DEVELOPMENT OF AN IMPROVED APPROACH TO BIOMETRIC FINGERPRINT IMAGE COMPRESSION USING COIFLET SIGNAL TRANSFORMATION ALGORITHM

BY

BABATUNDE STEVEN EMMANUEL, B.Eng. (UAM), 2002, M.Sc. (Ibadan), 2010
PHD/ENG/7049/2010-2011
sbemmanuel@yahoo.com

A THESIS SUBMITTED TO THE SCHOOL OF POSTGRADUATE STUDIES, AHMADU BELLO UNIVERSITY IN PARTIAL FULFILMENT FOR THE AWARD OF DOCTOR OF PHILOSOPHY IN ELECTRICAL ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
FACULTY OF ENGINEERING
AHMADU BELLO UNIVERSITY, ZARIA
NIGERIA

JULY, 2015

DECLARATION

I declare that the work in this thesis entitled, **"Development of an Improved Approach to Biometric Fingerprint Image Compression Using Coiflet Signal Transformation Algorithm"** has been carried out by me in the Department of Electrical and Computer Engineering. The information derived from the literature has been duly acknowledged in the text and a list of references provided. No part of this thesis was previously presented for another degree or diploma at this or any other Institution.

Babatunde Steven **EMMANUEL** _____ _____

Signature Date

CERTIFICATION

This thesis entitled: DEVELOPMENT OF AN IMPROVED APPROACH TO BIOMETRIC FINGERPRINT IMAGE COMPRESSION USING COIFLET SIGNAL TRANSFORMATION ALGORITHM, by Babatunde Steven **EMMANUEL** meets the regulations governing the award of the degree of Doctor of Philosophy of the Ahmadu Bello University, and is approved for its contribution to knowledge and literary presentation.

Chairman, Supervisory Committee (Prof. M. B. Mu'azu)	_____ Signature	_____ Date
Member, Supervisory Committee (Dr. S. M. Sani)	_____ Signature	_____ Date
Member, Supervisory Committee (Dr. S. Garba)	_____ Signature	_____ Date
Head of Department (Prof. M. B. Mu'azu)	_____ Signature	_____ Date
Dean, School of Postgraduate Studies (Prof. A. Z. Hassan)	_____ Signature	_____ Date

DEDICATION

This research work is dedicated to the Almighty God who is the source of my inspiration and intellectual endowments.

ACKNOWLEDGEMENT

My highest appreciation goes to the Almighty God who is the source of my inspiration and intellectual strength.

I wish to express my profound gratitude and appreciation to my supervisors: Prof. M.B. Mu'azu (Head of Department, Electrical and Computer Engineering Department), Dr. S.M. Sani and Dr. S. Garba for their assistance and unflinching support during the course of this work despite their tight schedules. Your relentless patience and guidance has greatly enhanced my intellectual abilities. I am immensely grateful!

I also appreciate the moral support given to me by Professor Bajoga, Dr. Usman, Dr. Akinsanmi, Dr. A.M.S. Tekanyi (PG Coordinator, Electrical and Computer Engineering Department) and all the members of staff of Electrical and Computer Engineering Department.

I gratefully acknowledge the Director-General, Nigerian Institute of International Affairs (NIIA), Lagos, Professor Bola A. Akinterinwa and the Management and staff of the NIIA for giving me the opportunity to undertake this programme. I am particularly grateful for the enabling environment to carry out my research within the shortest possible time.

Finally, my special thanks goes to my darling wife and companion, Dr. Stella A. Emmanuel, my two great jewels; Tolulope and Oluwaseun Emmanuel for their understanding and support. To all my extended family, friends and well-wishers, you all are gratefully acknowledged for your prayers and the moral support.

ABSTRACT

Biometric fingerprint images require substantial storage, transmission and computation costs, thus their compression is advantageous to reduce these requirements. This research work presents a novel approach to biometric fingerprint image compression by the innovative application of non-uniform quantization scheme in combination with level-dependent threshold strategy applied to wavelet transformation as opposed to the widely used uniform quantization scheme. Comparative analysis of Coiflet wavelets implemented with level dependent thresholds and Daubechies wavelets were conducted on the basis of percentage retained energy, RE (%). The RE (%) values for Coiflet wavelet ranged from 99.32% to 99.69% as opposed to the values for Daubechies wavelet which ranged from 98.45% to 99.15%. These results revealed that the Coiflet wavelet bases performed better than the Daubechies wavelet. Hence, the choice of Coiflet wavelet for image transformation in the proposed compression algorithm was justified. The performance analysis of uniform and non-uniform scalar quantization schemes for biometric fingerprint image compression was conducted. The non-uniform quantization method based on Lloyd-Max approach performed better than the uniform quantization method used in the existing fingerprint compression standards. The Signal-to-Quantization Noise Ratio (SQNR) values for non-uniform quantization increased from 19.2977 dB for 3 bit per pixel (bpp) to 44.6083 dB for 7 bpp whereas for the same range (3 bpp to 7 bpp) for uniform quantization, SQNR values increased from 17.0903 dB to 40.1349 dB. Therefore, non-uniform quantization based on Lloyd-Max approach was employed for this compression algorithm. The implementation of the proposed biometric fingerprint image compression algorithm involved three stages, namely: the transformation of biometric fingerprint image; non-uniform quantization of transformed image and the entropy coding which is the final stage. In order to determine the overall performance of the algorithm, Peak Signal-to-Noise Ratio (PSNR) and Compression Ratio (CR) were used as performance metrics. PSNR was used as a measure of the resultant image quality after compression and the Compression Ratio was used as a measure of the degree of compression achievable. A trade-off was made between the achievable compression ratio and the realizable image quality which is a function of the achievable PSNR in the overall compression process. The overall performance of the proposed compression algorithm achieved an improvement in terms of compression ratio of 20:1 over the existing compression standard for biometric applications which have a compression ratio limit of 15:1. The improvement was largely due to the novel approach employed in this research work as stated above.

