on this, we an assign any given pattern  $x$  to the class of its closest prototype by determining its proximity of each  $m_j$ . If Euclidean distance is our measure of proximity then the distance to the prototype is given by:

$$(3.2.18) \quad D_j(x) = \|x - m_j\| \text{ for } j = 1, 2, \dots, M$$

This is equivalent to computing

$$d_j(x) = x^T m_j - \frac{1}{2}(m_j^T m_j) \text{ for } j = 1, 2, \dots, M$$

and assigning  $x$  to class  $\omega_j$  if  $d_j(x)$  yields the largest value.

Finally, the decision boundary which separate class  $\omega_i$  from  $\omega_j$  is given by values for  $x$  for which  $d_i(x) - d_j(x) = 0$ . The surface defined by this decision boundary is the



perpendicular bisector of the line segment joining  $m_i$  and  $m_j$ . In practice the mean distance classifier works well when the distance between means is large compared to the spread or randomness of each class with respect to its mean.

It is simple to implement and is guaranteed to give an error rate within a factor of two of the ideal error rate obtained using the statistical or supervised Bayes' classifier. The Bayes' classifier is a more informed algorithm as the frequencies of occurrence of the features of interest are used to aid the classification process. Without this information the mean distance classifier can yield biased classifications. This however, can be combated by applying training patterns at the natural rates at which they arise in the raw training set.

### 3.2.10 Mathematical Morphology

Morphological operators often take a binary image and a structuring element as input and combine them using a set operator (intersection, union, inclusion, complement). They process objects in the input image based on characteristics of its shape which are encoded in the structuring element.

Usually, the structuring element is sized  $3 \times 3$  and has its origin at the center pixel. It is shifted over the image and at each pixel of the image, its elements are compared with the set of the underlying pixels. If the two sets of elements match the condition defined by the set operator, the pixel underneath the origin of the structuring element is set to a predefined value (0 or 1 for binary images). A morphological operator is therefore defined by its structuring element and the applied set operator.

#### Dilation

To dilate means to grow or expand. Dilation is one of the two basic operators in the area of mathematical morphology, the other being erosion. It is typically applied



to binary images, but there are versions that work on gray-scale images. The basic effect of the operator on a binary image is to gradually enlarge the boundaries of regions of foreground pixels. Thus areas of foreground pixels grow in size while holes within those regions becomes smaller.

The dilation operator takes two pieces of data as inputs. The first is the image which is to be dilated while the second is the set of coordinate points known as a structuring element also known as the kernel. It is this structuring element that determines the precise effect of the dilation on the input image. The mathematical definition of dilation for binary images is as follows. Suppose that  $X$  is the set of euclidean coordinates corresponding to the input binary image and that  $K$  is the set of coordinates for the structuring element.

Let  $KX$  denote the translation of  $K$  so that the origin is at  $X$ . Then the dilation of  $X$  by  $K$  is the set of all points  $X$  such that the intersection of  $KX$  with  $X$  is non-empty. The mathematical definition of gray-scale dilation is identical except for the way in which the set of coordinates associated with the input image is derived. In addition to this, the coordinates are 3-Dimensional rather than 2-Dimensional

## Erosion

To erode also means to shrink or reduce and is applied to binary images but may also work on gray-scale image. The basic effect of the operator on a binary image is to erode away the boundaries of regions of foreground pixels. Thus, area of foreground pixels shrink in size and holes within those areas become larger.

The erosion operator takes two data as inputs of which the first is the image which is to be eroded and the second is a set of coordinate points know as a structuring element or kernel. The structuring element determines the precise effect of the erosion



on the input image.

The mathematical definition of erosion for binary image is as follows. Suppose that  $X$  is the set of Euclidean coordinates corresponding to the input binary image and that  $K$  is the set of coordinates for the structuring element. Let  $KX$  denote the translation of  $K$  so that its origin is at  $X$ . Then the erosion of  $X$  by  $K$  is the set of all points  $X$  such that  $KX$  is a subset of  $X$ . Again, the mathematical definition for gray-scale erosion is identical except in the way in which the set of coordinates associated with the input image is derived and their coordinates are 3-Dimensional rather than 2-Dimensional.

### Thinning

Thinning is a morphological operation that is used to remove selected foreground pixels from binary images just like erosion or opening. It is commonly used to tidy up the output of edge detectors by reducing all lines to single pixel thickness. Thinning is normally only applied to binary images and produces another binary image as output. Just like other morphological operators the behavior of the thinning operation is determined by a structuring element. The structure elements used for thinning are of the extended type described under the hit-and-miss transform.

The thinning operation is related to the hit-and-miss transform and can be expressed in terms of it. The thinning of an image  $I$  by a structuring element  $J$  is

$$(3.2.19) \quad \text{thin}(I, J) = I - \text{ham}(I, J) \text{ where } \text{ham} = \text{hit} - \text{and} - \text{miss}$$

where the subtraction is a logical subtraction defined by  $X - Y = X \cap Y^c$ . The thinning operation is calculated by translating the origin of the structuring element to each possible pixel position in the image and at each such position comparing it with the underlying image pixels. If the foreground and background pixels in the



structuring element exactly match foreground and background pixels in the image then the image pixel underneath the origin of the structuring element is set to background otherwise it is left unchanged. The structuring element must always have a one or a blank at its origin if it is to have any effect.

The choice of structuring element determines under which situations a foreground pixel will be set to background and hence it determines the application for the thinning operation. One of the most common uses of thinning is to reduce the threshold output of the edge detectors such as the Sobel operator to lines of a single pixel thickness while preserving the full length of those lines.

### 3.2.11 Digital Filters

In image processing, filters are used to suppress either the high frequencies in the image thus smoothing the image or the low frequencies thus enhancing or detecting edges in the image. Images are often corrupted by noise from several sources and the most frequent of which are the additive noise (example is Gaussian noise) and impulse noise (e.g. salt and pepper noise). Linear filters such as the mean filters are the primary tool for smoothing digital images degraded by additive noise. An image can be filtered either in the frequency or in the spatial domain.

The former involves transforming the image into the frequency domain by multiplying it with the frequency filter function and re-transforming the result into the spatial domain. The filter function is shaped so as to attenuate some frequencies and enhance others. For example, a simple lowpass function is 1 for frequencies smaller than the cut-off frequency and 0 for all others.

The corresponding process in the spatial domain is to convolve the input image



$F(i, j)$  with the filter function  $H(i, j)$ . This can be written as:

$$(3.2.20) \quad G(i, j) = h(i, j) \odot f(i, j)$$

The mathematical operation is identical to the multiplication in the frequency space but the results of the digital implementations vary since we have to approximate the filter function with a discrete and finite kernel.

The discrete convolution can be defined as a 'shift and multiply' operation where we shift the kernel over the image and multiply its value with the corresponding pixel values of the image. For a square kernel with size  $M \times M$ , we can calculate the output image with the follow formula:

$$(3.2.21) \quad G(i, j) = \sum_{m=-\frac{M}{2}}^{\frac{M}{2}} \sum_{n=-\frac{M}{2}}^{\frac{M}{2}} h(m, n) f(i - m, j - n)$$

Various standard kernels exist for specific applications, where the size and the form of the kernel determine the characteristics of the operation.

In contrast to the frequency domain, it is possible to implement non-linear filters in the spatial domain. In this case, the summations in the convolution function are replaced with the non-linear operator:

$$(3.2.22) \quad G(i, j) = O_{m,n}[h(m, n) f(i - m, j - n)]$$

For most non-linear filters the elements of  $H(i, j)$  are all 1

### Mean Filter

Mean filtering also called smoothing or averaging filtering is a method for smoothing thus reducing the amount of intensity variation between one pixel and the next image. It is often used to reduce noise in images.

The idea of mean filtering is to replace each pixel value in an image with the average value of its neighbors including itself. This has the effect of eliminating pixel



values which are unrepresentative of their surroundings. Mean filtering like other convolution filters is based around a kernel which represents the shape and size of the neighborhood to be sampled when calculating the mean. Smaller kernels are often used. Applying a small kernel more than once produces a different effect as in single pass with large kernel.

## Median Filter

Median filter also known as the rank filter is used to reduce noise in an image. However, it preserves useful details in the image much better than the mean filter. The median filter considers each pixel in the image in turn and looks at its nearby neighbors to decide whether it is representative of its surroundings. Instead of reducing the pixel value with the mean of neighboring pixel values, it replaces it with the median of those values. The median is calculated into numerical order and then replacing the pixel being considered with the middle pixel value. If the neighborhood under consideration contains an even number of pixels, the average of the two middle pixel values is used.

## Gaussian Smoothing

The Gaussian Smoothing operator, (Davies, 1990) is a 2-Dimensional convolution operator that is used to blur images and remove detail and noise. In this sense, it is similar to the mean filter but it uses a different kernel that represents the shape of a Gaussian hump. This kernel has the following special properties:

1. The Gaussian distribution in 1-Dimension has the form

$$(3.2.23) \quad G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$$

where  $\sigma$  is the standard deviation of the distribution.



2. In 2-Dimensions, a circularly symmetric Gaussian has the form:

$$(3.2.24) \quad G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

The idea of Gaussian smoothing is to use the 2-Dimensional distribution as a point-spread function and is achieved by convolution. Since the image is stored as a collection of discrete pixels, a discrete approximation to the Gaussian function is required before convolution can be performed. In theory Gaussian distribution is zero everywhere which will require an infinitely large convolution kernel but in practice it is effectively zero more than about three standard deviations from the mean and so the kernel can be truncated at this point.

A further way to compute a Gaussian smoothing with a large standard deviation is to convolve an image several times with a smaller Gaussian. Although this is computationally complex, it can have applicability if the processing is carried out using a hardware pipeline.

### 3.2.12 Edge Detectors

Edges are places in the image with strong intensity contrast. Since edges often occur at image location representing object boundaries, edge detection is extensively used in image segmentation when dividing the image into areas corresponding to different objects. Representing an image by its edges has an advantage of reducing the amount of data significantly while retaining most of the image information.

Edge consist of mainly high frequencies and can be detected by applying a high pass frequency filter in the Fourier domain or by convolving the image with an appropriate kernel in the spatial domain. In practice, edge detection is performed in the spatial domain because it is computationally less expensive and often yields better results.



Edges again, correspond to strong illumination gradients and can therefore be highlighted by calculating the derivatives of the image. The position of the edge can be estimated with the maximum of the first derivatives or with the zero-crossing of the second derivatives. For a discrete one-dimensional function  $f(i)$ , the first derivatives is approximated by:

$$(3.2.25) \quad \frac{df(i)}{d(i)} = f(i+1) - f(i)$$

and this is equivalent to convolving the function with  $[-1, 1]$ . Similarly, the second derivative can be estimated by convolving  $f(i)$  with  $[1, -2, 1]$ . Different edge detection kernels which are based on the above formula are used in evaluating the first or second derivative of a 2-Dimensional image.

There are two common approaches: Prewitt compass edge detection and gradient edge detection used to estimate the first derivative in a 2-Dimensional image. Prewitt compass edge detection involves convolving the image with set of kernel which is usually 8 and each of which is sensitive to a different edge orientation. The kernel producing the maximum response at a pixel location determines the edge magnitude and orientation. Different sets of kernels might be used of which example includes Sobel and Robinson kernels.

Gradient edge detection is the second and more widely used technique. Here the image is convolved with only two kernels, one estimating the gradient in the x-direction,  $G_x$ , and the other gradient in the y-direction,  $G_y$ . The absolute gradient magnitude is then given by:

$$(3.2.26) \quad |G| = \sqrt{G_x^2 + G_y^2}$$

and is often approximated with

$$(3.2.27) \quad |G| = |G_x| + |G_y|$$



In many implementations, the gradient magnitude is the only output of a gradient edge detector, however the edge orientation might be calculated within

$$(3.2.28) \quad \theta = \arctan\left(\frac{G_y}{G_x}\right) - \frac{3\pi}{4}$$

The most common kernels used for the gradient edge detector are the Sobel, Robert Cross and Prewitt operators. After having calculated the magnitude of the first derivative, pixels corresponding to the edge are identified. A good way is to threshold the gradient image (Mallat, 1989) assuming that all pixels having a local gradient above the threshold must represent an edge. An alternative technique is to look for the local maximal in the gradient image, thus producing one pixel wide edge. A more sophisticated technique is used by the Canny edge detector (Canny, 1986). It first applies a gradient edge detector to the image and then finds the edge pixels using non-maximal suppression and hysteresis tracking.

A general problem for edge detection is its sensitivity to noise and the reason being that calculating the derivative in the spatial domain corresponds to accentuating high frequencies and hence magnifying noise. This problem is addressed in the Canny and Marr operators (Canny, 1986; Marr, 1982; Marr and Ullman, 1981; Marr and Hildreth, 1980), by convolving the image with a smoothing operator (Gaussian) before calculating the derivative.

The operators included in this section are those whose purpose is to identify meaningful features on the basis of distributions of pixel gray-levels. The two categories of operators included here are Edge Pixel Detectors which assigns a value to a pixel in proportion to the likelihood that the pixel is part of an image edge. The second is the Line Pixel Detectors, which also assigns a value to a pixel in proportion to the likelihood that the pixel is part of an image line.



## Sobel Edge Detector

The Sobel operator performs a 2-D spatial gradient measurement on an image and so emphasizes on regions of high spatial frequency that correspond to edges. Typically it is used to find the approximate absolute gradient magnitude at each point in an input gray-scale image. In theory, at least, the operator consists of a pair of  $3 \times 3$  convolution kernels as shown in the figure below. One kernel is simply the other rotated by  $90^\circ$  and is very similar to the Roberts Cross operators.

These kernels are designed to respond maximally to edges running vertically and horizontally relative to the pixel grid, one kernel for each of the two perpendicular orientation. The kernels can be applied separately to the input image to produce separate measurements of the gradient component in each orientation. These can

-1	0	+1	+1	+2	+1
-2	0	+2	0	0	0
-1	0	+1	-1	-2	-1
$G_x$			$G_y$		

Figure 3.1: Sobel convolution kernel

then be combined together to find the absolute magnitude of the gradient at each point and the orientation of that gradient. The magnitude of the gradient is given by:

$$(3.2.29) \quad |G| = \sqrt{G_x^2 + G_y^2}$$

An approximate magnitude is computed using

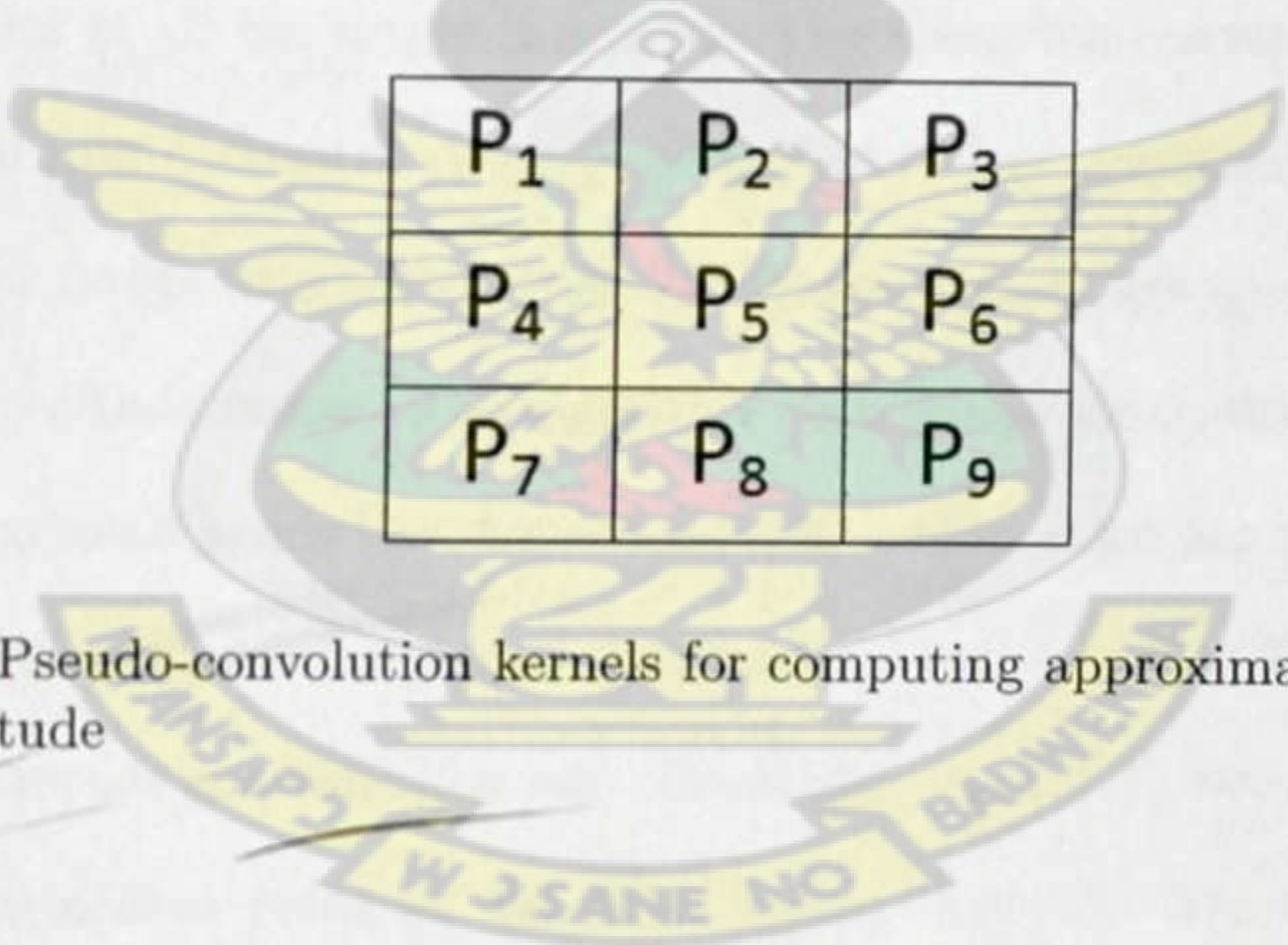
$$(3.2.30) \quad |G| = |G_x| + |G_y|$$



The angle of orientation of the edge relative to the pixel grid giving rise to the spatial gradient is written as:

$$(3.2.31) \quad \theta = \arctan\left(\frac{G_y}{G_x}\right)$$

In this case, orientation  $\theta$  is taken to mean that the direction of maximum contrast from black to white runs from left to right on the image and other angles are measured anti-clockwise from this. Often, this absolute magnitude is the only output the user sees and the two components of the gradient are conveniently computed and added in a single pass over the input image using the pseudo-convolution operator shown in Figure 3.2.12 below. Using this kernel, the approximate magnitude is given as:



P <sub>1</sub>	P <sub>2</sub>	P <sub>3</sub>
P <sub>4</sub>	P <sub>5</sub>	P <sub>6</sub>
P <sub>7</sub>	P <sub>8</sub>	P <sub>9</sub>

Figure 3.2: Pseudo-convolution kernels for computing approximate gradient magnitude

$$(3.2.32)$$

$$|G| = |(P_1 + 2 \times P_2 + P_3)| - |(P_7 + 2 \times P_8 + P_9)| + |P_3 + 2 \times P_6 + P_9| + |P_1 + 2 \times P_4 + P_7|$$

The Sobel operator is slower to compute but its large convolution kernel smoothen the input image to a greater extent and so makes the operator less sensitive to noise. This operator also generally produce considerably higher output value for similar edges as compared with the others.



## Canny Edge Detector

The Canny operator, Canny (1986) was designed to be an optimal edge detector. It takes as input a gray-scale image and produces as output an image showing the positions of tracked discontinuities. The Canny operator works in a multi-stage

-1	0	+1	+1	+1	+1
-1	0	+1	0	0	0
-1	0	+1	-1	-1	-1
$G_x$			$G_y$		

Figure 3.3: Prewitt gradient edge detector

process. First of all the image is smoothed by Gaussian convolution and then a 2-Dimensional first derivative operator is applied to the smoothed image to highlight regions of the image with high first spatial derivatives. Edges give rise to ridges in the gradient magnitude image. The algorithm then tracks along the top of these ridges and sets to zero all pixels that are not actually on the ridge top to give a thin line in the output by a process known as non-maximal suppression. The tracking process exhibits hysteresis controlled by two thresholds:  $T_1$  and  $T_2$  with  $T_1 > T_2$ . Tracking can only begin at a point on the ridge higher than  $T_1$ . Tracking then continues in both directions out from that point until the height of the ridge falls below  $T_2$ . This hysteresis helps to ensure that noisy edges are not broken up into multiple edge fragments.

The effect of the Canny operator is determined by three parameters; the width of the Gaussian kernel used in the smoothing phase, the upper and lower thresholds used by the tracker. Increasing the width of the Gaussian kernel reduces the detector's sensitivity to noise at the expense of losing some of the finer details in the image.



The localization error in the detected edges also increases slightly as the Gaussian width is increased.

Setting the lower threshold too high will cause noisy edges to break up and setting the upper threshold too low increases the number of spurious and undesirable edge fragments appearing in the output. One problem with the basic Canny operator is to do with Y-junctions thus places where three ridges meet in the gradient magnitude image. Such junctions can occur where an edge is partially occluded by another object. The tracker will treat two of the ridges as a single line segment and the third on as a line that approaches but does not quite connect to that line segment.

### Prewitt Compass Edge Detector

Prewitt Compass Edge Detection is an alternative approach to the differential gradient edge detection. The operation outputs two images, one estimating the local edge gradient magnitude and the other estimating the edge orientation of the input image.