TABLE OF CONTENTS

Title page i

Declaration ii

Certification iii

Dedication iv

Acknowledgement v

Abstract vi

Table of Contents vii

List of Tables x

List of Figures xi

Abbreviations, Definitions, Glossary and Symbols xiii

CHAPTER ONE: INTRODUCTION

1.1 Background of Study 1

1.2 Types of Digital Image 2

1.2.1 Binary Image 2

1.2.2 Grayscale Image 2

1.2.3 True Colour Image 2

1.2.4 Indexed Image 3

1.3 Data Redundancy 3

1.3.1 Coding Redundancy 4

1.3.2 Interpixel Redundancy 4

1.3.3 Psychovisual Redundancy 5

1.4 Characteristics of Fingerprint Image 6

1.5 Fingerprint Image Compression Techniques and Standards 7

1.6 Motivation **9**

1.7 Aim and objectives **9**

1.8 Statement of Problem **10**

1.9 Justification of Study **10**

1.10 Methodology **10**

1.11 Significance of Study **11**

1.12 Scope and Limitation of Work **12**

1.13 Thesis Outline **12**

CHAPTER TWO: LITERATURE REVIEW

2.1 Introduction **13**

2.2 Review of Fundamental Concepts **13**

2.2.1 Transformation 14

2.2.2 Quantization 31

2.2.3 Entropy Coding 41

2.2.4 Validation and Performance Measures 57

2.3 Review of Similar Works **59**

CHAPTER THREE: MATERIALS AND METHODS

3.1 Introduction **69**

3.2 Materials **69**

3.3 Methods **69**

3.3.1 2-D Discrete Wavelet Transform (DWT) of Source Fingerprint Image 70

3.3.2 Non-Uniform Scalar Quantization using Lloyd-Max Procedure 78

3.3.3 Arithmetic Entropy Coding to Generate Image Bit-Streams 85

3.3.4 Reconstruction of Source Fingerprint Image from Compressed Bit-Stream 88

3.3.5 Validation and Evaluation of the Performance of Compression Algorithm 90

CHAPTER FOUR: RESULTS AND DISCUSSIONS

4.1 Introduction 95

4.1 Results of Performance Analysis of Wavelet Bases 95

4.2 Results of the Quantization Schemes 96

4.3 Results of Performance Evaluation of the Compression Algorithm 97

4.4 Discussion of Results 99

4.5 Performance Analysis of Wavelet Bases 99

4.6 Comparative Analysis of Global Threshold and Level-Dependent Threshold Strategies 101

4.7 Comparison of the Non-uniform Quantization and Uniform Quantization Methods 102

4.8 Performance Evaluation of the Compression Algorithm 106

4.9 Contribution to Knowledge 109

CHAPTER FIVE: CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion 111

5.2 Recommendations 113

References 114

Appendix I 120

Appendix II 121

LIST OF TABLES

Table 2.1:	Comparison between the Properties of Fourier and Wavelet Analyses	20
Table 2.2:	Information of Some Wavelet Families	26
Table 2.3:	Summary of the Review of Similar Works Based on Wavelet Image Compression Methods	66
Table 3.1:	Wavelet Analysis Filter Design Parameters	71
Table 3.2:	Design Parameters for Lossy Compression Based on Lloyd Max Algorithm	82
Table 3.3:	Coding Efficiency for Shannon-Fano, Huffman and Arithmetic Coding Schemes	86
Table 4.1:	Results of RE (%) and NZ (%) for Orthogonal Coiflet Wavelets	95
Table 4.2:	Results of RE (%) and NZ (%) for Orthogonal Daubechies Wavelets	95
Table 4.3:	Results of the Comparative Analysis of Global Threshold and Level-Dependent Threshold Strategies in Terms of Percentage Retained Energy	96
Table 4.4:	Comparison of the Non-uniform Quantization and Uniform Quantization Methods on the Basis of Quality and Distortion Measures	96
Table 4.5:	Comparison of the Dither-Based Non-uniform Quantization and Lloyd-Max Non-uniform Quantization Methods on the Basis of Quality and Distortion Measures	97
Table 4.6:	Performance Analysis of the Proposed Compression Algorithm on Fingerprint Images at Different Compression Ratios	98
Table 4.7:	PSNR Values Obtained for a Compression Ratio of 20:1	98

LIST OF FIGURES

Figure 1.1:	Biometric Features of a Fingerprint Image	7
Figure 2.1:	Block Diagram of a Wavelet Transform Based Image Compression System	14
Figure 2.2:	Basic Wavelet Filter Pair	15
Figure 2.3:`	Wavelet Packet Decomposition over Three Levels	16
Figure 2.4	Mallat Multi-Resolution Decomposition Tree	17
Figure 2.5:	Two-Scale Wavelet Decomposition	17
Figure 2.6:	Plots of Threshold Functions (a) Original Signal (b) Hard threshold, (c) Soft Threshold	29
Figure 2.7:	Block Diagram of Quantization Process and the Associated Noise	34
Figure 2.8:	Uniform, Gaussian, Laplacian and Gamma Distributions	48
Figure 2.9:	A Typical Huffman Binary Tree	49
Figure 2.10:	Typical Representation of the Arithmetic Coding Process	51
Figure 2.11:	Arithmetic Entropy Coding as the Final Stage of a Compression Process	52
Figure 3.1:	Flowchart for MATLAB Implementation of 2-D DWT Analysis of Fingerprint Image	73
Figure 3.2:	Estimation of RE (%) and NZ (%) in Wavelet Approximated Fingerprint Image using Daubechies Order 2 (db2) at Decomposition Level Three	76
Figure 3.3:	Estimation of RE (%) and NZ (%) in Wavelet Approximated Fingerprint Image using Coiflet Order 2 (coif2) at Decomposition Level Three	76
Figure 3.4:	Estimation of RE (%) and NZ (%) in Wavelet Approximated Fingerprint Image using Daubechies Order 4 (db4) at Decomposition Level Three	77
Figure 3.5:	Estimation of RE (%) and NZ (%) in Wavelet Approximated Fingerprint Image using Coiflet Order 4 (coif4) at Decomposition Level Three	77
Figure 3.6:	Histogram Plot of 5 bpp Quantized Fingerprint Output (a) and 8 bpp Source fingerprint Input (b)	79
Figure 3.7:	Plot of the Uniform Quantization Output Levels at 5 bpp	80

Figure 3.8:	Plot of the Non-niform Quantization Output Levels at 5 bpp	81
Figure 3.9:	Flowchart of the Lloyd-Max Quantization Procedures	83
Figure 3.10:	Program Flowchart of the Proposed Coiflet-based Lossy Compression Algorithm for Biometric Fingerprint Image	89
Figure 3.11:	3-level Decomposition of Fingerprint Image in the Compression Process	91
Figure 3.12:	Compression of Fingerprint Image with Values of Compression Ratio and their Corresponding PSNR Values	92
Figure 3.13:	Compression of Single Fingerprint Image with Values of Compression Ratio and their Corresponding PSNR Values	93
Figure 3.14:	Compression of Fingerprint Image with Values of Compression Ratio and their Corresponding PSNR Values	93
Figure 4.1:	Estimation of RE (%) in Wavelet Approximated Fingerprint Image using Daubechies and Coiflet Wavelet Bases at Decomposition Level Three	100
Figure 4.2:	Estimation of NZ (%) in Wavelet Approximated Fingerprint Image using Daubechies and Coiflet Wavelet Bases at Decomposition Level Three	101
Figure 4.3:	Plot of the Efficiency of Global Threshold and Level Dependent Threshold Strategies in Terms of Percentage Retained Energy	102
Figure 4.4:	Plot of MSQE Values for Non-uniform and Uniform Quantization	103
Figure 4.5:	Plot of SQNR in decibels (dB) for Non-uniform and Uniform Quantization	104
Figure 4.6:	Plot of MSQE Values for Dither-based Non-uniform Quantization and Lloyd-Max Non-uniform Quantization Schemes	105
Figure 4.7:	Plot of SQNR in decibels (dB) for Dither-based Non-uniform Quantization and Lloyd-Max Non-uniform Quantization Schemes	106
Figure 4.8:	Bar chart Plot of PNSR Values against their corresponding Compression Ratio (CR) for fingerprint image cmp00001.pgm	107
Figure 4.9:	Bar chart Plot of PNSR Values against their corresponding Compression Ratio (CR) for fingerprint image cmp00002.pgm	108

ABBREVIATIONS, DEFINITIONS, GLOSSARY AND SYMBOLS

Definitions of Terms

Algorithm: A set of procedures to a system task which comprises an encoding or decoding process.

Bit stream: Partially encoded or decoded sequence of bits comprising an entropy-coded segment.

Coding process: A general term for referring to an encoding process, a decoding process, or both.

Compressed image: A coded representation of an image.

Compression: Reduction in the number of bits used to represent source image data.

Decoder: An embodiment of a decoding process.

Decoding process: The process which takes as its input compressed image data or bit stream and outputs a continuous tone image

Dequantization: The inverse procedure to quantization by which the decoder recovers a representation of the data coefficients.

Reconstructed image: Grayscale image which is the output of the decoder

Source image (data): 8-bit grayscale image used as input to any encoder.

Image: A two-dimensional array of pixel values.

Downsampling: A procedure by which the spatial resolution of an image is reduced.

DWT: (Discrete Wavelet Transform) A linear transformation, implemented by a multirate filter bank, that maps a digital input signal to a collection of output subbands.

Encoder: An embodiment of an encoding process.

Encoding process: A process which takes as its input a grayscale source image and outputs compressed image data.

Entropy-coded Segment: An independently decodable sequence of entropy encoded bytes of compressed image data.

Entropy decoder: An embodiment of an entropy decoding procedure.

Entropy decoding: A lossless procedure which recovers the sequence of symbols from the sequence of bits produced by the entropy coder.

Entropy encoder: An embodiment of an entropy encoding procedure.

Entropy encoding: A lossless procedure which converts a sequence of input symbols into a sequence of bits such that the average number of bits per symbol approaches the entropy of the input symbols.

Image data: Either source image data or reconstructed image data.

Lossless: A descriptive term for encoding and decoding processes and procedures in which the output of the decoding procedure(s) is identical to the input to the encoding procedure(s).

Quantize: The act of performing the quantization procedure for a DWT coefficient.

Upsampling: A procedure by which the spatial resolution of an image is increased.

Abbreviations and Acronyms

BAC	-	Binary Arithmetic Coding
CBPC	-	Column Based Pass-Parallel Coding
CF	-	Context Formation
CR	-	Compression Ratio
CJIS	-	Criminal Justice Information System
CDF	-	Cohen Daubechies Feauveau
CDF	-	Cumulative Distribution Function
Coif	-	Coiflet
DCT	-	Discrete Cosine Transform
DEC	-	Decomposition
DFT	-	Discrete Fourier Transform
DWT	-	Discrete Wavelet Transform
dB	-	Decibel
db	-	Daubechies
EBCOT	-	Embedded Block Code with Optimized Truncation
EBC	-	Embedded Block Code
E[XN]	-	Expectation of XN
FT	-	Fourier Transform
FS	-	Fourier Series
FFT	-	Fast Fourier Transform
FWT	-	Fast Wavelet Transform
EZW	-	Embedded Zero-tree Wavelet
IDWT	-	Inverse Discrete Wavelet Transform
NIST	-	National Institute of Standards and Technology

NZ	-	Number of Zeros
MSE	-	Mean Square Error
MSQE	-	Mean Square Quantization Error
Pixel	-	Picture Element
PDF	-	Probability Density Function
PSNR	-	Peak Signal to Noise Ratio
PSQNR	-	Peak Signal to Quantization Noise Ratio
RBG	-	Red Blue Green
LL	-	Low Low
HH	-	High High
LH	-	Low High
HL	-	High Low
RE	-	Retained Energy
REC	-	Reconstruction
SOFM	-	Self Organizing Feature Map
SPPT	-	Sample Parallel Pass Type
SPIHT	-	Set Partitioning in Hierarchical Trees
STFT	-	Short Term Fourier Transform
WT	-	Wavelet Transform
WSQ	-	Wavelet Scalar Quantization
WBCT	-	Wavelet Based Contourlet Transform
VQ	-	Vector Quantization
1-D	-	One-Dimension
2-D	-	Two-Dimension