When using this edge detection the image is convolved with a set of usually 8 convolution kernels, each of which is sensitive to edges in a different orientation. For each pixel the local edge gradient magnitude is estimated with the maximum response of all 8 kernels at this pixel location:

$$(3.2.33) \quad |G| = \max(|G_i| : i = 1 \cdots n)$$

where  $G_i$  is the response of the  $i^{th}$  kernel at the particular pixel position and  $n$  is the number of convolution kernels. The local edge orientation is estimated with the orientation of the kernel that yields the maximum response. Various kernels can be used for this operation. Two templates out of the set of 8 kernels is produced by taking one of the kernels and rotating its coefficients circularly. Each of the resulting



-1	0
0	-1

$G_x$

0	+1
-1	0

$G_y$

Figure 3.4: Prewitt compass edge detecting templates sensitive to edges at  $0^\circ$  and  $45^\circ$

kernels is sensitive to an edge orientation ranging from  $0^\circ$  to  $315^\circ$  in steps of  $45^\circ$  where  $0^\circ$  corresponds to a vertical edge.

The maximum response  $\bar{G}$  for each pixel is the value of the corresponding pixel in the output magnitude image. The values for the output orientation image lie between 1 and 8 depending in which of the 8 kernels produced the maximum response.

The compass edge detector is an appropriate way to estimate the magnitude and orientation of an edge. Although differential gradient edge detection needs a rather time consuming calculation to estimate the orientation from the magnitudes in the x-direction and y-direction, the edge detector obtains the orientation directly from the kernel with the maximum response. In our case the compass operator is limited to 8 possible orientations. However experiments, (Korner, 1996; Vetterli and Herley, 1992) shows that most direct orientation estimates are not much more accurate.

On the other hand, the compass operator needs 8 convolutions for each pixel, whereas the gradient operator needs only 2 one kernel being sensitive to edges in vertical direction and the other to the horizontal direction. The result for the edge magnitude image is very similar with both methods provided the same convolving kernel is used. There are various kernels that can be used for Compass Edge Detection and the most common ones are shown in Figure 3.2.12

For every template, the set of all eight kernels is obtained by shifting the coef-



ficients of the kernel circularly. The result for using different templates is similar; the main difference is the different scale in the magnitude image. The advantage of Sobel is that only 4 out of the 8 magnitude values must be calculated. Since each pair of kernels rotated about  $180^\circ$  opposite is symmetric, each of the remaining four values can be generated by inverting the result of the opposite kernel.

-1	0	+1	0	+1	+2
-2	0	+2	-1	0	+1
-1	0	+1	-2	-1	0

SOBEL

-3	-3	+5	-3	+5	+5
-3	0	+5	-3	0	+5
-3	-3	+5	-3	-3	-3

KIRSCH

-1	0	+1	0	+1	+1
-1	0	+1	-1	0	+1
-1	0	+1	-1	-1	0

ROBINSON

Figure 3.5: Examples compass edge detecting kernels with each showing two kernels out of the eight

## Line Detection

While edges thus boundaries between regions with relatively distinct gray-scale are by far the most common type of discontinuity in an image, instances of thin lines in an image occur frequently enough that it is useful to have a separate mechanism for detecting them. Here we present a convolution based technique which produces an image description of the thin lines in an input image.

The line detection operator consists of a convolution kernel tuned to detect the presence of lines of a particular width  $n$ , at a particular orientation  $\theta$ . Figure 3.2.12



shows a collection of four such kernel of which each respond to lines of single pixel width at the particular orientation shown. These masks are tuned for light lines

-1	-1	-1	-1	2	-1	-1	-1	2	-1	-1	2	2	-1	-1
2	2	2	-1	2	2	-1	-1	2	-1	-1	-1	-1	2	-1
-1	-1	-1	-1	2	-1	-1	2	-1	-1	2	-1	-1	-1	2
A	B	C	D											

Figure 3.6: Four line detection kernels which respond maximally to horizontal, vertical and oblique single pixel wide lines

against a dark background and would give a big negative response to the dark lines against the light background. If you are only interested in detecting dark lines against a light background, then you should negate the mask values. Alternatively, you might be interested in either kind of line in which case you could take the absolute value of the convolution output. In the discussion and examples below, we will use the kernels above without an absolute value. Let  $R_i$  denotes the response of kernel  $i$  and each of the kernels across an image. Then for any particular point, if  $R_i > R_j$  for all  $h \neq i$ , that point is more likely to contain a line whose orientation and width corresponds to that of kernel  $i$ . Usually thresholding  $R_i$  will eliminate weak lines corresponding to edges and other features with intensity gradients which have a different scale than the desired line width. In order to find complete lines, line fragments are joined together.

### 3.2.13 Distance Transform

The distance transform is an operator normally only applied to binary images. The result of the transform is a gray-scale image that looks similar to the input image, except that the gray-level intensities of points inside the foreground regions are changed to show the distance to the closest boundary from each point. The figure



below shows a distance transform for a simple rectangular shape. There is a dual to the distance transform described above which produces the distance transform for

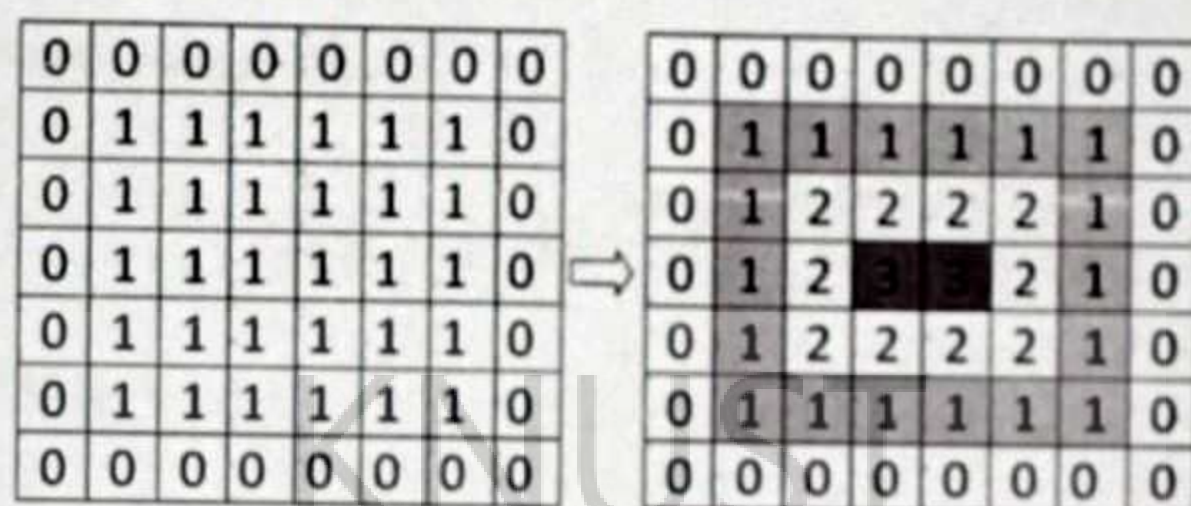


Figure 3.7: The distance transform of a simple shape using chessboard distance metric

the background region rather than the foreground region. It can be considered as a process of inverting the original image and then applying the standard transform as above.

There are several different sorts of distance transform, depending upon which distance metric is being used to determine the distance between pixels. The example shown in Figure 3.2.13 uses the chessboard distance metric but both the Euclidean and City block metric can be used as well. The distance transform can be calculated much more efficiently using algorithms in only two passes and the algorithm is based on recursive morphology.

Distance metric is useful in image processing to be able to calculate the distance between two pixels in an image, but this is not as straightforward as it seems. The presence of the grid makes several distance metric possible which often give different answers to each other for the distance between the same pair of points. Three of the most important ones are being discussed below.



## Euclidean Distance

This is the familiar straight line distance that most people are familiar with. If the two pixels that we are considering have coordinates  $(x_1, y_1)$  and  $(x_2, y_2)$ , then the Euclidean distance is given by:

$$(3.2.34) \quad D_{Euclid} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

## City Block Distance

This is also known as the Manhattan distance. This metric assumes that in going from one pixel to the other it is only possible to travel directly along pixel grid lines. Diagonal moves are not allowed. Therefore, the city block distance is given by:

$$(3.2.35) \quad D_{City} = |x_1 - x_2| + |y_1 - y_2|$$

## Chessboard Distance

This metric assumes that you can make moves on the pixel grid as if you were a King making moves in chess thus diagonal move counts the same as a horizontal move. This means that the metric is given by:

$$(3.2.36) \quad D_{Chess} = \max(|x_1 - x_2|, |y_1 - y_2|)$$

However it should be noted that the last two metrics are usually much faster to compute than the Euclidean metric and so are sometimes used where speed is critical but accuracy is not too important.

### 3.2.14 Image Synthesis

Image synthesis is the process of creating new images from some form of image description. Images that are normally synthesized include computer graphics, scenes or image based on geometric shape description. Synthetic images are often used to



verify the correctness of operators by applying them to known images and images could be binary, gray-scale or color.

### 3.3 Analysis of Fingerprint Data

Galton who turn out to be the first scientific on the study fingerprint image divided fingerprint into three major classes namely arches, loops, and whorls. Henry, later refined Galtons classification by increasing the number of classification which is well-known and widely accepted. In this view the position of cores and deltas are claimed to be enough to classify the fingerprints into six categories, which include arch, tented arch, left-loop, right-loop, whorl, and twin-loop.

1. The loops is believe to have constitute 60 to 70 percent of the patterns encountered. In a loop, one or more of the ridges enters on either side of the impression running from the delta to the core and terminates on or in the direction of the side where the ridges entered and they have exactly one delta in it. Loops that have ridges that enter and leave from left side are called the left loops and those that enter and leave from right side are called the right loops. In twin loops the ridges containing the core points have their exits on different sides.
2. In the whorl, some of the ridges make a turn through at least one circuit and any pattern containing two or more deltas is considered a whorl.
3. In arch patterns, the ridges run from one side to the other of the pattern with no backward turn and they come in two types thus plain or tented. The plain arch tends to flow easily through the pattern with no significant changes whiles the tented arch does make a pronounced change with no easy flow.



In simple terms, classification of fingerprints are made with the assumption that patterns without delta is an arch, a loop when it contains one and only one delta and always a whorl when it contains 2 or more delta. On the order ranking, fingerprint ridge details are predominantly described at three levels namely, patterns, minutiae points and pores and ridge shape forming the level one, two and three respectively however Automated Fingerprint Identification Systems (AFISs) only admits Level one and Level two features. No two fingerprints are alike, but the pattern of fingerprint is inherited from close relatives and people in our immediate family and that is considered "level one details". The details of an actual finger or palm print are not inherited which is deemed "level 2 and level 3 details" and is used to identify fingerprints from person to person. The method below explained how this is achieved:

#### 1. SEGMENTATION

Image segmentation is the first step in fingerprint enhancement algorithm and is the process of separating the foreground regions from the background regions of an image. The foreground regions usually correspond to the clear fingerprint area having the ridges and valleys which is the area of interest while the background corresponds to the regions outside the borders of the fingerprint area, which do not contain any valid fingerprint information. Cropping out of these regions (background) is made to minimize the number of operations to undertake on the fingerprint image. Exhibition of a very low gray-scale variance value is profound in the background regions of a fingerprint image whereas the foreground regions have a very high variance hence a method based on variance threshold can be used to perform the segmentation. The steps involved in these mean and variance based segmentation are illustrated as below:



(a) Firstly, the image  $I(i, j)$  is divided into non overlapping blocks of size  $w \times w$ .

(b) The mean value  $M(I)$  is then evaluated for each block using the following equation:

$$M(I) = \frac{1}{w^2} \sum_{i=-w/2}^{w/2} \sum_{j=-w/2}^{w/2} I(i, j)$$

(c) The mean value calculated above is then used to find the variance using the equation below:

$$V(I) = \frac{1}{w^2} \sum_{i=-w/2}^{w/2} \sum_{j=-w/2}^{w/2} (I(i, j) - M(i, j))^2$$

(d) If the variance is less than the global threshold value selected empirically, then the block is assigned to be a background region otherwise it is assigned to be part of the foreground.

## 2. NORMALIZATION

Next to segmentation is normalization, which is a process that changes the range of pixel intensity values. For instance if the intensity range of the image is 50 to 180 and the desired range is 0 to 255 then the process entails subtracting 50 from each of pixel intensity resulting in the range of 0 to 130. Then each pixel intensity is multiplied by  $255/130$ , giving the range 0 to 255. Let  $I(i, j)$  denote the gray-level value at pixel  $(i, j)$ ,  $M_0$  and  $V_0$  denote the estimated mean and variance of  $I$ , respectively, and  $N(i, j)$  the normalized gray-level value at pixel  $(i, j)$ . The normalized image is defined as follows:

$$N(i, j) = \begin{cases} M_0 + \sqrt{V_0(I(i, j) - M_i)^2} & \text{if } I(i, j) < M \\ M_0 - \sqrt{V_0(I(i, j) - M_i)^2} & \text{otherwise} \end{cases}$$

Normalization is a pixel-wise operation and does not change the clarity of the ridge and valley structures. The main purpose of normalization is to reduce



the variations in gray-level values along ridges and valleys, which facilitates the subsequent processing steps.

### 3. ORIENTATION ESTIMATION

The orientation flow is then estimated from the least square method using the following equations after dividing the input image  $I$  into non overlapping blocks of size  $w \times w$  and then computing the gradients  $\delta_x$  and  $\delta_y$  at each pixel.

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

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

Where  $\delta_x(u, v)$  and  $\delta_y(u, v)$  represents gradient magnitudes at each pixel in  $x$  and  $y$  directions respectively. The direction of block centered at pixel  $(i, j)$  is then computed using the following equation:

$$\theta(i, j) = \frac{1}{2} \arctan \left( \frac{V_x(i, j)}{V_y(i, j)} \right)$$

Due to the presence of noise, corrupted ridge and valley structures in the input image, the estimated local ridge orientation,  $\theta(i, j)$ , may not always be correct hence a low-pass filter is used to modify the corrupted local ridge orientation.

### 4. SMOOTHING AND FINE TUNING

Smoothing of ridges is performed, which is a process of finding out the valid frequency of the binary image of ridges. Filters corresponding to these distinct frequencies and orientations are then generated. The direction of gravity of progressive blocks are then determined, using the following equations given

( $P = 3$ ):

$$A = \sum_{k=0}^{P-1} \sum_{l=0}^{P-1} V_x \quad \text{and} \quad B = \sum_{k=0}^{P-1} \sum_{l=0}^{P-1} V_y$$



Given the fact that singular points are points where the orientation fields are discontinuous, makes orientation play a crucial role in estimating the core point of a fingerprint image. Hence, the need for another mechanism to fine tune the orientation field so as to avoid any spurious core points and irregularities that may occur due to the presence of noise. The orientation field for coarse core point is fine tuned by adjusting the orientation using the following proceeds:

If:  $B(i, j) \neq 0$  then:  $\theta = 0.5 \tan^{-1}(B/A)$

else:  $\theta = \pi/2$

if  $\theta < 0$  then

if:  $A < 0$  then:  $\theta = \theta + \pi/2$

else:  $\theta = \theta + \pi$

else if  $A < 0$  then:  $\theta = \pi/2$

The  $\theta$  value is then calculated which gives the orientation of the image.

The above are the essential steps involved in the overall fingerprint recognition process however an accurate estimation of such parameters is what researchers are still aiming high at achieving so as to make precise conclusion in fingerprint identification and recognition. In the year 2007 Y. Wang et al. propose an enhanced gradient-based algorithm for coarse estimation of fingerprint orientation fields. The basic idea of the scheme was to estimate the dominant orientation of a base block from its four overlapping neighborhoods.

In this method, gradient vectors are first calculated by taking the partial derivatives of image intensity at each pixel in the Cartesian coordinates which is denoted as  $[g_x, g_y]^T$  and this orientation maps represent the directions of ridge flow in regular spaced grid. The magnitude of these fields are omitted since only the angle information is of interest and is commonly represented as  $\{\theta_{xy}\}$ , where  $\theta_{xy} \in [0, \pi]$ , however



the orientation  $\theta$  is orthogonal to  $\bar{\varphi}$  where  $\bar{\varphi}$  is the dominant gradient angle of a local base block. Since a ridge line has two edges, the gradient vectors at both sides of a ridge are opposite to each other and calculation of  $\bar{\varphi}$  by taking just the average of gradient angles directly in a local base block may likely lead to the cancellation of each other. To solve this problem, Kass and Witkin (1987) proposes a simple but clever idea of doubling the gradient angles before the averaging process so as  $(\varphi + \pi)$  becomes  $(2\varphi + 2\pi)$  which is equal to  $2\varphi$  however, in practice,  $2\varphi$  are the angles of squared gradient vectors  $[g_{sx}, g_{sy}]^T$  that has the following relation with  $[g_x, g_y]^T$  according to trigonometric identities:

$$\begin{bmatrix} g_{sx} \\ g_{sy} \end{bmatrix} = \begin{bmatrix} g^2 \cos 2\varphi \\ g^2 \sin 2\varphi \end{bmatrix} = \begin{bmatrix} g^2(\cos^2 \varphi - \sin^2 \varphi) \\ g^2(2 \sin \varphi \cos \varphi) \end{bmatrix} = \begin{bmatrix} g_x^2 - g_y^2 \\ 2g_x g_y \end{bmatrix}$$

The average squared gradient  $[\bar{g}_{sx}, \bar{g}_{sy}]$  of a block specified by a window size  $W$  can therefore be calculated by

$$(3.3.37) \quad \begin{bmatrix} \bar{g}_{sx} \\ \bar{g}_{sy} \end{bmatrix} = \begin{bmatrix} \sum_W g_{sx} \\ \sum_W g_{sy} \end{bmatrix} = \begin{bmatrix} \sum_W (g_x^2 - g_y^2) \\ \sum_W 2g_x g_y \end{bmatrix}$$

but conventional gradient-based methods divide the input fingerprint image into equal sized blocks of  $N \times N$  pixels, and average over each block independently. The direction of orientation field in a block  $B$  can be calculated by

$$(3.3.38) \quad \theta_B = \frac{1}{2} \arctan \left( \frac{\sum_{i=1}^N \sum_{j=1}^N 2g_x(i, j)g_y(i, j)}{\sum_{i=1}^N \sum_{j=1}^N g_x^2(i, j) - g_y^2(i, j)} \right) + \frac{\pi}{2}$$

but it should be noted that function  $\arctan(*)$  gives an angle value whose range is  $[-\pi, \pi)$  and corresponds to the squared gradients, whiles  $\theta_B$  is the desired orientation angle between  $[0, \pi)$ . In order to measure the reliability of estimation for  $\theta_s$ , Kass and Witkin (1987) introduced a metric called *coherence* which calculates the strength



of the averaged gradient in the distribution of local gradient vectors and can be evaluated (Kass and Witkin, 1987) as

$$(3.3.39) \quad Coh_B = \frac{|\sum_{i=1}^N \sum_{j=1}^N (g_{sx}(i, j), g_{sy}(i, j))|}{\sum_{i=1}^N \sum_{j=1}^N |g_{sx}(i, j), g_{sy}(i, j)|}$$

Instead of using equal-sized blocks, Jain et al. (1997) proposes a ranked scheme to adjust the estimation resolution of local orientation fields via iterative steps by introducing the concept of consistency level which is a normalized deviation that compares the orientation estimates of the other blocks in a neighborhood from the centered objective block. If the consistency level is above a certain threshold, the local orientations around the objective block are re-estimated with a smaller window size.