Mathematical Symbols

μ	-	Mean
α	-	Lower interval
β	-	Upper interval
ῡ	-	Sequence of decoded code values
Φ	-	Translation Parameter
Ψ	-	Scaling Parameter
ω_0	-	Fundamental Frequency
T_0	-	Fundamental Period
$\varphi_{j_o,k}(x)$	-	Scaling parameter
$\Psi_{j,k}(x)$	-	Wavelet parameter
$L^2(\mathbb{R})$	-	Square Integrable Real Space
$\mathbb{R}$	-	Real Numbers
σ_q^2	-	Mean Square Quantization Error
Δ	-	Quantization Step-size
σ	-	Standard Deviation
ϵ	-	Quantization error
$P(a_j)$	-	Probability that the source will produce a_j
σ^2	-	Variance of source distribution
μ	-	Mean of source distribution
$f_X(x)$	-	Probability Density Function
$\emptyset_0(S)$	-	Encoded source sequence
$\hat{S}(v')$	-	Decoded symbol sequence
$\tilde{v_k}$	-	Sequence of decoded code values
v'	-	Encoded code values of the source input

$E(S,L)$	-	Coding Efficiency
$H(S)$	-	Source entropy
$l'(S,L)$	-	Average code length
s_j	-	Symbol's probability of occurrence
l_j	-	Length of codeword
r_b	-	Bits per pixel of original image
r_c	-	Bits per pixel of compressed image
C_r	-	Compression Ratio

CHAPTER ONE

INTRODUCTION

1.1 Background of Study

Images contain large amount of information that requires huge storage space and large transmission bandwidth. Image data processing and storage attract cost and the cost is directly proportional to the size of data. In spite of the advancements made in mass storage and processing capacities, these have continued to fall below capacity requirements of application systems (Ashok et al., 2010). Therefore, it is advantageous to compress an image by storing only the essential information needed to reconstruct the image. An image can be thought of as a matrix of pixel (or intensity) values and in order to compress it, redundancies must be exploited. Image compression is the general term for the various algorithms that have been developed to address these problems.

Data compression algorithms are categorized into two, namely; lossless and lossy compression techniques. A lossless technique guarantees that the compressed data is identical to the original data whereas in lossy compression technique, images are compressed with some degree of data loss or degradation while still retaining their essential features. This distinction is important because lossy techniques are much more effective at compression than lossless methods. Lossy technique is the preferred choice for fingerprint image compression to reduce computation, storage and transmission costs. Huge volumes of fingerprint images that need to be stored and transmitted over a network of biometric databases are an excellent example of why data compression is important. The cardinal goal of image compression is to obtain the best possible image quality at a reduced storage, transmission and computation costs (Mallat, 2009).

1.2 Types of Digital Image

A digital image can be considered as a large array of discrete dots, each of which has a brightness associated with it. These dots are called picture elements or simply pixels (McAndrew, 2004). Spatial resolution is the density of pixels over the image. High spatial resolution simply means more pixels are used to display the image. Spatial frequencies define the amount by which image pixel values change with distance. High frequency components are characterized by large changes in pixel values over small distance and these are attributed to edges and noise. Low frequency components are parts characterized by little change in the pixel values and these are attributed to background and textures of images (McAndrew, 2004). There are four basic types of digital image, namely (McAndrew, 2004):

i) Binary image;

ii) Grayscale image;

iii) True colour image; and

iv) Indexed image

1.2.1 Binary Image

In this image type, each pixel is either black or white. There are only two possible values for each pixel (0, 1). Only one bit is needed to represent a pixel. Data for which a binary representation may be suitable include text, fingerprint and architectural plans (McAndrew, 2004).

1.2.2 Grayscale Image

This is an image in which each pixel is a shade of gray, that is from 0 (black) to 255 (white). This range means that each pixel can be represented by 8 bits, or exactly one byte. The grayscale levels are suitable for representing any type of natural images. Other grayscale range exist (-255 to 255), such image representations are often used in medical imaging applications (McAndrew, 2004).

1.2.3 True Colour Image

In this format, each pixel has a particular colour which is described by the amount of Red, Green, and Blue (RGB) in it. With image component range of 0 to 255, this gives a total of 256^3 = 16,777,216 different possible colours in the image. Since the number of bits required for representing each pixel is 24 bits, such images are referred to as 24-bits colour images (McAndrew, 2004).

1.2.4 Indexed Image

In this colour image type, the images only have a small subset of the 256^3 possible colours. For ease of storage and data handling, the image has an associated colour-map, or colour palette which is simply a list of all the colour components in the image. As opposed to the RGB image, each pixel of the indexed image has a value which does not give its colour but an index to the colour in the map (McAndrew, 2004).

In the compression of a digital gray-scale fingerprint, the source image has a lot of redundancies that need to be removed to achieve compression. These redundancies are not obvious in the spatial domain. Therefore, some kind of transformation method is needed to convert the image into another analysis domain that makes the redundancies more obvious. This process will lead to image decomposition and pixel energy de-correlation.

1.3 Data Redundancy

Since data compression refers to the process of reducing the amount of data required to represent a given quantity of information, a clear distinction must be made between data and information. Data are the means by which information is conveyed (Gonzalez and Woods, 2002). Various amounts of data may be used to represent a specified amount of information depending on the data source. More often than not, a large amount of data only contains a

small amount of relevant information (Gonzalez and Woods, 2002). This is a case of data redundancy which is a critical issue in digital image compression.

In digital images, the neighbouring pixels are correlated and, therefore, contain redundant information. The fundamental objectives of any compression process are to reduce redundancy and eliminate irrelevancy. Redundancy is defined as duplication of pixels, as well as irrelevancy which refers to insignificant pixels which are not noticeable by the Human Visual System (HVS) (Iqbal, 2004). In digital image compression, three basic data redundancies can be identified and exploited, namely (Gonzalez and Woods, 2002; Singla and Sharma, 2013):

i) Coding redundancy

ii) Interpixel redundancy

iii) Psychovisual redundancy

Image compression is achieved when one or more of these redundancies are reduced or eliminated. These redundancies are presented in greater details in the next sections.

1.3.1 Coding Redundancy

This occurs when the gray level pixels representing an image are coded in a way that uses more code symbols than absolutely necessary to represent each gray level pixel. Put differently, coding redundancy results when the codes assigned to pixel values of an image signal have not been selected to take full advantage of the probabilities of occurrence of the values (Gonzalez and Woods, 2002; Singla and Sharma, 2013).

1.3.2 Interpixel Redundancy

This is an important form of data redundancy. It directly relates to the interpixels correlation within an image. The value of any given pixel can be reasonably predicted from the value of its neighbours. As a result, the information carried by individual pixel is relatively small. In other words, much of the visual contribution of a single pixel to an image is redundant and it

can as well be derived from the values of its neighbours (Gonzalez and Woods, 2002; Singla and Sharma, 2013). The level of correlation between image pixel values is responsible for interpixel redundancy. This correlation is as a result of structural or geometric relationship between objects in the image. Consequently, interpixel redundancy is also referred to as spatial redundancy or geometric redundancy (Gonzalez and Woods, 2002). In order to reduce the interpixel redundancy in an image, its array of pixel values must be transformed into a mathematical domain where the image energy correlation of the constituent pixels can be lowered. This is achievable by the 2-D discrete wavelet transform (Gonzalez and Woods, 2002; Singla and Sharma, 2013).

1.3.3 Psychovisual Redundancy

The human eye does not respond with equal sensitivity to all visual information. Certain information simply has less relative importance than other information in normal visual processes. Such information with less relative importance are said to be psychovisually redundant (Gonzalez and Woods, 2002). This redundancy can be eliminated using suitable technique without any visually perceptible loss in image quality. Since the elimination of psychovisually redundant data results in a loss of quantitative information, the compression process is said to be lossy at this stage (Gonzalez and Woods, 2002; Singla and Sharma, 2013).

Image compression addresses the problem of reducing the amount of data required to represent a digital image. The basis for this reduction process is the removal of redundancies in the digital image data (Gonzalez and Woods, 2002; Iqbal, 2004).

1.4 Characteristics of Fingerprint Image

Fingerprint is the biometric features of a finger. The development of these features is congenital and maintains uniqueness among the population (Kazi et al., 2011). A fingerprint

usually appears as a series of dark lines that represents the high peaking portion of the friction ridge skin, while the valleys between these ridges appear as white space and are the low shallow portion of the friction ridge skin. Minutia is the term used to describe the location and direction of the ridge endings and bifurcations along a ridge path. The upper most point on the inner most ridge of the fingerprint image is known as core (George et al., 2012). The fingerprint has been used for personal identity verification for more than a century, and is the most widely used in biometrics today (Rajkumar et al., 2012). Fingerprint images are digitized at a resolution of 500 pixels per inch (ppi) with 256 levels of gray-scale information per pixel. A single fingerprint is about 700,000 pixels and needs about 0.6 Mbytes for storage. A pair of hands, then, requires about 6 Mbytes of storage (CJIS, 2000; Kumar et al., 2010). This huge storage requirement by fingerprint images impacts adversely on the efficiency of biometric application systems. The only way to improve on these resource requirements is to compress these images, such that they can be cost-effectively stored, transmitted and then reconstructed without compromising the essential biometric features such as the fingerprint's core, ridge endings and bifurcations. These features are shown in Figure 1.1.

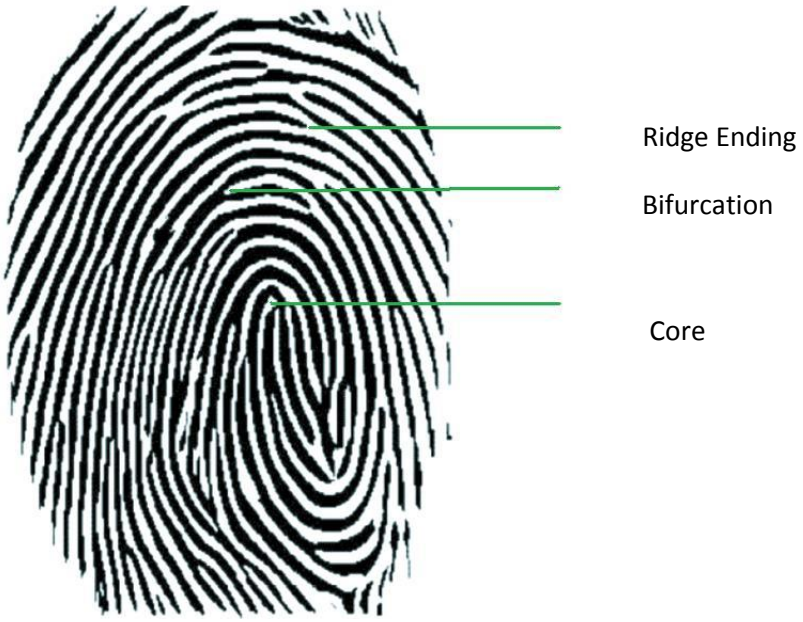


Figure 1.1: Biometric Features of a Fingerprint Image

1.5 Fingerprint Image Compression Techniques and Standards

The efficiency of the application of wavelet transform on image coding was significantly boosted by the introduction of embedded zero-tree wavelet (EZW) algorithm introduced by Shapiro (1993). The algorithm has since undergone significant improvements in the set partitioning in hierarchical trees (SPIHT) introduced by Said and Pearlman (1996). The EZW and SPIHT performed better than JPEG with most images. However, they both produced blurring effect on feature pattern of fingerprint images which renders the data useless for biometric application. Wavelet Scalar Quantization (WSQ) is a compression standard developed specifically for the compression of fingerprint images to improve the capability of preserving the fingerprint features for biometric pattern recognition. A compression ratio limit of 15:1 is specified for WSQ fingerprint compression standard (NIST, 2011). In order words, its performance becomes unsatisfactory at compression ratio higher than 15:1 (Onyshczak and Youssef, 2000; CJIS, 2000). The embedded block coding with optimized truncation of embedded bit-streams (EBCOT) by Taubman (2000) have resulted in

modern wavelet image compression and coding techniques. As a matter of fact, the latest Joint Photographic Expert Group 2000 (JPEG 2000) image coding standard was developed based on the EBCOT algorithm (Mallat, 2009; Usevitch, 2001). The EBCOT-based JPEG2000 as a robust general-purpose compression standard has the limitation of not being able to adequately preserve the crucial biometric features of fingerprint images at high compression ratio and it has the problem of complex algorithm implementation. The JPEG was the earlier version of JPEG2000 standard and it was based on discrete cosine transform technique while the JPEG 2000 was based on wavelet transform technique (Strang and Nguyen, 1996; Usevitch, 2001; Mallat, 2009).