However, if the window size is small, the ridge lines over the whole site becomes parallel and slowly varied since the general parallel and anisotropy properties of the ridge patterns from the four overlapping neighborhoods of the target block are

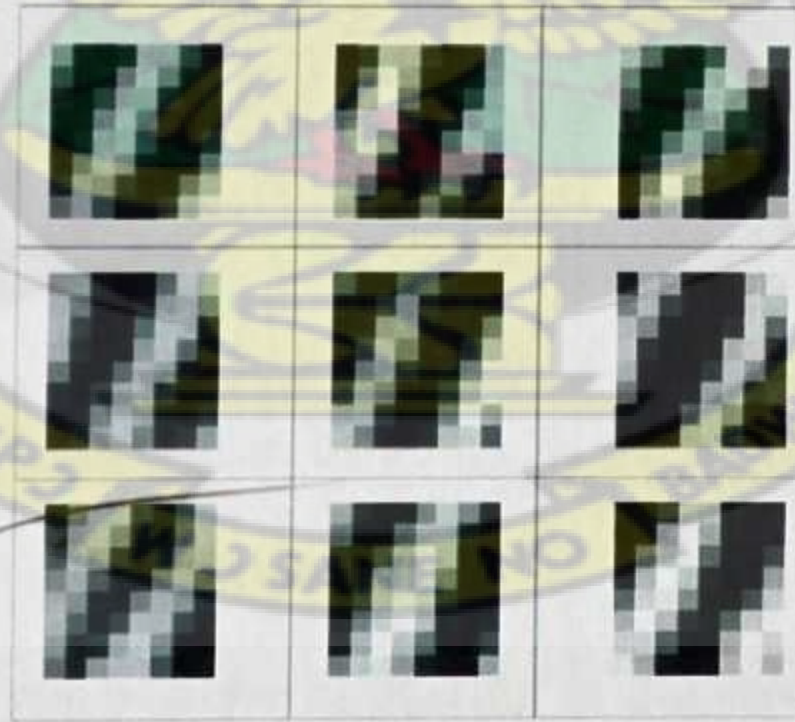


Figure 3.8: Example of a real  $3 \times 3$  fingerprint image block site: paralleled ridge pattern.

highly correlated to each other as shown in Figure 3.8. This outstanding feature of fingerprint ridge is what Y. Wang et al. tried to exploit by enhancing the traditional gradient-based methods and that was achieved by first grouping every  $2 \times 2$



interconnecting blocks into a neighborhood  $D$  and every  $3 \times 3$  interconnecting blocks into a site with the four neighborhoods overlapping with each other. An example of such formulation is illustrated in Figure 3.9. Taking into consideration the centered block  $V$  in Figure 3.9 for which the dominant orientation field is to be estimated, let blocks  $\{I, II, IV, V\}$  form a neighborhood marked  $D_1$ . Similarly, blocks  $\{II, III, V, VI\}$  form  $D_2$ , blocks  $\{IV, V, VII, VIII\}$  form  $D_3$ , and blocks  $\{V, VI, VIII, IX\}$  form  $D_4$  but should be taken to notice that the target block  $V$  is included in all four neighborhoods but placed at different corners. From Eqn 3.3.38 and 3.3.39, the averaged square gradients and their corresponding coherence measure can be calculated for each neighborhood  $D_a, D_b, D_c, D_d$ , respectively while the result yield two paired vectors  $\theta = \{\theta_a, \theta_b, \theta_c, \theta_d\}$  and  $Coh = \{Coh_a, Coh_b, Coh_c, Coh_d\}$ . The maximum value in  $Coh$  is selected and assigned to the corresponding angle in  $\theta$  of the centered block  $IV$ . For instance, if

$$Coh_b = \max\{Coh_a, Coh_b, Coh_c, Coh_d\},$$

then

$$\theta_5 = \theta_b.$$

Here, noise must be well inferred in order to be better accept how this scheme can improve noise resistance of the estimation.

The most common type of noise appearing in fingerprint images is the smudges that link parallel ridges together and these are usually caused by dirt and result of poor separation in ridge lines. By assumption, we claim the presence of dirt is uncorrelated and the random noise follows symmetric distribution with zero mean. However the ridge lines on the contrary have strong directions, meaning, the variation of gray intensities along a ridge flow is much slower than that in its perpendicular di-



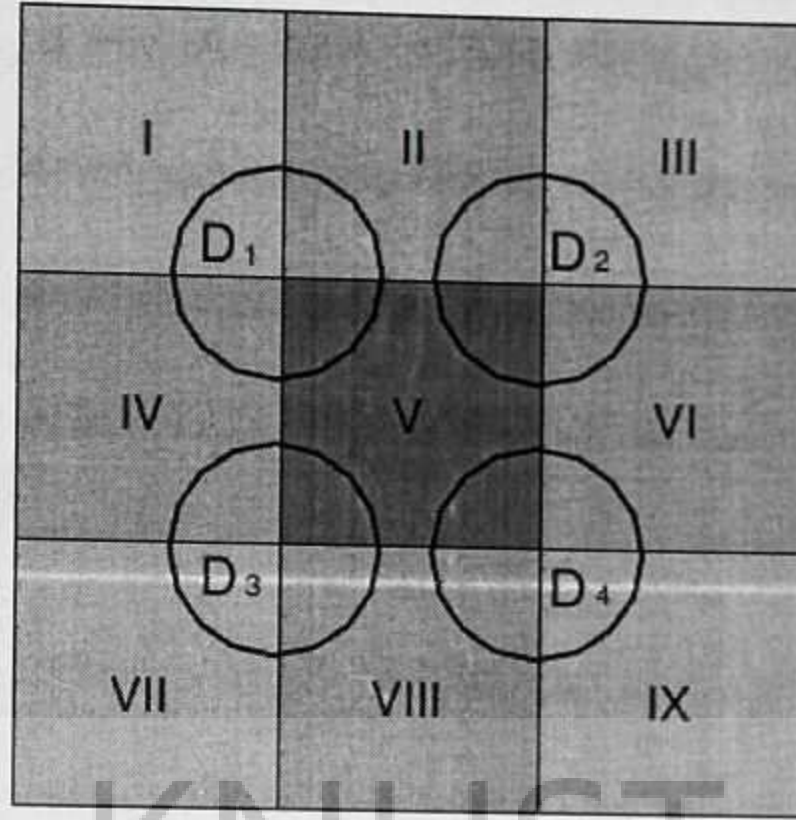


Figure 3.9: A four overlapping neighborhoods  $D_1, D_2, D_3, D_4$  of  $3 \times 3$  bloc with target block  $V$  in the center.

rection and this feature is termed anisotropy (Kass and Witkin, 1987). In a smudged area, the anisotropy feature of ridge structures is corrupted by the random noise whose orientation pattern is symmetric distributed over all directions making the resulting orientation estimate of the contaminated area puzzled.

Consider a noisy region of pixels  $W$ , the gradient vector at each pixel is composed of true gradients of the original ridges  $[g_{tx}, g_{ty}]^T$  and the disturbance gradients of noise factor  $[\Delta g_x, \Delta g_y]^T$ . That is,

$$(3.3.40) \quad \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} g_{tx} + \Delta g_x \\ g_{ty} + \Delta g_y \end{bmatrix}$$

Substituting Eqn. (3.3.40) into Eqn. (3.3.37) yields

$$\begin{bmatrix} \bar{g}_{sx} \\ \bar{g}_{sy} \end{bmatrix} = \begin{bmatrix} \sum_W g_{sx} \\ \sum_W g_{sy} \end{bmatrix} = \begin{bmatrix} \sum_W ((g_{tx} + \Delta g_x)^2 - (g_{ty} + \Delta g_y)^2) \\ \sum_W 2(g_{tx} + \Delta g_x)(g_{ty} + \Delta g_y) \end{bmatrix}$$

and above expression is evaluated for  $\bar{g}_{sx}$  as

$$(3.3.41) \quad \bar{g}_{sx} = \sum_W (g_{tx}^2 - g_{ty}^2) + 2 \left( \sum_W g_{tx} \Delta g_x - \sum_W g_{ty} \Delta g_y \right) + \left( \sum_W \Delta g_x^2 - \sum_W \Delta g_y^2 \right)$$



Since the variances of random noise in  $x$  and  $y$  directions are the same, the third term in the above equation can be crossed out. Therefore, Eqn. (3.3.41) can be simplified as

$$(3.3.42) \quad \bar{g}_{sx} = \sum_W (g_{tx}^2 - g_{ty}^2) + 2 \left( \sum_W g_{tx} \Delta g_x - \sum_W g_{ty} \Delta g_y \right)$$

where the first term is the expected value for  $\bar{g}_{sx}$  while the second term is the difference between the covariances of noise and the two original gradients. Similarly, the expression  $\bar{g}_{sy}$  can also be evaluated to get

$$(3.3.43) \quad \begin{aligned} \bar{g}_{sy} &= \sum_W 2(g_{tx} + \Delta g_x)(g_{ty} + \Delta g_y) \\ &= 2 \sum_W g_{tx} g_{ty} + 2 \left( \sum_W g_{tx} \Delta g_y + \sum_W g_{ty} \Delta g_x \right) + 2 \sum_W \Delta g_x \Delta g_y \end{aligned}$$

Since the original gradients and the noise gradients are independent variables on the orthogonal directions, the covariances in the second term above are equal to

$$\begin{aligned} \sum_W g_{tx} \Delta g_y &= \sum_W g_{tx} \cdot \sum_W g_{ty} = 0 \\ \sum_W g_{ty} \Delta g_x &= \sum_W g_{ty} \cdot \sum_W g_{tx} = 0 \end{aligned}$$

taking into account that the mean of the noise distribution is zero. Therefore, Eqn.

(3.3.43) can be simplified as

$$(3.3.44) \quad \bar{g}_{sy} = 2 \sum_W g_{tx} g_{ty} + 2 \sum_W \Delta g_x \Delta g_y$$

If the noise factor is very small (i.e.,  $\Delta g_y \rightarrow 0, \Delta g_x \rightarrow 0$ ), the last terms in both Eqns. (3.3.42) and (3.3.44) approaches zero, making the estimation closer to the expected value. However, if the previous assumption of noise distribution is no longer symmetric with zero mean, the cross terms in Eqns. (3.3.41) and (3.3.43)



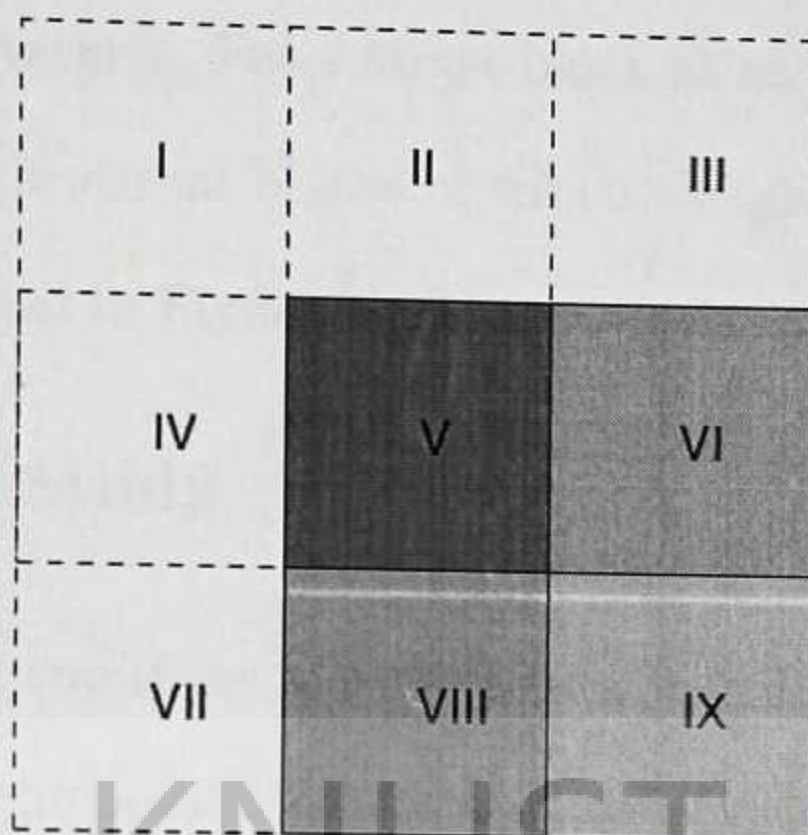


Figure 3.10: A patched site where the target block is at an edge.

cannot be canceled. For such cases, the estimates obtained from averaging over the same area can suffer from strong noise effects. Since fingerprints can be viewed as smooth and continuous textures, ridge flows that pass through a base block are most likely to extend into one of the four neighborhoods as was defined previously. In this approach,  $\bar{g}_{sx}$  and  $\bar{g}_{sy}$  of each base block was estimate over four times rather than only once. Taking the example shown in Figure (3.9), the four estimates would be  $\bar{g}_{s(x,y)|D_a}$ ,  $\bar{g}_{s(x,y)|D_b}$ ,  $\bar{g}_{s(x,y)|D_c}$  and  $\bar{g}_{s(x,y)|D_d}$  and the best estimation is then selected from the least noise-affected neighborhood according to the reliability measure of coherence. Since the coherence measure in a region is influenced by the local noise effect, the redundant estimation from four different but overlapping neighborhoods is able to give better results. Another interesting point is that this enhanced scheme actually separate the estimation window and the one that watches noise, that is, the averaging process is taken over each neighborhood whiles the noise is controlled in the bigger site that surrounds the centered objective block. In order to eliminate noise, it is better to have larger window size to include more information of the ridge structures. A smaller estimation window is desired to maintain the resolution



especially for higher curvatures. For a target block at an edge, the scheme is applied by patching the site with virtual blocks of which the gradients are assumed to be zeros and this is illustrated in Figure (3.10)

### 3.4 Focus of the Study

Although the Fourier transform is a useful tool for analyzing the frequency components of a signal, has the problem of its inability to tell at what time a particular frequency rises or falls since it take over the whole time axis at a time making it global. The Short-Time Fourier Transform thus STFT in attempt to solve this problem introduced the concept sliding window, which gives information in both time and frequency. However, another problem arises where the length of the window limits the resolution of the frequency. At this point the Wavelet Transform became the suggested solution to the above stated problem and this transform is based on small wavelets with limited duration.

#### 3.4.1 Overview of the Method

Assume the sequences  $\{a(k)\}_{k=-\infty}^{\infty}$ , which belong to the space  $l_1 \cup l_2$ , the discrete-time signals. The  $z$  transform of a signal  $\{a(k)\}$  is defined as  $A(z) = \sum_{k=-\infty}^{\infty} z^{-k} a(k)$  where  $z = e^{iw}$ . Given a signal  $x = \{x(k)\}_{k=-\infty}^{\infty}$ , a lifting scheme of the wavelet transform can be constructed either in the primal or dual mode with the following steps:

1. Decomposition: Generally, the primal lifting scheme for decomposition of signals consists of four steps: a) split; b) predict; c) update or lifting; and d) normalization. Let  $x$  be a signal and  $X(z)$  its  $z$  transform.

(a) Split: The signal  $x$  is split into even and odd subarrays:  $e_1 = \{e_1(k) =$



$x(2k)\}$ ,  $o_1 = \{o_1(k) = x(2k + 1)\}$ ,  $k \in \mathbb{Z}$ . In the  $z$ -transform domain, this operation corresponds to the following relation:  $E_1(z) = (X(z) + X(-z))/2$ ,  $O_1(z) = z(X(z) - X(-z))/2$ , where  $E_1(z)$  and  $O_1(z)$  denote the  $z$ -transforms of  $e_1$  and  $o_1$  respectively.

(b) Predict: The even array  $e_1$  is used to predict the odd array  $o_1$ . Then, the new odd array  $O_1^v$  is defined as the difference between the existing array  $o_1$  and the predicted one. To be specific, we apply some prediction filter  $U$  to the array  $e_1$ , in order for the result to approximate the array  $o_1$ . Then, we subtract this result from the array  $o_1$  :  $O_1^v(z) = O_1(z) - U(z)E_1(z)$ . This denote the new array provided the filter  $U$  is properly chosen and this step results in decorrelation of the signal.

(c) Update/Lifting: The even array is updated using the new odd array that is being convolved with the update filter whose transfer function we denote by  $V(z)/2$  :  $E_1^v(z) = E_1(z) + (1/2)V(z)O_1^v(z)$ . Generally, the goal of this step is to eliminate high frequencies which appears when the original signal  $x$  is downsampled into  $e_1$ . By doing so  $e_1$  is transformed into a downsampled and smoothed (low-pass filtering) replica  $E_1^v$  of  $x$ .

(d) Normalization: Finally, the smoothed array  $s_1$  and the array of details  $d_1$  are obtained from the following operation:  $s_1 = \sqrt{2}E_1^v$ ,  $d_1 = O_1^v/\sqrt{2}$ . The key issue in this lifting scheme is how to properly choose the filters  $U$  and  $V$ .

2. Reconstruction: The reconstruction of the signal  $x$  from the arrays  $s_1$  and  $d_1$  is implemented in a reverse order.

(a) Undo Normalization:  $E_1^v = s_1/\sqrt{2}$  and  $O_1^v = \sqrt{2}d_1$ .



- (b) Undo Update/Lifting: The even array  $E_1(z) = E_1^v(z) - V(z)O_1^v(z)/2$  is restored.
- (c) Undo Undo Predict: The odd array  $O_1(z) = O_1(z) + U(z)E_1(z)$  is restored.
- (d) Undo Split: It is the standard restoration of the signal from its evenx and odd components. In the  $z$  domain, it appears as  $X(z) = E_1(z) + z^{-1}O_1(z)$ .

### 3. Dual Mode

- (a) Update: The even array is averaged with the filtered odd array  $E_1^v(z) = (E_1(z) + V(z)O_1(z))/2$
- (b) Predict: The odd array is predicted with the filtered new even array  $O_1^v(z) = O_1(z) - U(z)E_1^v(z)$ . The reconstruction is achieved in the reverse order.

The scheme presented above yield efficient algorithms for the implementation of the forward and backward transforms of the signal  $x \leftrightarrow s_1 \cup d_1$ . For completeness, we denote  $\phi(z) = (1 + z^{-1}U(z^2))/2$  and define the following filters:

$$\begin{aligned}\tilde{G}(z) &= \sqrt{2}z\phi(-z), & \tilde{H}(z) &= \sqrt{2}(1 + zV(z^2)\phi(-z)) \\ H(z) &= \sqrt{2}\phi(z), & G(z) &= \sqrt{2}z^{-1}(1 - zV(z^2)\phi(z))\end{aligned}$$

where,  $\tilde{H}(z)$  and  $\tilde{G}(z)$  are the low- and high-pass primal analysis filters respectively and  $H(z)$  and  $G(z)$  are the low- and high-pass primal synthesis filters respectively.

The decomposition and reconstruction equations can be represented as follows:

$$\begin{aligned}S_1(z^2) &= \frac{1}{2}(\tilde{H}(z)X(z) + \tilde{H}(-z)X(-z)) \\ D_1(z^2) &= \frac{1}{2}(\tilde{G}(z)X(z) + \tilde{G}(-z)X(-z)) \\ X(z) &= H(z)S_1(z^2) + G(z)D_1(z^2)\end{aligned}$$



where  $S_1(z)$  and  $D_1(z)$  are the  $z$  transforms of the arrays  $s_1$  (smooth) and  $d_1$  (details) respectively.

In this work, the Reverse Bi-orthogonal mother wavelet (B-Spline) in versions of 3.1, 3.9 and 4.4 was used to give the directional characteristics of the decomposed fingerprint image clearly. Precisely, the LH band represents the vertical information, the HL band gives the horizontal characteristics, the LL band shows the overall information of the original fingerprint while the HH represent the diagonal characteristics. Wavelet transform is always defined in terms of a 'mother' wavelet  $\psi$  and a scaling function  $\phi$  along with their dilated and translated versions. In 2D, scaling function  $\phi(x, y)$  and mother wavelet  $\psi(x, y)$  is defined as tensor products of the following 1 - D wavelets  $\psi(x)$ ,  $\psi(y)$  and scaling functions  $\phi(x)$ ,  $\phi(y)$  as shown with equation below.

$$(3.4.45) \quad \text{Scaling Function: } \phi(x, y) = \phi(x) \times \phi(y)$$

$$(3.4.46) \quad \text{Vertical Wavelets: } \psi^y(x, y) = \phi(x) \times \psi(y)$$

$$(3.4.47) \quad \text{Horizontal Wavelets: } \psi^x(x, y) = \psi(x) \times \phi(y)$$

$$(3.4.48) \quad \text{Diagonal Wavelets: } \phi^d(x, y) = \psi(x) \times \psi(y)$$

The use of wavelet transform on image shows that the transform can analyze singularities easily that are horizontal, vertical or diagonal. This gives the hope to using the directional resolving power of wavelet in the fingerprint recognition to track the variation in orientation of fingerprint ridges. Wavelets provides time-scale map of any signal and can provide extraction of features that vary in time. The above mentioned features makes wavelet an ideal tool for analyzing signals of temporary or non-stationary nature. Hence the use of wavelet in fingerprint recognition system to increases performance of system.