The differences between JPEG2000 and WSQ standards are in their wavelet transform decomposition structures and the entropy coding method used. In wavelet decomposition, JPEG 2000 applies Mallat's algorithm or the pyramidal approach with Cohen Daubechies Feauveau (CDF), a variant Daubechies wavelet filters, while WSQ uses a fixed wavelet packet basis with the same CDF wavelet filter. WSQ uses raster scanning order, while JPEG2000 uses vertical bitplane scanning order (Strang and Nguyen, 1996). More significantly, both standards are based on Daubechies wavelets and have been adopted as fingerprint compression standards. Arithmetic entropy coding method was used in JPEG2000 while Huffman coding was used in WSQ. However, for the quantization process, uniform scalar quantization was used in both compression standards (Libert et al., 2012). It is noteworthy that JPEG 2000 is designed for general-purpose compression with significant flexibility. It has the disadvantage of complex algorithm implementation and high computation cost (Libert et al., 2012).

1.6 Motivation

Computer systems have provided the capability for the acquisition of massive volumes of biometric fingerprint data, and it is reasonable to develop computational techniques to help extract, store in compact form and derive meaningful patterns and structures from these data in order to address the problem of data overload. Additionally, data mining and identity verification processes for law enforcement, immigration services, and forensic applications can immensely benefit from an efficient lossy compression method. This is the motivation behind this research work.

1.7 Aim and Objectives

The aim of this research work is to implement an efficient and reliable compression method that can significantly reduce the image size of fingerprint images while retaining the essential biometric features resulting in reduction of storage, transmission and computation costs. The objectives are to:

(i) achieve 2-D discrete wavelet transform of fingerprint image into an array of transformed coefficients using an efficient wavelet bases implemented with robust threshold strategy;

(ii) achieve an efficient quantization of transformed image coefficients to realize further compression with minimum distortion or degradation in the process;

(iii) achieve an efficient entropy coding using the most effective symbolic representation scheme. An efficient entropy coding provides the process for coding strings of symbols in a source image for cost-effective storage, and it requires no training or pre-stored tables or codebook generation;

(iv) achieve an efficient compression with good quality that retains essential fingerprint image features with significant reduction in storage and computation costs. The

proposed algorithm is expected to achieve a compression ratio higher than the WSQ compression limit of 15:1 and still retains the essential biometric features of the compressed images;

(v) validate and evaluate the performance of proposed algorithm using the compression ratio (CR) and peak signal-to-noise ratio (PSNR) as performance metrics.

1.8 Statement of Problem

The problems that have been identified in the existing fingerprint compression standard include: the limitation of a compression ratio of 15:1 which could be improved with better algorithm. High complexity of image encoding process of the existing techniques is also a problem. Most of the existing methods require the generation of codebooks or lookup tables which require additional computational cost for implementation. In addition, the significant degradation in the biometric features of fingerprint at compression ratio higher than 15:1 as a consequence of the uniform quantization method used in the existing compression scheme remains a major challenge.

1.9 Justification of Study

Fingerprint images contain very large data which require massive storage, computation, transmission bandwidth costs. The preservation of the essential features of fingerprint images for effective biometric application is very crucial. Therefore, the investigation of an efficient compression method that can significantly reduce fingerprint image size while preserving its biometric properties (the core, ridge endings and bifurcations) with compression ratio higher than 15:1 is justified.

1.10 Methodology

The research methodologies employed in this study covered the transformation. Quantization and entropy coding stages of a standard lossy compression scheme and these include:

i) 2-Dimensional Discrete Wavelet Transform (DWT) of source fingerprint image;

ii) Non-uniform scalar quantization using Lloyd-Max procedure for transformed images;

iii) Arithmetic entropy coding of transformed and quantized coefficients to generate image bit-streams;

iv) Reconstruction of Fingerprint image from bit-stream; and

v) Validation and evaluation of compression algorithm using Compression Ratio (CR) and Peak Signal to Noise Ratio (PSNR) performance metrics

1.11 Significance of Study

Biometrics system is an emerging technology that deals with automatic identification of individuals, based on their physiological and behavioral characteristics. The biometric traits must satisfy universality, uniqueness, permanence, accessibility and collectivity (Kazi et al., 2011). Fingerprint is one of the physiological biometrics used for human identity verification (George, et al., 2012, Villegas et al., 2006). The methods for human identity authentication based on biometrics using the physiological and behavioural characteristics of a person have been evolving continuously and seen significant improvement in performance and robustness over the last few years. However, the performances of most of the existing systems are far from satisfactory in terms of robustness and accuracy (George, et al., 2012). To address these challenge and satisfy the requirements of new and emerging biometric applications, there is the need for the development of improved method of cost effectively storing and transmitting huge volume of fingerprint images that are collected for biometric applications.

1.12 Scope and Limitation of Work

This research work covers source coding of biometric fingerprint image using the combined techniques of wavelet transform, non-uniform quantization, level-dependent threshold and entropy coding for efficient storage. It does not include channel coding for transmission bandwidth and computation costs estimation and management.

1.13 Thesis Outline

This thesis is divided into six chapters. Chapter one presents the background of study, research motivation, problem statement, significance of study, as well as aim and objectives of the research work. The review of fundamental concepts which are pertinent to this research work and a critical review of similar research works are presented in chapter two. Chapter three outlines the various research methodologies adopted in the work. Chapters four discusses the results of the experimental analyses carried out in the work, as well as, contribution to knowledge. Lastly, the research summary, conclusion and recommendation for future research work are provided in Chapter five.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

The review of literature is approached in two ways, namely:

i) Review of fundamental concepts pertinent to the research work; and

ii) Critical review of similar research works in chronological order.

2.2 Review of Fundamental Concepts

A standard lossy image compression system involves three fundamental image coding stages, namely (Strang and Nguyen, 1996; Usevitch, 2001):

(i) Transformation;

(ii) Quantization; and

(iii) Entropy Coding.

The transformation which is achieved through suitable mathematical transformation technique (such as Fourier and Wavelet transforms) is applied to the image with the aim of converting it into a different domain where the compression will be easier. In the transform domain, correlation level can be lowered, and the energy can be concentrated in a small portion of the transformed image. The quantization, which can either be vector or scalar quantization or their variations, is the stage that is mostly responsible for the 'lossy' character of the system. It entails a reduction in the number of bits used to represent the pixel values of the transformed image (also called transform coefficients). Coefficients of low contribution to the total energy or the visual appearance of the image are coarsely quantized (represented with a small number of bits) or even discarded, whereas more significant coefficients are subjected to a finer quantization (represented with more bits). At the entropy coding stage, further compression is achieved with the aid of some entropy coding scheme such as

Huffman, Shanon Fano, arithmetic coding etc. where the non-uniform distribution of the symbols in the quantization result is exploited so as to assign fewer bits to the most likely symbols and more bits to unlikely ones. This results in a size reduction of the resulting bit-stream on the average and the conversion that takes place at this stage is lossless (Strang and Nguyen, 1996; Usevitch, 2001).

In summary, fingerprint image compression algorithm using wavelet transform method is implemented in two parts, namely (Usevitch, 2001):

(i) Image encoding for converting image to optimized bit stream; and

(ii) Image decoding for reconstructing the image from optimized bit stream.

The implementation of each of the parts of the proposed algorithm is in three stages. They are Discrete Wavelet Transform (DWT)/Inverse DWT, Scalar quantizer/Scalar dequantizer and Entropy encoder/decoder. This is depicted in Figure 2.1:

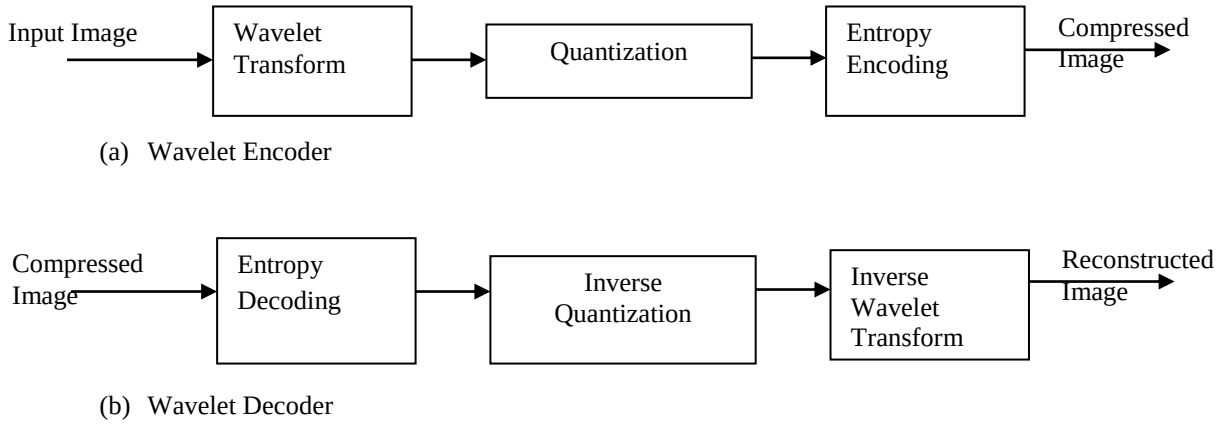


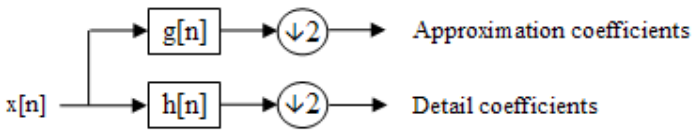
Figure 2.1: Block Diagram of a Wavelet Transform Based Image Compression System

2.2.1 Transformation

Wavelet coding is based on the idea that the coefficients of a transformed image that de-correlates the energy of its pixel values can be coded more efficiently than the original image's array of pixel values. This is because wavelet transform basis packs most of the important visual information, such as the biometric features of fingerprint image, into a small number of coefficients. The remaining coefficients can be truncated to zero using suitable

thresholding method with little degradation in image quality (Gonzalez and Woods, 2002; Mallat, 2009).

Wavelet image pixel energy de-correlation for compression involves three aspects: decomposing an image signal to obtain wavelet coefficients, modifying the wavelet coefficients through appropriate threshold technique and reassembling the signal from the coefficients to reconstruct the original signal. Wavelet has two analysis functions: wavelet and scaling functions which are implemented as high pass and low pass filters respectively (Mallat, 2009). The decomposition of signal into different frequency bands is simply obtained by successive high pass and low pass filtering of the source image signal. This filter pair is called multi-resolution analysis filter pair (Strang and Nguyen, 1996). For many signals, the low-frequency content g[n] is the most important part and this is known as approximation. It is what gives the signal its identity. The high-frequency content h[n], on the other hand, imparts flavour or nuance and this is referred to as details. The approximations are the high-scale or low-frequency components of the signal. The details are the low-scale or high-frequency components. The filtering process, at its most basic level is as depicted in Figure 2.2 (Strang and Nguyen, 1996).



.

Figure 2.2: Basic Wavelet Filter Pair

The original signal sequence, x[n] passes through two complementary filters g[n] and h[n] and produces two signals. Unfortunately, the output produced twice as much data samples as the input samples. Therefore, to correct this problem, every second (even) data sample in the array is discarded (a process referred to as down-sampling or decimation) (Srivastava et al., 2011). This process introduces aliasing (distortion) in the image signal which can be

accounted for later in the process chain (Misiti et al., 1996). Two basic decomposition algorithms are commonly used. These include: Wavelet Packet Decomposition (WPD) and the Mallat Algorithm (Misiti et al., 1996).