### 3.5 Fingerprint Feature Extraction and Methods Involved

Directional information obtained from wavelets does not represent all directions and hence cannot be used directly as the feature vectors to represent a given fingerprint. For this reason, the need for a feature vector which describes in detail the directional information contained in the fingerprint is desired. To the satisfaction of this, the gradient of Gaussian and coherence is applied to the wavelet decomposed image. The

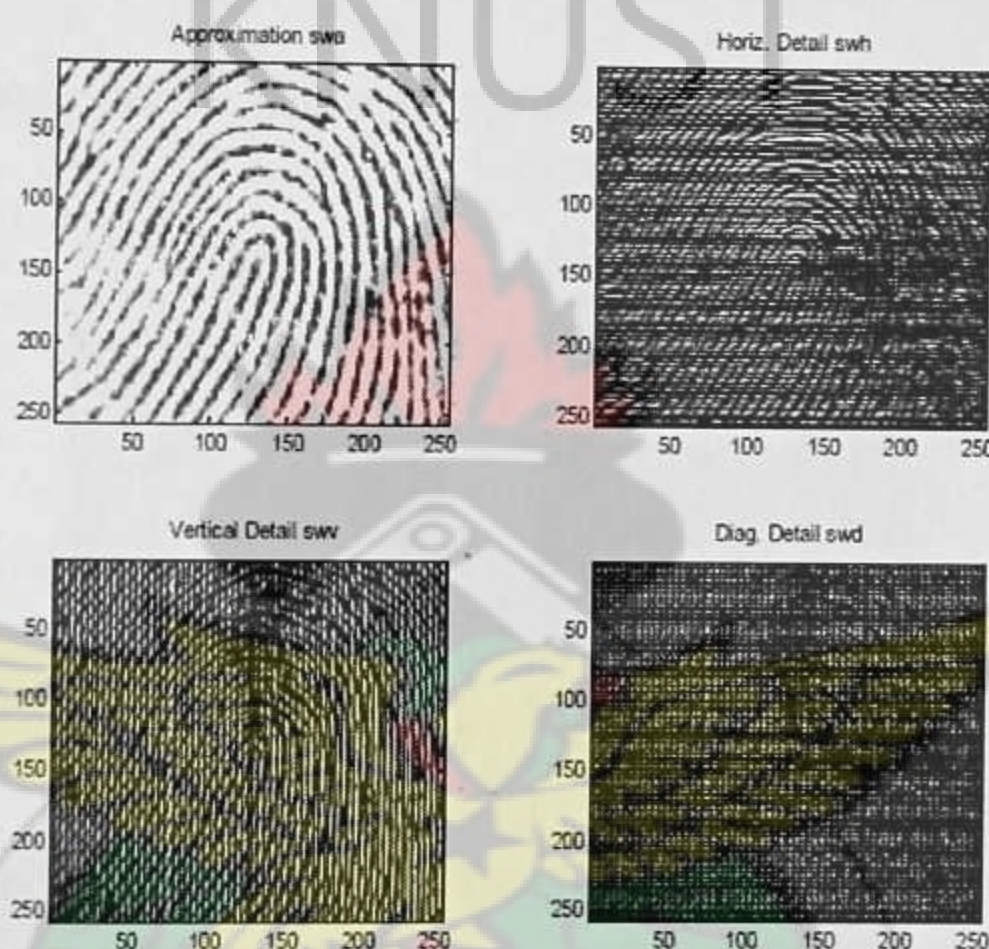


Figure 3.11: Fingerprint Wavelet Decomposition.

gradient  $G_{mn}$  and its corresponding angle  $\theta_{mn}$  at the position  $(m, n)$  are defined as

$$(3.5.49) \quad G_{mn} = (|G_{mn}^x| + |G_{mn}^y|)$$

$$\theta_{mn} = \tan^{-1} \left[ \frac{G_{mn}^y}{G_{mn}^x} \right]$$

The quantities  $G_{mn}^x$  and  $G_{mn}^y$  represent the components of  $G_{mn}$  in the horizontal and vertical directions respectively. Once  $G_{mn}$  and  $\theta_{mn}$  are obtained, the coherence is computed next. The coherence  $\rho_{mn}$  is defined as

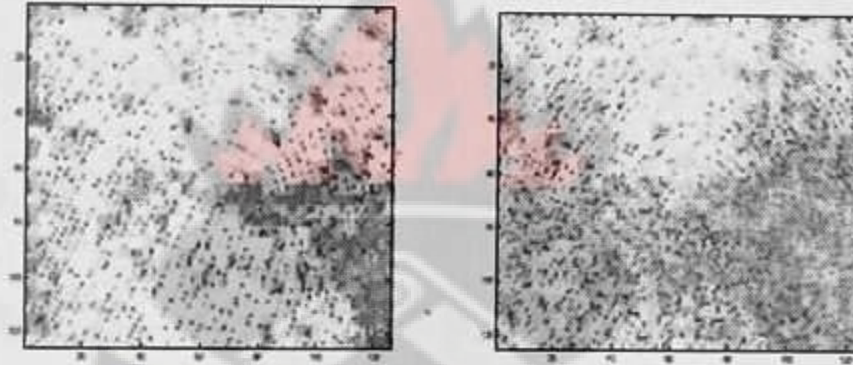
$$(3.5.50) \quad \rho_{mn} = \frac{\sum_{(i,j) \in w} G_{ij} \cos(\theta_{mn} - \theta_{ij})}{\sum_{(i,j) \in w} G_{ij}}$$



with window  $w$  of size  $(5 \times 5)$  whiles the images of the coherence are shown in Figure (3.12). From thence, the dominant local orientation  $\theta$  is calculated from the gradient and coherence and this is defined as

$$(3.5.51) \quad \theta = \frac{1}{2} \tan^{-1} \left( \frac{\sum_{m=1}^M \sum_{n=1}^N \rho_{mn}^2 \sin(2\theta_{mn})}{\sum_{m=1}^M \sum_{n=1}^N \rho_{mn}^2 \cos(2\theta_{mn})} \right) + \frac{\pi}{2}$$

Where,  $M$  and  $N$  are equal to 8. Thus, each  $8 \times 8$  window represents one directional information. All the sub band of the fingerprint image after applying wavelet is con-



A: Approximate image B: Horizontal details

Figure 3.12: Coherence Image of the Approximation image and Horizontal Details.

sidered for the center area features. The center point for the sub band is made fixed by considering only the pixel with the maximum variance along rows and columns. A  $16 \times 16$ -sized window is taken around the center point where the Correlation, Contrast, Homogeneity and Energy are determined forming another set of fingerprint features. Below are the explanations and formulations to the individual components constituting the center area features.

1. **CORRELATION:** It is a measure of how correlated a pixel is to its neighbor over the whole image. The range of values taken on by the Gray-Level Co-occurrence Matrix (GLCM) is  $[-1, 1]$ . Correlation is 1 or -1 for a perfectly positively or negatively correlated image. Correlation is NaN thus Not-a-Number for a



constant image and is given by the Equation 3.5.52

$$(3.5.52) \quad \text{Correlation} = \sum_{i=1}^N \sum_{j=1}^N \frac{(i - \mu_i)(j - \mu_j)P(i, j)}{\sigma_i \sigma_j}$$

2. **ENERGY:** It is the sum of squared elements in the GLCM. The range for GLCM is given by  $[0, 1]$ . Energy is 1 for a constant image and Equation 3.5.53 gives the Energy of the center area of the fingerprint image.

$$(3.5.53) \quad \text{Energy} = \sum_{i=1}^N \sum_{j=1}^N P(i, j)^2$$

3. **CONTRAST:** It is a measure of the intensity contrasts between pixels and its neighbors and that is given by Equation 3.5.54. The range for Gray-Level Co-occurrence Matrix is given by  $[0, (\text{size}(\text{GLCM}, 1) - 1)^2]$  and is zero for a constant image.

$$(3.5.54) \quad \text{Contrast} = \sum_{i=1}^N \sum_{j=1}^N (|i - j|)^2 P(i, j)$$

4. **HOMOGENEITY:** It is the measures the closeness of the distribution of elements in the GLCM to the diagonal. The range for gray-level co-occurrence matrix is  $[0, 1]$  and is 1 for a diagonal GLCM. By using Equation 3.5.55, homogeneity of center area in the fingerprint image can be evaluated.

$$(3.5.55) \quad \text{Homogeneity} = \sum_{i=1}^N \sum_{j=1}^N \frac{P(i, j)}{1 + (i - j)}$$

For fingerprint image alignment, the Core and Delta which are the singular points are used. Many are the methods that exist for detecting these singular points but the most popular one is based on Poincar index. Figure 3.13. shows a typical core point detected in a fingerprint image





Figure 3.13: Core Point Position in Image.

### 3.6 Fingerprint Recognition

A biometric recognition system usually runs into two distinct modes thus identification or verification.

1. **IDENTIFICATION:** It is the means of trying to find out a person's identity by examining the biometric pattern calculated from the person's biometric features. In the case of identification, the system is trained with the patterns of several persons and for each of the persons, a template is calculated. Patterns to be identified is matched against every known template which yield a score or a distance describing likeness between the pattern and the template. The system sets the pattern to the person with the most similar template and to avoid impostor patterns from being correctly identified, the agreement has to exceed a certain level. If this level is not reached, the pattern is rejected.
2. **VERIFICATION:** In this process, a person's identity is asserted a priori and only patterns to be verified are collated with the individual templates. In like manner, system is checked whether the similarity between pattern and template is adequate to provide access to secured area. Scores also known as weights and



distance transform are used to declare the similarity between the pattern and the template. The higher the score or the smaller the distance is, the higher is the similarity between them.

As explained above, access is only granted to secured areas if the score for a trained person or the person whose pattern is to be verified against is higher than a particular threshold. In theory, genuine scores (distance) should always be greater (less) than the scores (norm) of the impostors. If this would be true then a single threshold that separates the two bands of scores could be used to discriminate between genuine and impostors.

### 3.6.1 Performance of the Recognition Systems

In the real world of biometry, for several reasons, this assumption as stated above ever hardly holds since there are some cases where impostor patterns generate scores that are higher than the scores of some genuine patterns. This situation however present a fact that no matter how the threshold is selected some classification errors will still occur. Higher threshold may be chosen such that no impostor scores will exceed that limit leading to no patterns falsely accepted by the system however genuine patterns with scores lower than the highest impostor scores are falsely rejected. In opposition to this, one may choose a very low threshold such that no genuine patterns are falsely rejected but may lead to some impostor patterns falsely accepted. As this point a considerable threshold may be chosen somewhere between these two points where both false rejections and false acceptances may occur. With reference to the impostor patterns, assume a test data consisting both impostor and genuine patterns. As depicted in Figure 3.14, the belonging scores would be somehow spread around a certain mean score depending on the choice of the classification threshold



between all and none of the impostor patterns are falsely accepted by the system. The threshold depending fraction of the falsely accepted patterns divided by the number of all impostor patterns is called False Acceptance Rate (FAR). Its value is one, if all impostor patterns are falsely accepted and zero if none of the impostor patterns is accepted. The figure to the right of Figure 3.14 give the values of FAR for the score distribution with varying threshold. Similar to the impostor scores is

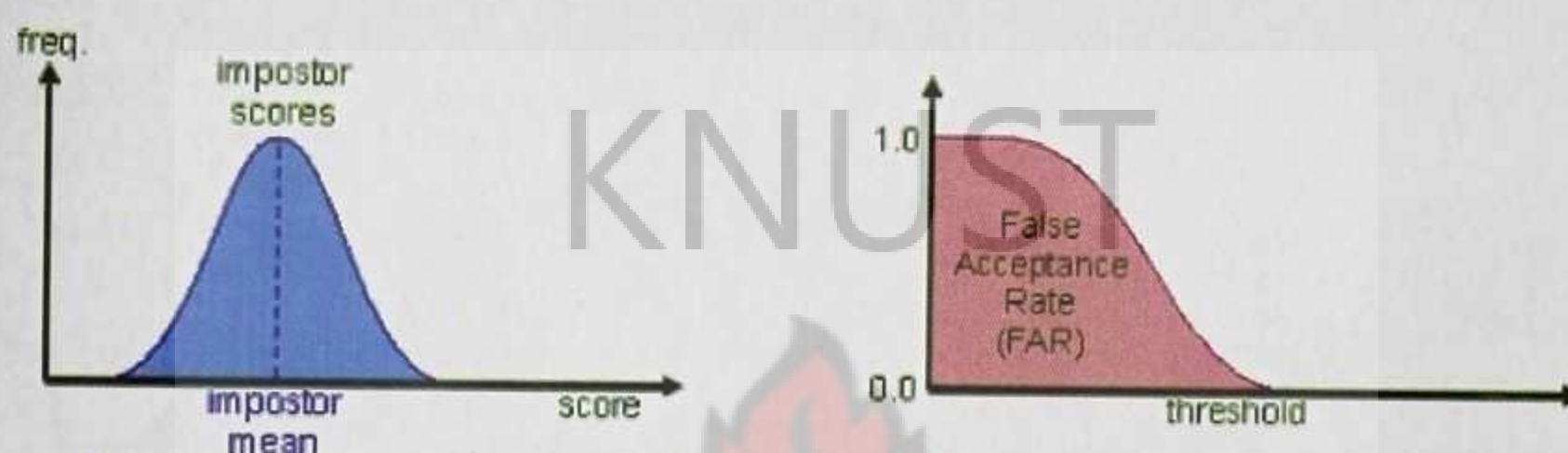


Figure 3.14: Determination of False Acceptance Rate (Impostor Score)

the genuine scores, where its mean score is higher than the mean value of the impostor patterns as shown in Figure 3.15. If the classification threshold applied to the classification scores is too high, some of the genuine patterns will be falsely rejected. The fraction of the number of rejected genuine patterns divided by the total number of genuine patterns is called the False Rejection Rate (FRR) and its value lies in between zero and one. Again the figure to the right of Figure 3.15 gives the FRR for the score distribution with varying threshold. The choice of the threshold value then becomes a problem if the distributions of the genuine and the impostor scores overlap, as can be found in Figure 3.16. When the score distributions overlap, the FAR and FRR intersect at a certain point with a value equal for both of them and this is called the Equal Error Rate (EER). Comparison of two biometric systems using the FAR and FRR is not sufficient to judge its betterment. Since the FAR and FRR are threshold dependent, a change in its value may cause undesirable result



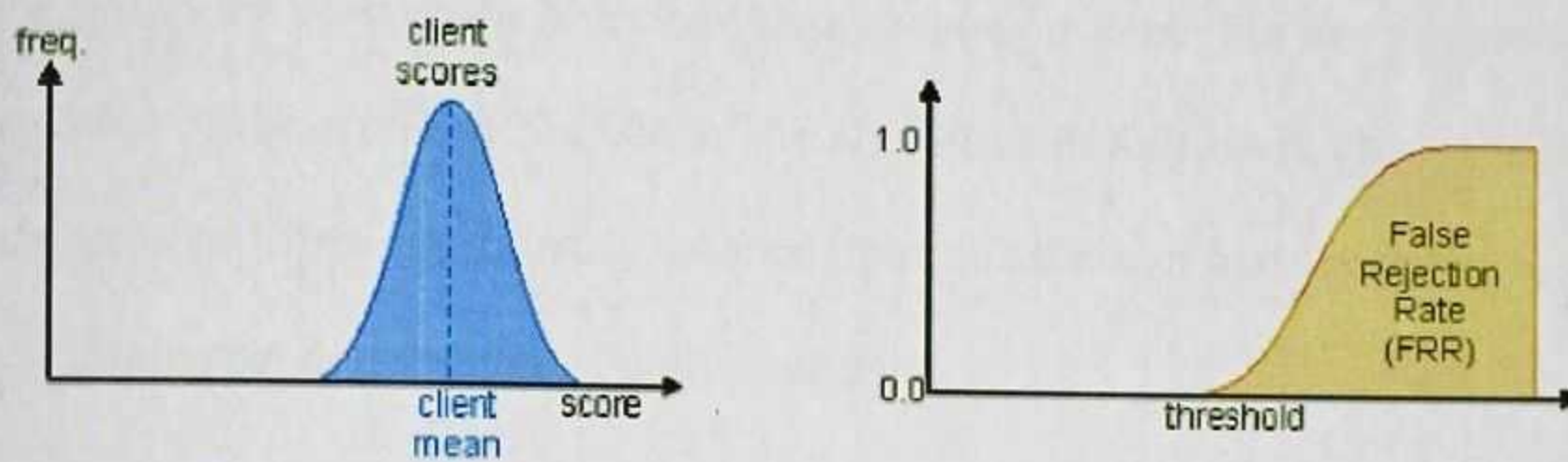


Figure 3.15: Determination of False Rejection Rate (Genuine Score)

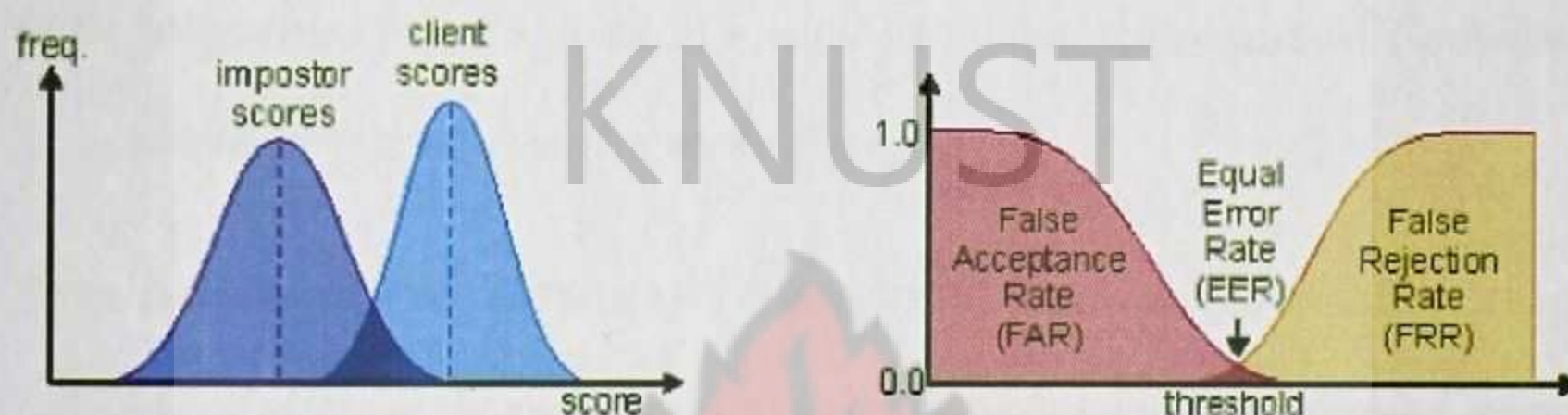


Figure 3.16: Estimation of Equal Error Rate (EER) value

and there will be no reasonable way to conclude if a system with a higher FAR and a lower FRR performs better than a system with a lower FAR and a higher FRR value. Hence the need for EER which is threshold independent performance measure. The smaller the EER is, the better is the system's performance. This in theory works fine, if the EER of the system is calculated using an infinite and representative test set, which of course is not possible under real world conditions. To get comparable results it is therefore necessary that the EERs that are compared are calculated on the same test data using the same test protocol.

### 3.6.2 Matching Approach

In fingerprint matching, the final feature vectors of the test fingerprints are compared with the final features of the database (source) fingerprints to verify whether a person should be accepted or rejected using matching techniques such as Euclidean



Distance, Support Vector Machines and many among others. For the purpose of this work we restrict the study to Euclidean Distance. In this approach, three consecutive steps are involved thus Euclidean Distance determination of feature vectors, sum of matched dominant features and finally matching.

1. **EUCLIDEAN DISTANCE DETERMINATION:** This stage is where the two fingers features are stored to a single matrix and the feature point of first row matched with associated feature point of second row using the standard Euclidean function where in our case is the second norm.
2. **SUM OF MATCHED DOMINANT FEATURES:** Here the Dominant features are extracted features obtained by concatenation of the local dominant orientation, center area and the canny edge parameters after apply the Discrete Wavelet Transform on the fingerprint image. For two fingers dominant features to be matched, select any one feature from each finger dominant features, and calculate the similarity of the two features associated with the two referenced feature points. If the similarity is larger than a threshold, assign value one to that feature and finally get the sum of that features as one value.
3. **MATCHING STEP:** This stage uses the elastic match algorithm to decide whether the two fingers are matched which is based on Euclidean distance and Sum of matched dominant features.



## CHAPTER IV

# ANALYSIS OF FEATURE EXTRACTION AND RECOGNITION

KNUST

### 4.1 Introduction

This chapter discusses the data that was used and how the extraction was done as well as the performance of the proposed method. It also explains the process involved and a vivid description of how some parameters are evaluated. Subsequent sections takes care of the detailed analysis to achieving the objectives of the study.

### 4.2 Source of Data

There are a number of dataset available that captures difficult scenarios that makes fingerprint recognition harder. Among which are:

1. National Institute of Standards and Technology (NIST)
2. Fingerprint Verification Competition 2000 (FVC2000)
3. Fingerprint Verification Competition 2002 (FVC2002)
4. Institute of Automation Chinese Academy of Sciences (CASIA)
5. Fingerprint Verification Competition 2004 (FVC2004)



Among the listed, FVC2004 database which contains DB1\_A, DB2\_A, DB3\_A and DB4\_A datasets but only DB3\_A dataset was used for the thesis. Thirty volunteers with average age of twenty four was selected at random. This dataset (DB3\_A) from FVC2004 database was established by capturing the fore and the middle fingers of both the hands (four fingers in total) of each volunteer present and this was done at three sessions where no efforts were made to control the image quality and the sensor plates not systematically cleaned. At each section, four impressions were acquired of each of the four fingers of each volunteer.

1. during the first sessions, individuals were asked to put the finger at a slightly different vertical position (in impressions 1 and 2) and to alternate low and high pressure against the sensor surface (impressions 3 and 4);
2. during the second session, individuals were requested to exaggerate skin distortion (impressions 1 and 2) and rotation (impressions 3 and 4) of the finger;
3. during the third session, fingers were dried (impressions 1 and 2) and moistened (impressions 3 and 4).

At the end of the data collection, a total of 120 fingers and 12 impressions per finger (1440 impressions) were gathered. Eight impressions out of the twelve impressions and 110 fingers out of the 120 fingers was selected at random making 880 fingerprint in all. However, fingers from 1 to 100 with the 8 impressions was kept in one set and named DB3\_A while the fingers from 101 to 110 was kept in another set and named as DB3\_B which was made available for fingerprint competition for parameter tuning of algorithms. Figure 4.1 shows a display of the 8 fingerprint impressions of the 100th volunteer.





Figure 4.1: The eight impressions of the 100th volunteer showing different variance.

### 4.3 Fingerprint Feature Analysis

With the chosen dataset (DB3\_A), a MATLAB (Matrix Laboratory) code was written using MATLAB2012b-Linux application running on Corbet Pack Machine with the following specification

1. 1.6 GHz Intel Dual Core (Processor)
2. 2GB of RAM (Memory)
3. 320 GB (Hard Disk)

having Ubuntu12.04 installed on it as the Operating System (OS). This MATLAB code works by

1. first reading the fingerprint image to obtain its matrix representation and resizing it from a resolution of  $300 \times 480$  to a resolution of  $256 \times 256$ , since wavelet transform best work on images with dimension in the powers of two where in our case is  $2^8 = 256$  at both dimensions as in Table 4.1. Here, fingerprint 1.1.tif



Table 4.1: Matrix Representation of Fingerprint 1\_1.tif

0.2902	0.3137	0.3137	0.3137	0.3333	...	...	0.2902	0.2706	0.2902	0.3137	0.2902
0.2706	0.3137	0.3333	0.3333	0.3137	...	...	0.2902	0.2902	0.3137	0.2706	0.2902
0.2706	0.3137	0.3137	0.3333	0.3333	...	...	0.2902	0.2706	0.3137	0.3137	0.2902
0.2902	0.3137	0.3529	0.3137	0.3137	...	...	0.3137	0.2902	0.2902	0.2902	0.2902
0.3137	0.2902	0.3333	0.3137	0.3333	...	...	0.3333	0.2902	0.3333	0.2902	0.2902
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
0.2510	0.3137	0.3137	0.3529	0.3137	...	...	0.4157	0.2706	0.1882	0.1647	0.1882
0.2902	0.2706	0.3137	0.3137	0.3137	...	...	0.5020	0.3765	0.2510	0.1020	0.1882
0.2706	0.2902	0.3137	0.3333	0.2902	...	...	0.4784	0.4392	0.3333	0.2078	0.1882
0.2510	0.2902	0.3137	0.3333	0.3137	...	...	0.4392	0.4588	0.4392	0.2902	0.2902
0.2510	0.3333	0.3137	0.3137	0.2706	...	...	0.4157	0.4392	0.4588	0.3529	0.2902

will be used to illustrate all the process involved in the data analysis.

2. A series of test was performed by subjecting the matrix representation of the fingerprint image to the various wavelet transforms available as show in Table 4.2 with Haar Wavelet inclusive thus fifty two in all at three levels of multi-resolution which decomposed the matrix in to four sub matrices. This contains the directional characteristics of the matrix, thus, features from the horizontal details, vertical details and diagonal details along with the approximation coefficient. Figure 4.2 shows these directional characteristics in the four directions aforementioned at level one resolution using fingerprint image 1\_1.tif. The application of these family wavelet transforms in image processing provide a refined ground to directional feature extraction however these features extracted does not represent the full directional information. Consequently, these can not be used directly as a feature vector to represent a given fingerprint uniquely hence the need for a feature vector which fully describes in detail the directional information of the fingerprint image.



Table 4.2: List of all the Wavelet Transform Families Used.

S/N	Bi-orthogonal Wavelet Family	Reverse Bi-orthogonal Wavelet Family	Daubechies	Coiflets	Symlets
1	bior1.1	rbior1.1	db1	coif1	sym2
2	bior1.3	rbior1.3	db2	coif2	sym3
3	bior1.5	rbior1.5	db3	coif3	sym4
4	bior2.2	rbior2.2	db4	coif4	sym5
5	bior2.4	rbior2.4	db5	coif5	sym6
6	bior2.6	rbior2.6	db6		sym7
7	bior2.8	rbior2.8	db7		sym8
8	bior3.1	rbior3.1	db8		
9	bior3.3	rbior3.3	db9		
10	bior3.5	rbior3.5	db10		
11	bior3.7	rbior3.7			
12	bior3.9	rbior3.9			
13	bior4.4	rbior4.4			
14	bior5.5	rbior5.5			
15	bior6.8	rbior6.8			



Figure 4.2: Directional characteristics of the wavelet transform at level one of resolution of fingerprint 1.1.tif Row one is the Approximation and Diagonal Details whiles Row is the Horizontal and Vertical Details



3. To address this problem at hand, the Gaussian gradient in the  $x$  and  $y$  direction thus  $G_{mn}^x$  and  $G_{mn}^y$  as shown in Finger 4.3 respectively are used to produce the angle (orientation) of the features thus  $\theta_{mn}$  as in Finger 4.4.



Figure 4.3: Gradient in the direction of  $x$  and  $y$  respectively.



Figure 4.4: Orientation of the fingerprint after extracting the Gradient in the  $x$  and  $y$  direction

4. The magnitude of the gradient ( $G_{mn}$ ) along with the coherence ( $\delta_{mn}$ ) as in Figure 4.5 and Figure 4.4 respectively, are applied to the wavelet directional features extracted to generate the local dominant orientation ( $\theta$ ) as formulated in Section 3.5 .

These matrix images, thus the coherence and the local dominant orientation, contains some vital information which can be used to represent a fingerprint uniquely and these features are:

- (a) correlation



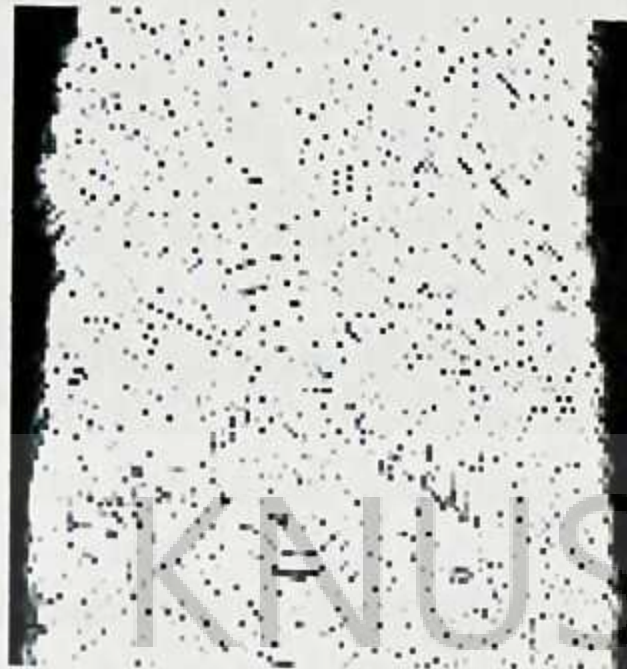


Figure 4.5: Magnitude of Gradient of the fingerprint



Figure 4.6: Coherence image of the fingerprint after applying reverse bi-orthogonal wavelet 3.1



(b) energy

(c) homogeneity and

(d) contrast

as defined in Equation 3.5.52 , Equation 3.5.53, Equation 3.5.55 and Equation 3.5.54 respectively. Table 4.3 gives a display of these unique features extracted from the coherence and the local dominant orientation matrix of the approximation coefficient image with fingerprint 1\_1.tif as an example at one level resolution using the reverse bi-orthogonal wavelet 3.1 whiles Table 4.4 gives that of reverse bi-orthogonal wavelet 3.9 and finally, Table 4.5 represent the unique features using reverse bi-orthogonal wavelet 4.4. These wavelet mentioned was not selected at random but was chosen according to performance. They prove to be good by providing significant result among the many family wavelet considered which will be further explained in Section 4.4.

Table 4.3: Unique feature vector from local dominant orientation and coherence matrix using reverse bi-orthogonal wavelet 3.1

Contrast				Correlation				Homogeneity				Energy			
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0.7707	0.7784	0.7796	0.7641	0.0306	0.0170	0.0205	0.0349	0.7214	0.7148	0.7230	0.7201	0.2062	0.2040	0.2075	0.2047
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0.8124	0.8329	0.8214	0.8196	0.0258	0.0048	0.0199	0.0106	0.7127	0.7063	0.7079	0.7108	0.1953	0.1950	0.1957	0.1939
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0.9127	0.8595	0.9679	0.9596	0.0665	0.1016	0.0262	-0.0051	0.7120	0.7073	0.7010	0.6955	0.1764	0.1741	0.1712	0.1752

Table 4.4: Unique feature vector from local dominant orientation and coherence matrix using reverse bi-orthogonal wavelet 3.9

Contrast				Correlation				Homogeneity				Energy			
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0.7751	0.7801	0.7838	0.7720	0.0379	0.0248	0.0257	0.0348	0.7210	0.7153	0.7223	0.7191	0.2043	0.2024	0.2054	0.2026
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0.8071	0.7920	0.8003	0.8041	0.0357	0.0437	0.0421	0.0283	0.7135	0.7166	0.7200	0.7123	0.1976	0.1978	0.2005	0.1961
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0.8634	0.9014	0.8620	0.9386	0.0492	-0.00431	0.0739	-0.0467	0.7028	0.7048	0.7138	0.6968	0.1883	0.1887	0.1853	0.1864

As could be seen from Table 4.3, the odd rows represent the features from the coherence matrix as the even once give the features corresponding to the local domi-



Table 4.5: Unique feature vector from local dominant orientation and coherence matrix using reverse bi-orthogonal wavelet 4.4

Contrast				Correlation				Homogeneity				Energy			
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0.7743	0.7803	0.7766	0.7725	0.0147	0.0044	0.0123	0.0139	0.7203	0.7164	0.7206	0.7160	0.2068	0.2049	0.2077	0.2044
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0.8319	0.8017	0.8244	0.8099	0.0053	0.0350	0.0138	0.0255	0.7145	0.7145	0.7143	0.7091	0.1970	0.1951	0.1963	0.1936
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0.9676	0.9093	0.9838	0.9633	0.0262	0.0552	-0.0109	0.0037	0.7034	0.7048	0.6961	0.6885	0.1756	0.1767	0.1773	0.1743

nant orientation matrix. For each feature say correlation, four values are generated by looking at the coefficient image (coherence and local dominant orientation) at different offsets (angles) thus  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$  and  $135^\circ$  of the Gray Level Co-occurrence matrix (GLCM). The same explains the content of Table 4.4 and Table 4.5. The Gray Level Co-occurrence Matrix (GLCM) as mentioned was evaluated by calculating how often a pixel with gray-level (grayscale intensity) value  $i$  occurred horizontally adjacent to a pixel with the value  $j$ .

In this work, GLCM at angles (offsets) of  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$  and  $135^\circ$  were taken into account in order to make the features to be extracted from the fingerprint image more invariant and robust to rotation. These features extracted from a particular fingerprint image form the feature vector, where in this illustration we chose again fingerprint 1\_1.tif as an example. The standard deviation and the threshold values was also extracted from the wavelet decomposed fingerprint image, which represent the edge parameters as shown in Table 4.6, 4.7 and 4.8 using the reverse bi-orthogonal wavelet 3.1, 3.9 and 4.4 respectively.

Table 4.6: Threshold and standard deviation of the edge detector using reverse bi-orthogonal wavelet 3.1

	Threshold Values		Standard Deviation
Approximation	0.1188	0.2969	0.5590
Horizontal	0.1125	0.2812	0.4836
Vertical	0.1188	0.2969	0.4996
Diagonal	0.1250	0.3125	0.4978



Table 4.7: Threshold and standard deviation of the edge detector using reverse bi-orthogonal wavelet 3.9

	Threshold Values		Standard Deviation
Approximation	0.1188	0.2969	0.5590
Horizontal	0.0375	0.0938	0.2322
Vertical	0.1000	0.2500	0.2697
Diagonal	0.0625	0.1562	0.2727

Table 4.8: Threshold and standard deviation of the edge detector using reverse bi-orthogonal wavelet 4.4

	Threshold Values		Standard Deviation
Approximation	0.1250	0.3125	0.7215
Horizontal	0.0438	0.1094	0.1464
Vertical	0.0813	0.2031	0.1763
Diagonal	0.0625	0.1562	0.1090

As usual, the columns, one and two of Table 4.6 represent the threshold values of the approximation coefficient, horizontal details, vertical details and the diagonal details of the decomposed fingerprint using reverse bi-orthogonal wavelet 3.1 while the last one is the standard deviation. The same holds for Table 4.7 and Table 4.8 using reverse bi-orthogonal wavelet 3.9 and 4.4 used respectively.

Finally, a region of  $16 \times 16$  sized image is cropped from the wavelet decomposed fingerprint image around the point at which the maximum variance is detected in both the row and column. Features as extracted from the coherence and dominant orientation is also extracted from this cropped image, constituting the final stage of the feature extraction. Below are the tabular representation of the features extracted, Table 4.9 represent features from reverse bi-orthogonal 3.1, Table 4.10 for reverse bi-orthogonal wavelet 3.9 and Table 4.11 for reverse bi-orthogonal wavelet 4.4 in that order.

For each of the 800 fingerprint image, the features extracted from level one of resolution are concatenated to form a row vector. The same is done at level two



Table 4.9: Feature vector representation of the center area using the reverse bi-orthogonal wavelet 3.1

	Contrast				Correlation				Homogeneity				Energy			
Approximation	1.8708	16.6222	13.1333	9.8267	0.8877	-0.0038	0.2117	0.4060	0.7717	0.4370	0.5000	0.5610	0.2202	0.0901	0.1060	0.1274
Horizontal	2.3750	17.0844	15.9333	12.9644	0.8307	-0.2667	-0.1810	0.0396	0.7923	0.5135	0.5321	0.5602	0.3307	0.1743	0.1865	0.1992
Vertical	15.8875	18.7689	7.4083	6.2400	-0.1988	-0.4344	0.4224	0.5247	0.5296	0.4133	0.6456	0.6898	0.1718	0.1093	0.2279	0.2616
Diagonal	13.2375	6.1511	11.0583	12.3822	-0.0705	0.5087	0.1031	0.0048	0.5632	0.7149	0.6298	0.5618	0.1870	0.2874	0.2540	0.1826

Table 4.10: Feature vector representation of the center area using the reverse bi-orthogonal wavelet 3.9

	Contrast				Correlation				Homogeneity				Energy			
Approximation	1.8542	15.9644	12.8667	9.9289	0.8905	0.0702	0.2534	0.4191	0.7745	0.4563	0.5090	0.5595	0.2192	0.0936	0.1065	0.1253
Horizontal	2.4208	9.8844	9.9375	9.0311	0.6920	-0.2575	-0.2686	-0.1446	0.7622	0.5622	0.5743	0.5946	0.3269	0.1714	0.1768	0.1966
Vertical	6.0958	5.9867	2.9333	4.2356	-0.2459	-0.2438	0.3650	0.1200	0.6338	0.6324	0.7566	0.7185	0.2424	0.2435	0.3547	0.3175
Diagonal	7.4292	4.1111	8.2042	4.1911	-0.2660	0.3044	-0.3875	0.2907	0.5261	0.6718	0.4905	0.6542	0.1341	0.2207	0.1199	0.2104

Table 4.11: Feature vector representation of the center area using the reverse bi-orthogonal wavelet 4.4

	Contrast				Correlation				Homogeneity				Energy			
Approximation	3.6375	20.8489	17.5333	13.8178	0.8122	-0.0634	0.1052	0.2936	0.7885	0.4681	0.5271	0.5863	0.2609	0.1285	0.1380	0.1523
Horizontal	1.2833	2.8578	3.0250	2.9156	0.4589	-0.1736	-0.2483	-0.2058	0.8105	0.6770	0.6649	0.6756	0.4057	0.2827	0.2770	0.2762
Vertical	1.7583	1.6889	1.5667	1.9022	-0.0769	-0.0259	0.0356	-0.1555	0.7244	0.7257	0.7433	0.7225	0.3155	0.3192	0.3391	0.3184
Diagonal	1.2000	1.0756	1.2917	0.5956	-0.1249	0.0371	-0.2016	0.4670	0.7712	0.7904	0.7576	0.8626	0.3928	0.4035	0.3785	0.4923

and level three of resolutions. All these three feature vectors represent a particular fingerprint image uniquely which is stored and used for recognition purpose.

Any fingerprint recognition system typically returns a matching score on the basis of similarity between the input (test) and the database (template) image which is based on the Euclidean distance. The smaller the value is the more certain it is that the two fingerprint images come from the same individual. A fingerprint recognition system have two types of errors thus:

1. Two different fingerprint image seen as the same (called false match or false acceptance) and
2. Two same fingerprint image seen as different (called false non-match or false rejection).

There are two ways for testing the system performance which are mostly used as per the FVC2004 Protocol. Firstly, total genuine attempts are measured by comparing each fingerprint image to its remaining versions and the second method is measuring



total impostor attempts which are measured by comparing the first impression of each fingerprint image to the first image of all remaining impression of different fingerprint leading to a total of 2800 genuine attempts and 4950 impostor attempts respectively. The ratio of the number of false match to the impostor attempts is called False Acceptance Rate (FAR) while the number of false non-match to the genuine attempts is called the False Rejection Rate (FRR). Both FAR and FRR are dependent on each other and there is a trade-off between them (Maio and Maltoni, 1997) which measures the Equal Error Rate (EER). However, this EER can be estimated from the plot of False Reject Rate (FRR) and False Acceptance Rate (FAR) against all possible values of thresholds. The smaller the value of EER is as compared to other recognition system, the more efficient recognition system will be.

#### 4.4 Analysis of Fingerprint Recognition

The expectation of an increase in recognition rate as well as best choice of mother wavelet using %FAR and %FRR as performance indicators is highly dependent on the nature of the feature vectors extracted and how they are used for recognition. Tables 4.3 - 4.11 as discussed above gives the summary of all the needed features extracted, however, the organization of these features was not discussed. This organization is very important since they contain the uniqueness of a particular fingerprint image and also guide us on how the threshold should be set to make the recognition system more robust.

In this work, the organization of the total extracted features was categorized into three main groups base on the level of resolution. Hence, each level contains exactly 108 feature values after feature concatenation and since a three level of resolution was applied to the fingerprint image, a total of  $3 \times 108 = 324$  was expected for each



fingerprint impression. Using Figure 4.7 - 4.9 as a case study, it could be noted that each figure has in all three graphs marked as red, blue and black.

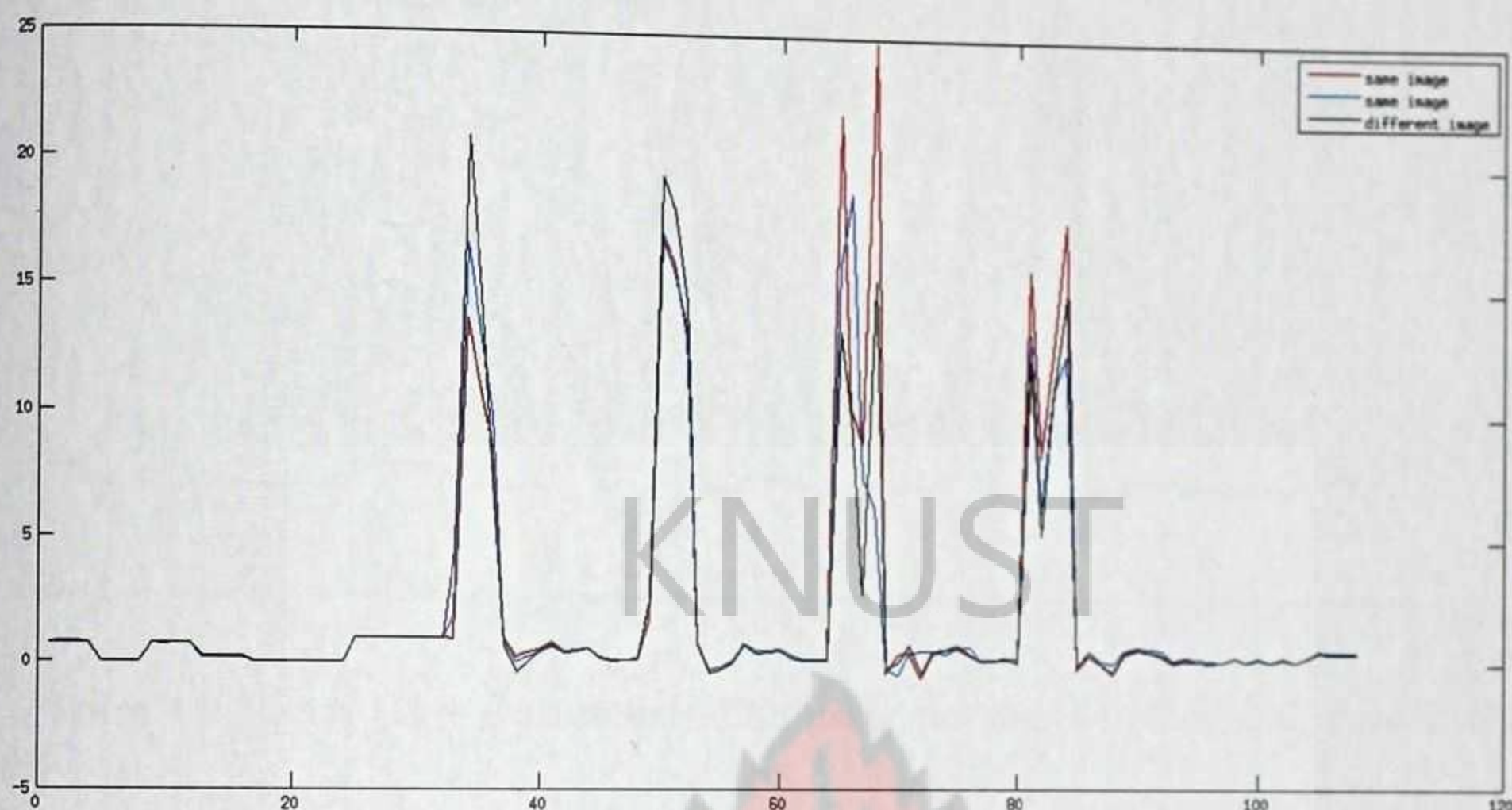


Figure 4.7: Level one feature extraction using reverse bi-orthogonal wavelet 3.1

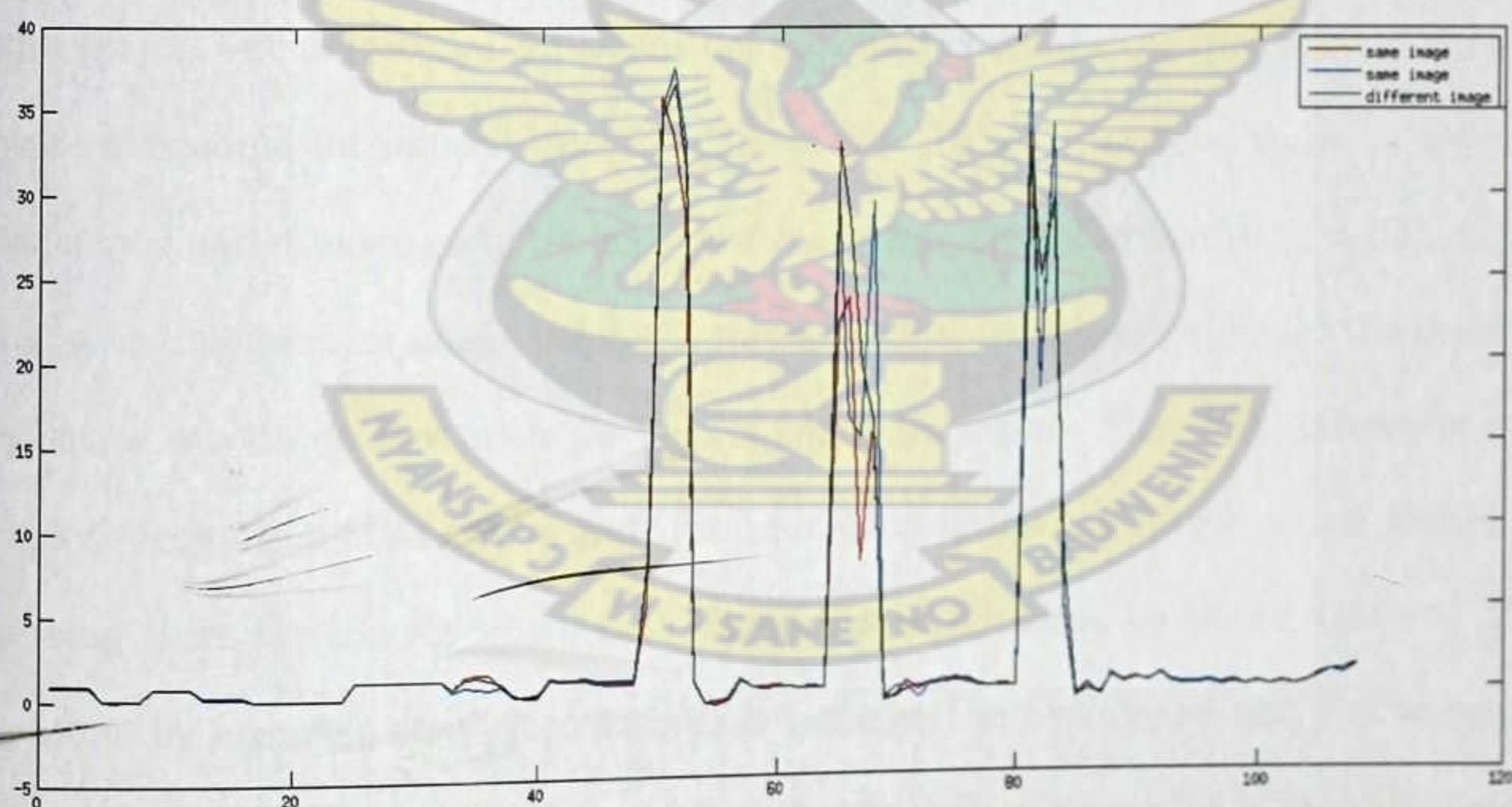


Figure 4.8: Level two feature extraction using reverse bi-orthogonal wavelet 3.1

The red and blue are representation a plot of the feature vector from the same fingerprint image where in this example, finger 1.1.tif and 1.2.tif was considered. The black one is a different fingerprint image thus finger 2.1.tif not belonging to the same



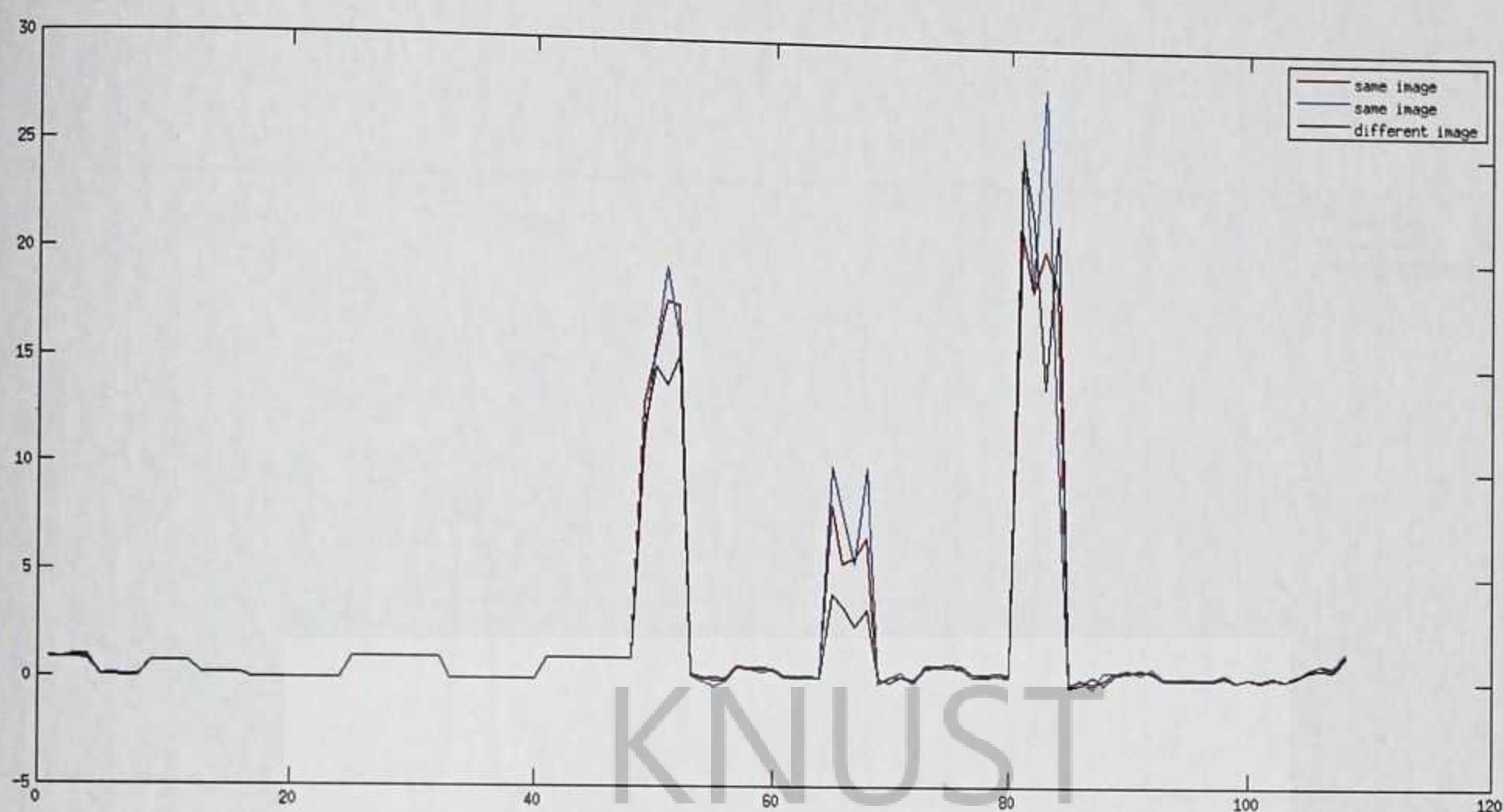


Figure 4.9: Level three feature extration using reverse bi-orthogonal wavelet 3.1

class of image. It is expected that, graph of the same image (i.e red and blue) should coincide with each whiles the different image (i.e black) should disassociate itself from the graph, but unfortunately, this is not so which is due to the inconsistency which was intentional introduced into the dataset (i.e DB3\_A) provided so as to test the efficiency and robustness of the recognition system. From the figures, it is difficult to tell which fingerprint image belong to the same class with the eye, hence the need for a better way to do that with the help of computer vision. Since the dataset is very noisy, recognition of an authorized user from an unauthorized user is accomplished by setting three criteria for which each fingerprint image must be tested against. This is done by ensuring that, the differences between the test image and the template image at level one, two and three against the predefined threshold is satisfied before accepting it as an authorized user otherwise is tag as an unauthorized user. The same is true when using the reverse bi-orthogonal wavelet 3.9 and 4.4 as shown in Figure 4.10 - 4.12 and Figure 4.13 - 4.15 respectively.

The performance parameters such as FAR, FRR and TSR for the various mother



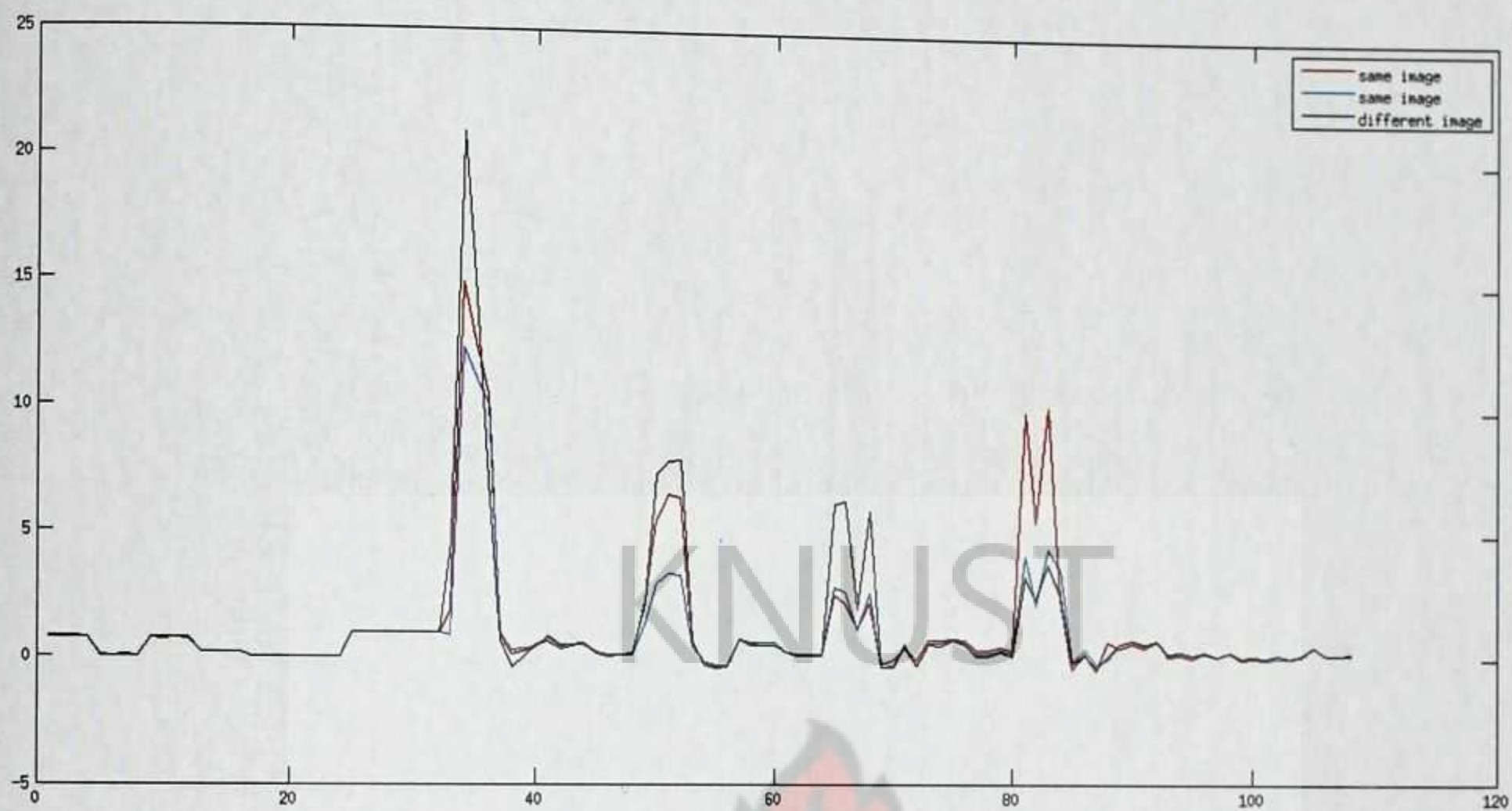


Figure 4.10: Level one feature extration using reverse bi-orthogonal wavelet 3.9

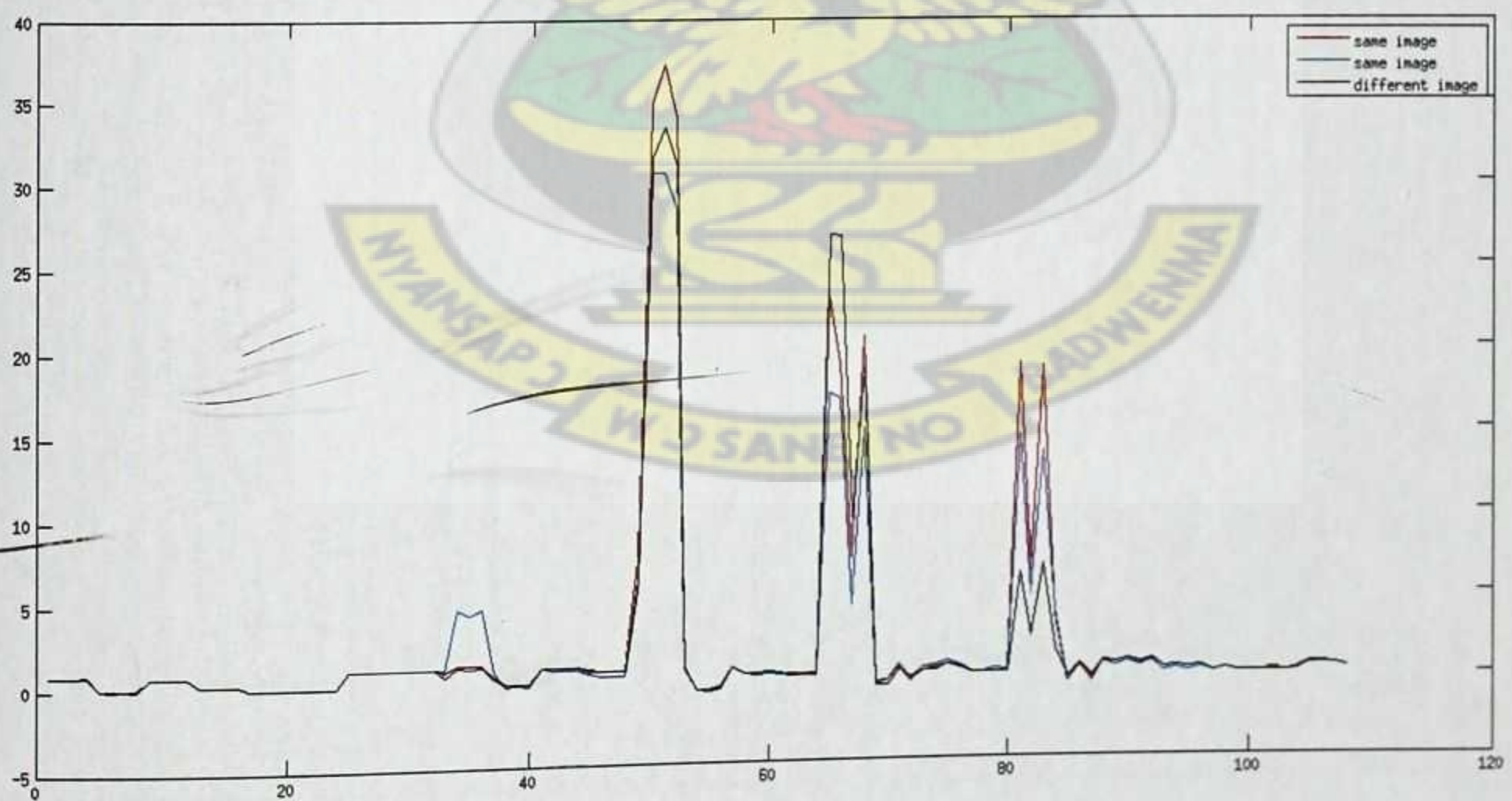


Figure 4.11: Level two feature extration using reverse bi-orthogonal wavelet 3.9



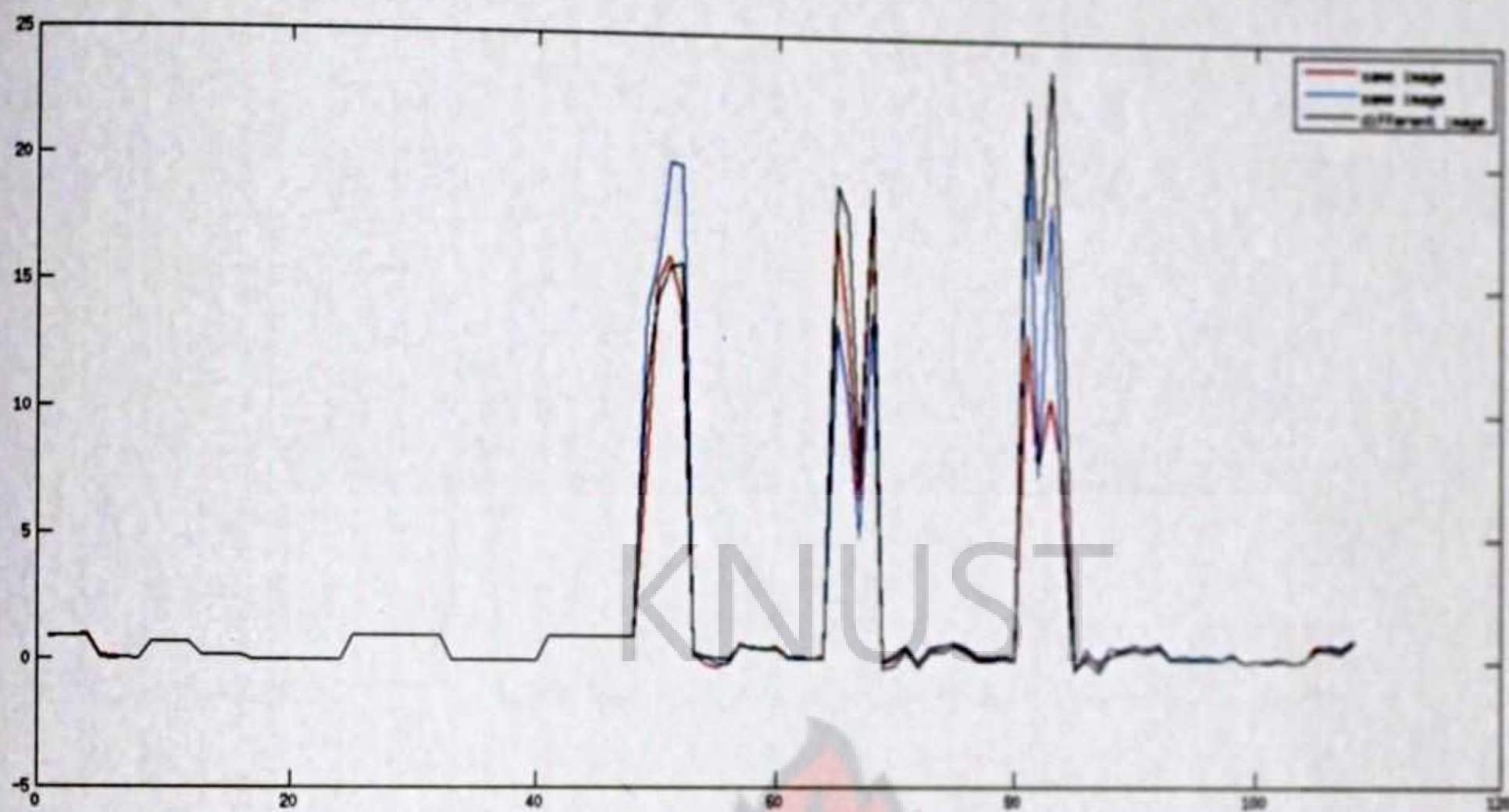


Figure 4.12: Level three feature extraction using reverse bi-orthogonal wavelet 3.9

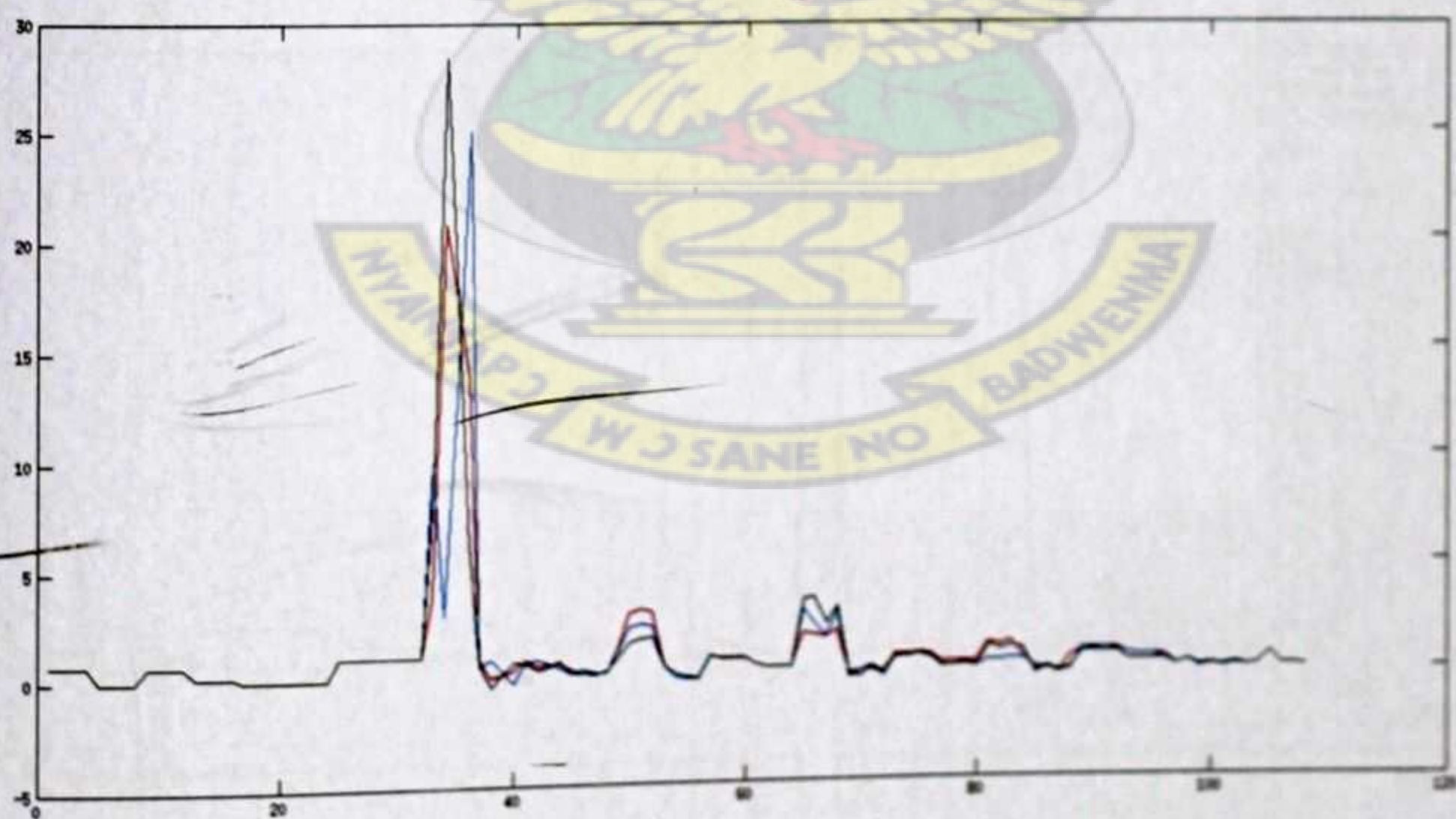


Figure 4.13: Level one feature extraction using reverse bi-orthogonal wavelet 4.4



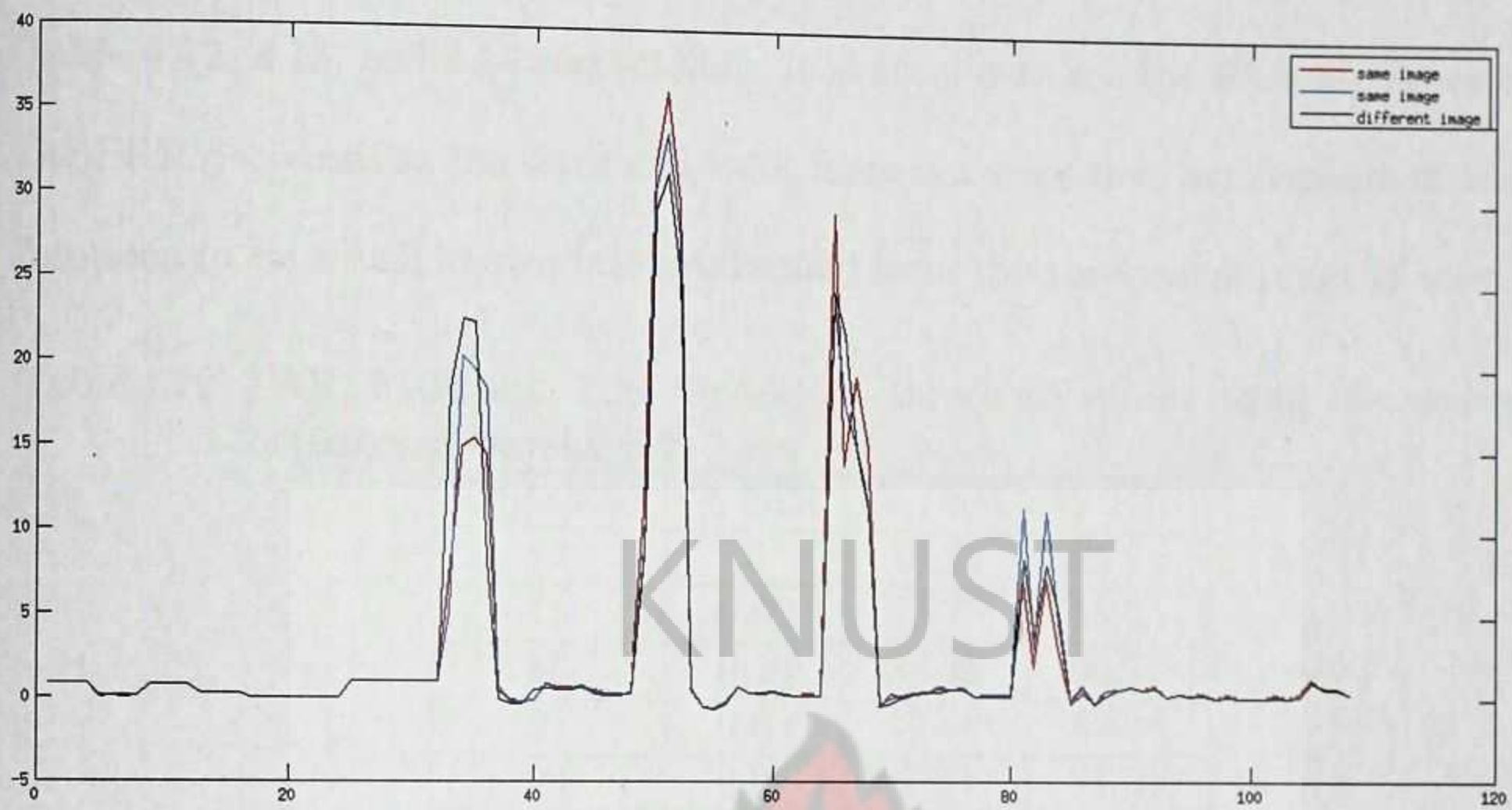


Figure 4.14: Level two feature extration using reverse bi-orthogonal wavelet 4.4

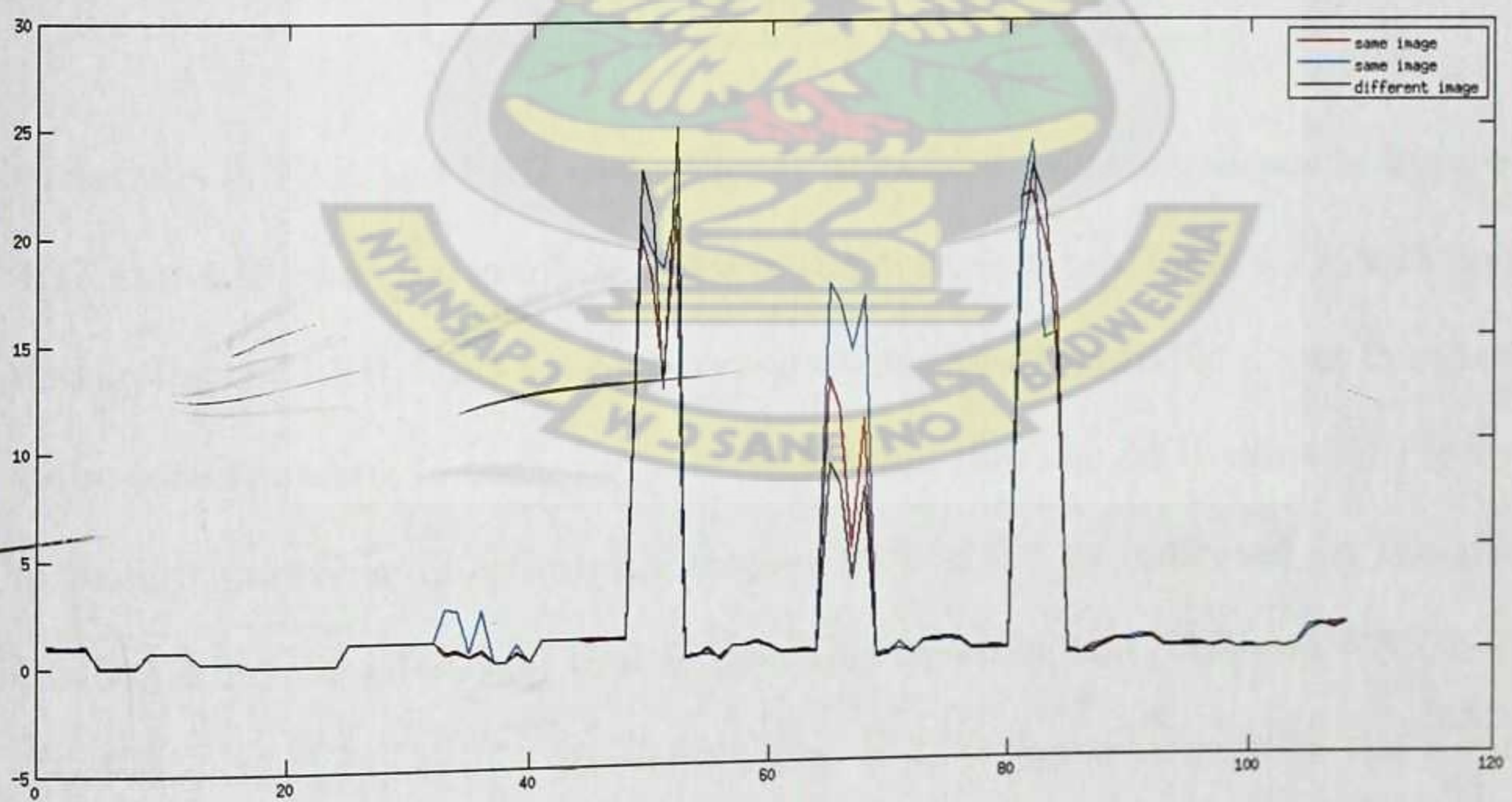


Figure 4.15: Level three feature extration using reverse bi-orthogonal wavelet 4.4



wavelets employed based on the extracted feature vectors are computed as shown in Table 4.12, 4.13, and 4.14 respectively. It is observed that the FAR increases whiles the FRR decreases as the threshold value increases since they are dependent and this happens to be a well known fact established from the theoretical point of view. The

Table 4.12: FAR, FRR and TSR for various threshold values using the reverse bi-orthogonal wavelet 3.1

Threshold	% FAR	% FRR	% TSR
12	0	76.25	23.75
15	0.02	53.75	46.25
18	0.10	33.21	66.79
21	0.63	17.64	82.36
24	2.06	9.00	91.00
27	5.13	4.64	95.36
30	11.68	2.32	97.68
33	21.47	0.86	99.14
36	33.47	0.39	99.61
39	47.07	0.11	99.89
42	61.66	0.04	99.96
45	73.66	0	100
48	82.48	0	100

variations of FAR and FRR with different threshold values as shown in Figure 4.16, 4.17 and 4.18 gives the graphical view of the data found in Table 4.12, 4.13 and 4.14 displaying the EER value as value corresponding to the variable  $y$  and the threshold value corresponding to the  $x$  variable. It is found that the EER value for Figure 4.16 in using the reverse bi-orthogonal mother wavelet 3.1 as indicated on the graph is 0.04643 which is 4.64% and that is the value at which the FAR and FRR becomes equal and this happen at a threshold value of 27 as represented with the  $x$ -variable shown on the graph. This mother wavelet used to generate the value of EER as per Table 4.12 yields a total success rate (TSR) of 95.36%.

In using the reverse bi-orthogonal mother wavelet 3.9, an EER value of 0.0575 thus 5.75% is noted as found on Figure 4.17 and this occurred at a threshold value



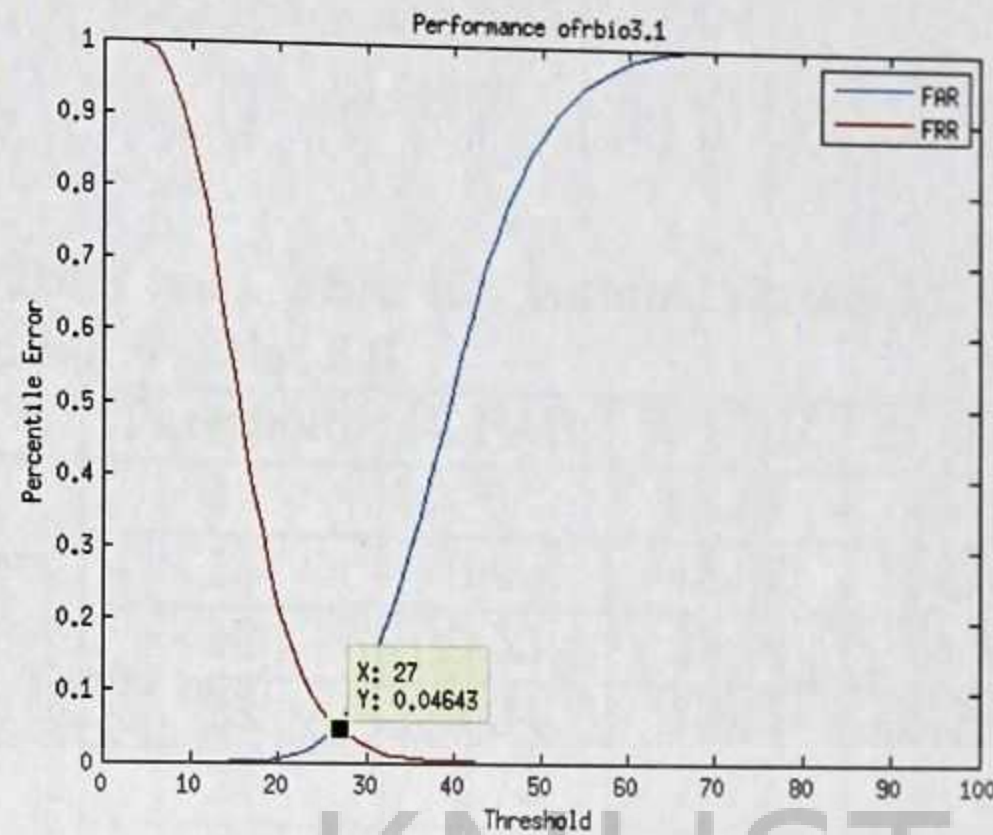


Figure 4.16: Graph of variation of FAR and FRR using rbio3.1 wavelet.

of 24 giving the total success rate of 94.25% as could be seen from Table 4.13

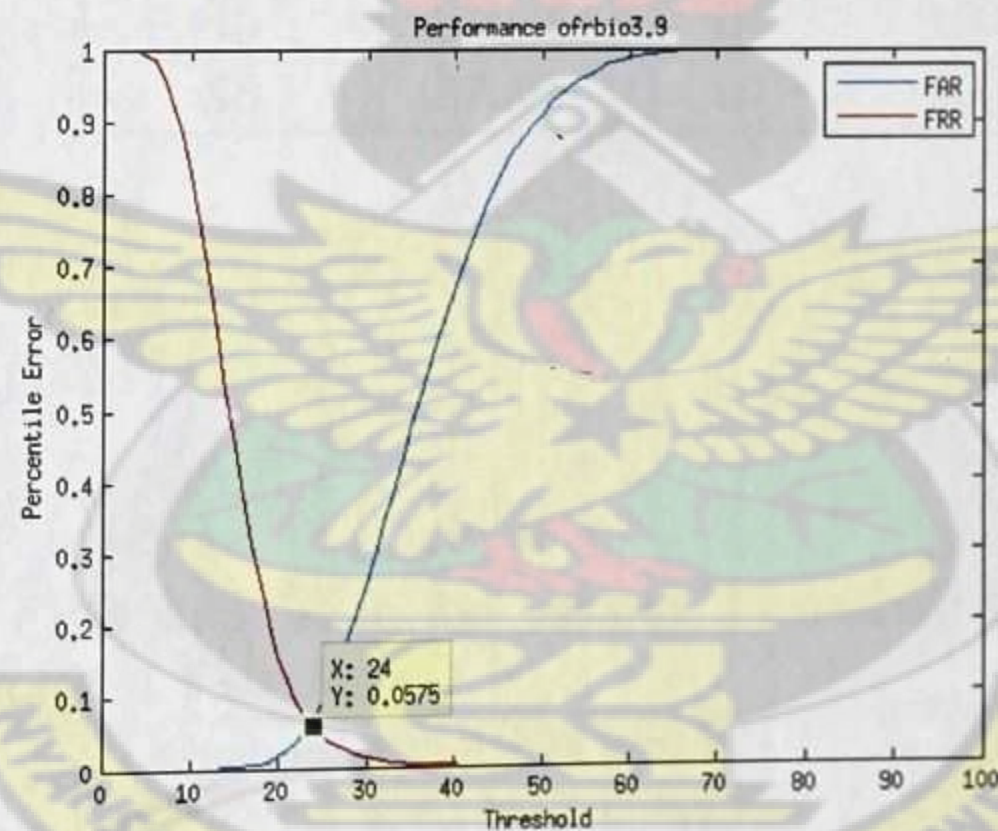


Figure 4.17: Graph of variation of FAR and FRR using reverse bi-orthogonal 3.9 wavelet.

Again, in the application of the reverse bi-orthogonal mother wavelet 4.4 in this experiment, an EER value of 0.05464 thus 5.46% was achieved at a threshold value of 26 as depicted in Figure 4.18. A total success rate from Table 4.14 reveals a value of 94.54%. From the analysis above, it is seen that the mother wavelet named reverse bi-orthogonal wavelet 3.1 which in short is rbio3.1 produced a better recognition rate



Table 4.13: FAR, FRR and TSR for various threshold values using reverse bi-orthogonal wavelet 3.9

Threshold	% FAR	% FRR	% TSR
12	0	68.61	31.39
15	0.02	45.07	54.93
18	0.53	25.54	74.46
21	2.34	12.11	87.89
24	6.73	5.75	94.25
27	14.81	2.64	97.36
30	25.47	1.32	98.68
33	38.08	0.54	99.32
36	51.23	0.18	99.82
39	62.95	0.04	99.96
42	73.13	0	100
45	81.92	0	100
48	87.86	0	100

Table 4.14: FAR, FRR and TSR for various threshold values using reverse bi-orthogonal 4.4 wavelet.

Threshold	% FAR	% FRR	% TSR
12	0	62.68	37.32
15	0	44.64	55.36
18	0.20	28.54	71.46
21	1.05	16.04	83.96
24	3.11	8.71	91.29
26	5.58	5.46	94.54
30	16.46	1.96	98.04
33	27.76	1.00	99.00
36	42.24	0.46	99.54
39	56.46	0.18	99.82
42	69.92	0.11	99.89
45	79.35	0.04	99.96
48	86.48	0	100



of 95.36% as compared to the other wavelet functions used. By this proceeds, we are prone to accepting 5.14% of the impostors whiles rejecting 4.64% of the genuine candidate so as to commit a minimal error of 4.64% which is the trade-off (EER). Table 4.15 - 4.21 gives a summary of all the wavelet families used in this thesis sorted in ascending order of performance. From the summary, mother wavelet Reverse Bi-Orthogonal 3.1, Reverse Bi-Orthogonal 4.4 and Reverse Bi-Orthogonal 4.4 gives a better result on performance, hence, for convenience sake we explained only such ones fully in this thesis however these process applies to the rest of the wavelet families not discussed here.

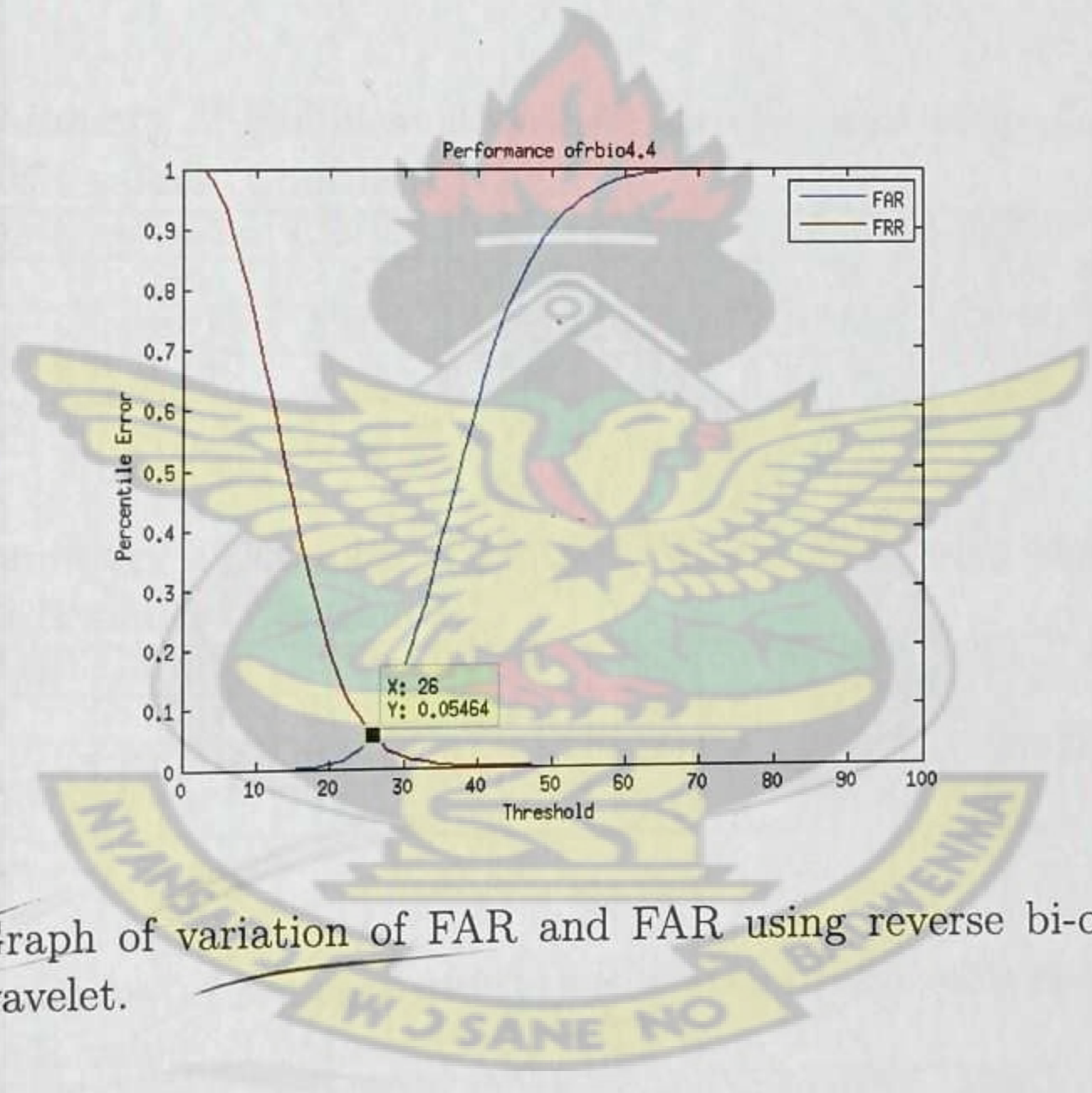


Figure 4.18: Graph of variation of FAR and FAR using reverse bi-orthogonal 4.4 wavelet.

Table 4.15: Summary of Families of Mother Wavelet used with their corresponding EER values

WAVELET NAME	RBIO3.1	RBIO4.4	RBIO3.9	BIOR3.1	SYM4	RBIO2.4	RBIO2.2	DB6
THRESHOLD	27	26	24	21	24.56	22.38	23.34	23.88
EER	0.0463	0.05464	0.0575	0.05812	0.06104	0.06187	0.06222	0.06442
%eEER	4.64	5.46	5.75	5.81	6.10	6.19	6.22	6.44



Table 4.16: Summary of Families of Mother Wavelet used with their corresponding EER values (Continued)

WAVELET NAME	BIOR3.5	RBIO3.5	DB5	BIOR2.6	BIOR5.5	BIOR3.9	DB9	RBIO2.6
THRESHOLD	25.18	23	24	25.81	19.19	25.38	23.75	24.04
EER	0.0646	0.065	0.06536	0.06595	0.066	0.06605	0.06631	0.06656
%EER	6.46	6.50	6.54	6.60	6.60	6.61	6.63	6.66

Table 4.17: Summary of Families of Mother Wavelet used with their corresponding EER values (Continued)

WAVELET NAME	BIOR3.3	RBIO3.7	DB10	COIF1	RBIO2.8	SYM7	COIF5	BIOR3.7
THRESHOLD	23.87	24	24.66	24	22.47	23	25.05	24.14
EER	0.06706	0.06714	0.06744	0.06893	0.06994	0.07047	0.07111	0.07163
%EER	6.71	6.71	6.74	6.89	6.99	7.05	7.11	7.16

Table 4.18: Summary of Families of Mother Wavelet used with their corresponding EER values (Continued)

WAVELET NAME	BIOR2.4	RBIO6.8	COIF4	DB2	RBIO5.5	COIF2	BIOR1.3	SYM5
THRESHOLD	25.25	25	24.72	23.83	26	23.9	24.42	23.34
EER	0.07181	0.0725	0.07261	0.07285	0.07286	0.07324	0.07331	0.07285
%EER	7.18	7.25	7.26	7.29	7.29	7.32	7.33	7.41

Table 4.19: Summary of Families of Mother Wavelet used with their corresponding EER values (Continued)

WAVELET NAME	BIOR2.2	SYM2	BIOR1.5	DB7	SYM6	DB4	RBIO1.3	BIOR2.8
THRESHOLD	25.14	23.73	24.81	23.78	24	23.44	22.39	24.91
EER	0.07446	0.0751	0.07541	0.07543	0.07561	0.07607	0.07677	0.07688
%EER	7.45	7.51	7.54	7.54	7.56	7.61	7.68	7.69

Table 4.20: Summary of Families of Mother Wavelet used with their corresponding EER values (Continued)

WAVELET NAME	RBIO1.1	SYM3	SYM8	DB3	COIF3	DB8	HAAR	RBIO1.5
THRESHOLD	23.2	24.34	28	23.34	23.61	23.36	23.11	22.16
EER	0.07778	0.07785	0.07786	0.07789	0.07874	0.07923	0.07966	0.08045
%EER	7.78	7.79	7.79	7.79	7.87	7.92	7.97	8.05

Table 4.21: Summary of Families of Mother Wavelet used with their corresponding EER values (Continued)

WAVELET NAME	DB1	BIOR4.4	BIOR6.8	BIOR1.1	RBIO3.3
THRESHOLD	23	22.07	23.62	22.93	24
EER	0.08214	0.08222	0.08333	0.08338	0.08357
%EER	8.12	8.22	8.33	8.34	8.36



## 4.5 Analysis of Choice of Threshold

Thresholding as it stands in fingerprint recognition plays an important role and this is a predefined value used in determining whether a particular fingerprint (test data) matches any known fingerprint in the template database. This decision is based on calculating the Euclidean distance ( $norm_2$ ) between the test data and the template database. When the distance value is less than the set threshold then the test data is deemed to be found in the template database and this information in totality helps in evaluating the FAR and the FRR. Many are the choices of defining the threshold range to be used for performance analysis but the most commonly used ones are based on the dispersion of distance values, while evaluating the Euclidean distance by choosing the minimum and maximum distance value found. However, these values when greater than 100 can be normalized so they range from zero to hundred.

In this work, since the dispersion of distance remains in the domain of zero to hundred, we chose to use the full range as shown in Table 4.22. The point at which the %FAR equals the %FRR which determines our %ERR is captured that forms the basis in analyzing the performance of a fingerprint algorithm. This value must be

Table 4.22: Comparison between the three selected mother wavelets.

Wavelets Name	Threshold	% FAR	% FRR	% TSR	% ERR
rbio3.1	27	5.13	4.64	95.36	4.64
rbio3.9	24	6.73	5.75	94.25	5.75
rbio4.4	26	5.58	5.46	94.54	5.46

small as possible and from the table, the reverse bi-orthogonal mother wavelet 3.1 proved to be the best with a threshold value of 27 with a recognition rate of 95.36%. Hence the value of 27 serves as the best threshold to minimize to the barest minimum the error committed in the FAR and FRR while maximizing the recognition rate.



## 4.6 Comparison with Related Work

In order to prove the efficiency of our method as compared to other related works, a comparison was made as shown in Table 4.23. In the table, the mathematical algorithms were abbreviated by the taking the first alphabet of the authors last name. Since performance of algorithm is based on EER (Equal Error Rate) value, our proposed method happens to perform significantly compared to the minutia based system proposed. At this point, it should be noted that, such comparison

Table 4.23: Experimental Results on Database DB3\_A of FVC2004

MATHEMATICAL ALGORITHM	EER VALUE
YJ (Jiang and Yau, 2000)	24.5
PN (Parziale and Niel, 2004)	19.7
WLC (Wang et al., 2006)	12.0
QYW (Qi et al., 2005)	9.6
TK (Tico and Kuosmanen, 2003)	7.1
M3gl (Medina-Perez et al., 2012)	6.1
Proposed Method	4.6

can only be done if those you competing with used the same dataset from the same database. In our case, the dataset is DB3\_A from FVC2004 database. Hence, we are comfortable to say our method performs better even to the extent that those which we claim do not perform better this reverse bi-orthogonal wavelet 3.9 and 4.4 produces a minimal EER values making it efficient to the existing system (Minutia base).



## CHAPTER V

# CONCLUSIONS AND RECOMMENDATION

KNUST

### 5.1 Conclusion

This thesis proposes a fingerprint recognition technique using a wavelet based texture pattern recognition method. It was observed that fingerprint captured from different scanners will not be a problem in fingerprint recognition using the proposed method. The Discrete Wavelet Transform (DWT) used in extracting the associated statistical properties or features in fingerprint such as Directional Information, Center Area with Edge parameters from all the four sub band provides a much more substantial result with less computational time since no preprocessing was required as is the case of minutiae based systems.

It was also observed that the directional resolving power of wavelets extracts the texture information in Horizontal, Vertical and Diagonal directions of the fingerprint images and the use of this texture information leads to an increase in the performance rate. The use of multi-resolution, compactness and de-noising property of wavelets makes it useful in fingerprint recognition system. We conclude by emphasizing that using reverse bi-orthogonal wavelet transform helps in increasing the recognition rate compared to the existing system.



## 5.2 Further Work

There are number of research in pattern recognition that uses wavelet transform leading to new research area where more and new wavelets are been constructed to solve specified problems that facades the accuracy fetching good properties to aid in recognition process. Although a better result has been achieved in using this wavelet transform, there is still more to be done by considering different views and scenario.

KNUST





## REFERENCES

- Ackerman, A. and Ostrovsky, R. (n.d.), Fingerprint recognition, Master's thesis, UCLA Computer Science Department. Retrieved November 28, 2012, from <http://www.cs.ucla.edu/honors/UPLOADS/andrew/thesis.pdf>.
- Alfredo, C. L., Ricardo, R. L. and Queeman, C. R. (n.d.), 'Fingerprint recognition', *Electrical Engineering Department - Polytechnic University*.
- Bhowmik, U. K., Ashrafi, A. and Adhami, R. R. (2009), 'A fingerprint verification algorithm using the smallest minimum sum of closest euclidean distance.', *International Conference on Electrical, Communications, and Computers* pp. 90 – 95.
- Boyle, R. and Thomas, R. (1988), 'Computer vision: A first course', *Blackwell Scientific Publication* pp. 32 – 34.
- Canny, J. (1986), 'Computational approach to edge detection', *IEEE Trans. Pattern Anal. Mach. Intell, PAMI-8*: pp. 679 – 698.
- Chaohong, W. and Govindaraju, V. (2006), 'Singularity preserving fingerprint image adaptive filtering', *IEEE ASSP Magazine* pp. 313–316.
- Chengming, W., Tiande, G. and Wang, S. (2009), 'Fingerprint feature-point matching based on motion coherence', *Second International Conference on Future Information Technology and Management Engineering* pp. 226 – 229.



Chouhan, R. and Khanna, P. (2011), 'Robust minutiae watermarking in wavelet domain for fingerprint security', *World Academy of Science, Engineering and Technology* **60**, 1612–1619.

Conti, V., Militello, C., Vitabile, S. and Sorbello, F. (2010), 'Introducing pseudo singularity points for efficient fingerprints classification and recognition', *International Conference on Complex, Intelligent and Software Intensive Systems* **10**, 368 – 375.

Davies, E. (1990), *Machine Vision: Theory, algorithms and practicalities*.

Dechman, H. G. (2012), Fingerprint identification standards for emerging applications, Technical report.

Elmir, Y., Elberrichi, Z., Adjoudj, R. and Benyettou, M. (2009), 'Personal identification by fingerprints based on gabor filters', *the proceedings of 2009 Conference sur l'Informatique*.

Hong, L., Wan, Y. and Jain, A. (1998), 'Fingerprint image enhancement: Algorithm and performance evaluation', *IEEE Trans. Pattern Anal. Mach. Intell* **20**, 777 – 789.

Hong, Z. (1991), 'Algebraic feature extraction of image for recognition', *Pattern Recognition* **24**, 211 – 219.

Jain, A. K., Hong, L. and Bolle, R. (1997), 'Online fingerprint verification', *IEEE Trans. Pattern Anal. Mach. Intell* **19**, 302 – 314.

Jiang, X. and Yau, W. Y. (2000), 'Fingerprint minutiae matching based on the local



and global structures.', *In Proceedings of the 15th International Conference on Pattern Recognition, Barcelona, Spain, 2*, 1038–1041.

Jie, Z., Jinwei, G. and Zhang, D. (2007), 'Singular points analysis in fingerprints based on topological structure and orientation field', *Springer-Verlag Berlin Heidelberg* pp. 261 – 279.

Karu, K. and Anil, K. J. (1996), 'Fingerprint classification', *Elsevier Science Ltd* **29(3)**, 389 – 404.

Kass, M. and Witkin, A. (1987), 'Analyzing oriented patterns, computer vision, graph, image processing', pp. 362 – 385.

Kekre, H. B. and Bhatnagar, S. (2007), 'Fingerprint matching techniques', *In Proceedings of National Conference on Applications Digital Signal Processing. (NCDSP - 2007), Mumbai, .*

Khalil, S. M., Dzulkifli, M., Khan, M. K. and Khaled, A. (2010), 'Fingerprint verification based on statistical analysis', *Journal of Computer and Information Science and Engineering* **10**, 1657 – 1672.

Khan, U. M., Khan, S. A. and Rehman, R. U. (2009), 'A fingerprint verification system using minutiae and wavelet based features', *International Conference on Emerging Technologies .*

Kingsbury, N. (2006), 'Rotation-invariant local feature matching with complex wavelet', *Proc. European Conference on Signal Processing (EUSIPCO), Florence .*

Korner, T. W. (1996), 'Fourier analysis', *UK: Cambridge University Press .*



- Kulwinder, S., Kiranbir, K. and Ashok, S. (2011), 'Fingerprint feature extraction', *International Journal of Computer Science and Technology* **2(3)**, 237–141.
- Lavanya, B. N. and Raja, K. B. (2011), 'Performance evaluation of fingerprint identification based on dct and dwt using multiple matching techniques.', *International Journal of Computer Science Issues*, **8(6)(1)**, 275–283.
- Maio, D. and Maltoni, D. (1997), 'Direct gray scale minutia detection in fingerprints', *Transactions on PAMI* **19(1)**.
- Mallat, S. (1989), 'Multiresolution approximations and wavelet orthonormal bases of  $l_2(\mathbb{R})$ ', *Trans. Amer. Math. Soc.* **315**, 69 – 87.
- Mansukhani, P., Tulyakov, S. and Govindaraju, V. (2007), 'Using support vector machines to eliminate false minutiae matches during fingerprint verification', *SPIE Defense and Security Symposium (DSS)*, Orlando, FL., USA. .
- Manvjeet, K., Mukhwinder, S., Akshay, G. and Parvinder, S. S. (2008), 'Fingerprint verification system using minutiae extraction technique', *World Academy of Science, Engineering and Technology* pp. 497 – 502.
- Marion, A. (1991), *An introduction to image processing*, Chapman and Hall.
- Marr, D. and Hildreth, E. C. (1980), 'Theory of edge detection', *Proc. R. Soc., B* **207**: pp. 187 – 217. —
- Marr, D. and Ullman, S. (1981), 'Directional selectivity and its use in early visual processing', *Proc. R. Soc.*, pp. 200–208.
- Marr, M. (1982), *Vision*, number Chapter 2, Freeman.



- Medina-Perez, M. A., Garcia-Borroto, M., Gutierrez-Rodriguez, E. A. and Altamirano-Robles, L. (2012), 'Improving fingerprint verification using minutiae triplets', *sensors* pp. 3418–3437.
- Nixon, M. S. and Aguado, S. A. (2002), 'Feature extraction and image processing', *Butterworth and Heinemann* pp. 31 – 65.
- Oberst, U. (2007), 'The fast fourier transform', *Control and Optimization, SIAM* **46(2)**, 496 – 541.
- Pankanti, S., Prabhakar, S. and Jain, A. (2002), 'On the individuality of fingerprints', *IEEE Trans. Pattern Anal. Mach. Intell* **24(8)**, 1010 – 1025.
- Parziale, G. and Niel, A. (2004), 'A fingerprint matching using minutiae triangulation.', *n Proceedings of the 1st International Conference on Biometric Authentication, Hong Kong, China LNCS 3072*, 241–248.
- Patil, D. B., Kulkarni, J. V. and Holambe, R. S. (2006), 'Fingerprint verification using wavelet and local dominant', *Pattern Recognition*.
- Pokhriyal, A. and Sushma, L. (2010), 'A new method of fingerprint authentication using 2d wavelet', *Journal of Theoretical and Applied Information Technology* **13(2)**, 131–138.
- Prabhakar, S., Pankanti, S. H. and Jain, A. K. (2003), 'Biometric recognition: Security and privacy concerns', *IEEE Security and Privacy* pp. 33–42.
- Qi, J., Yang, S. and Wang, Y. (2005), 'Fingerprint matching combining the global orientation field with minutia.', *Pattern Recogn. Lett.* **26**, 2424–2430.



Qinghui, Z. and Xiangfei, Z. (2010), 'Research of key algorithm in the technology of fingerprint identification', *Second International Conference on Computer Modeling and Simulation* pp. 282 – 184.

Roddy, A. R. and Stosz, D. J. (1997), 'Fingerprint features statistical analysis and system performance estimates', *Proceedings of the IEEE* **85**(9), 1390 – 1421.

Saquib, Z., Soni, K. S., Suhasaria, S., Parekh, D. and Rekha, V. (2011), 'A faulttolerant approach for detection of singular points in noisy fingerprint images', *IJCSI International Journal of Computer Science Issues*, **8**(2), 577–582.

Semmlo, J. L. (2004), 'Biosignal and biomedical image processing', *Marcel Dekker Inc.*

Seung-Hoon, C., Kim, J. K., Lim, S. J., Pan, S. B., Moon, D. and Chung, Y. (2009), 'Ridge-based fingerprint verification for enhanced security', *International Conference on Consumer Electronics* pp. 1 – 2.

Sifuzzaman, M., Islam, M. R. and Ali, M. Z. (2009), 'Application of wavelet transform and its advantages compared to fourier transform', *Journal of Physical Sciences* **13**, 121–134.

Stollnitz, E., DeRose, T. and Salesin, D. (2006), 'Wavelets for computer graphics', *UK: Cambridge University Press.*

Tico, M. and Kuosmanen, P. (2003), 'Fingerprint matching using an orientation-based minutia descriptor.', *IEEE Trans. Pattern. Anal. Mach. Intell.* **25**, 1009–1014.



- Vetterli, M. and Herley, C. (1992), 'Wavelets and filter banks: Theory and design', *IEEE Transactions on Signal Processing* **40**(9), 2207 – 2232.
- Wang, W., Li, J. and Chen, W. (2006), 'Fingerprint minutiae matching based on coordinate system bank and global optimum alignment.', *In Proceedings of the 18th International Conference on Pattern Recognition, Hong Kong, China* **4**, 401–404.
- Wang, Y., Jiankun, H. and Fengling, H. (2007), 'Enhanced gradient-based algorithm for the estimation of fingerprint orientation fields', *Applied Mathematics and Computation* pp. 823–833.
- Woodward, J., Orlans, O. and Higgins, T. (2003), *Biometrics*, McGraw-Hill/Osborne.
- Xiang, M., Xiaopei, W. and Quanping, H. (2009), 'Algorithm based on point feature for fingerprint image segmentation', *World Applied Sciences Journal* **7** (Special Issue of Computer & IT): pp. 168–174.
- Xuzhou, L. and Yu, F. (2009), 'A new fingerprint matching algorithm based on minutiae', *Proceedings of International Council of Chemical Trade Associations* pp. 869 – 873.
- Yan, Y., Zhang, J. and Lades, M. (1997), 'Face recognition: Eigenface, elastic matching and neural nets', *Proceedings of the IEEE* **85**(9).
- Yang, J. C. and Park, D. S. (2008), 'Fingerprint verification based on invariant moment features and nonlinear bpnn', *International Journal of Control, Automation, and Systems*, **6**(6), 800–808.



Yilong, Y., Tian, J. and Xiukun, Y. (2004), 'Ridge distance estimation in fingerprint images: Algorithm and performance evaluation', *EURASIP Journal on Applied Signal Processing* pp. 495–502.

Ying, H., Tieniu, T. and Yunhong, W. (n.d.), 'An effective algorithm for fingerprint matching'.

Zhang, D. (2000), 'Authomated biometric technologies and system'.

Zhang, Y. and Jing, X. (2010), 'Spectral analysis based fingerprint image enhancement algorithm', *IEEE ASSP Magazine* .

Zheng, J. D., Gao, Y. and Zhang, M. Z. (2009), 'Fingerprint matching algorithm based on similar vector triangle', *Second International Congress on Image and Signal Processing* pp. 1 – 6.