The wavelet packet decomposition is the general form of wavelet multi-resolution decomposition. This decomposition process is realized by iteratively filtering the image signal by a pair of conjugate mirror filters, that is low pass filter and highpass filter pair (Luo and Zhang, 2012) as shown in Figure 2.3. At each decomposition level, the wavelet coefficients are calculated.

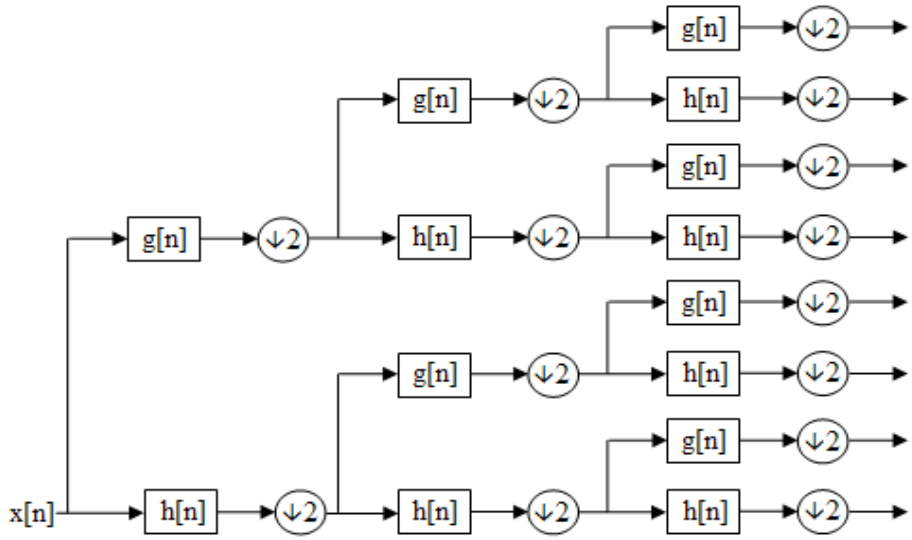


Figure 2.3: Wavelet Packet Decomposition over Three Levels (Misiti et al., 1996).

In the Mallat algorithm (2009) also known as wavelet transform decomposition, the decomposition process is iterative with successive approximations being decomposed in turn, so that one signal is broken down into many lower-resolution components. This is called the wavelet decomposition tree. Since the analysis process is iterative, in theory it can be continued indefinitely. In reality, the decomposition can proceed only until the individual details consist of a single sample or pixel. A three-level wavelet decomposition using Mallat algorithm is depicted in Figure 2.4.

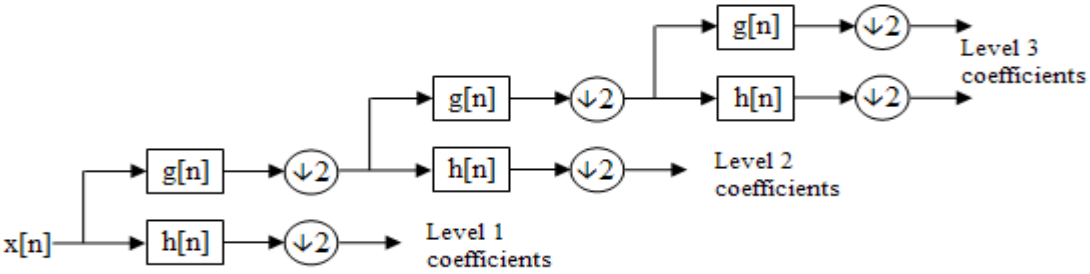


Figure 2.4: Mallat Multi-Resolution Decomposition Tree (Mallat, 2009).

This is a non-uniform band splitting method that decomposes the low frequency image portion that produced the approximation component into narrower bands and the high-pass detail component output at each level is left without any further decomposition. The resulting two-dimensional array of coefficients contains four bands of data, each labelled as LL (low-low), HL (high-low), LH (low-high) and HH (high-high) (Srivastava et al., 2011). The LL band is further decomposed iteratively to level three as shown in Figure 2.5.

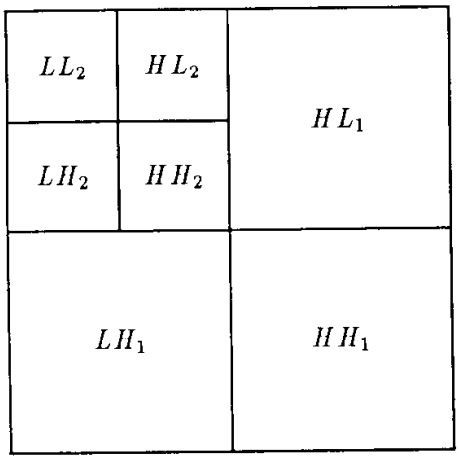


Figure 2.5: Two-Scale Wavelet Decomposition (Srivastava et al., 2011)

Wavelet thresholding process involves achieving an appropriate selection of the type of threshold for improved performance in image compression (Luo and Zhang, 2012).Wavelet thresholding explores the ability of wavelet bases to approximate image signal with only a few non-zero coefficients (Srivastava et al., 2011; Luo and Zhang, 2012). Therefore, the choice of wavelet bases is an important factor in wavelet thresholding. Thresholds are used to estimate the level of significant wavelet coefficient as distinct from the level of insignificant

wavelet coefficients. The coefficient values larger than threshold are regarded as significant and the values less than the threshold are regarded as insignificant (Luo and Zhang, 2012).

Most wavelet coefficients at their finest scale do not contribute to the energy of the image signal, whereas very few of the coefficients contribute to the perceptual quality of the image signal. The use of fixed or global threshold method produces equal truncation effect at all decomposition levels regardless of the variation in image energy distribution across levels. But to achieve a more comprehensive filtering operation, thresholding operation is applied level by level in the decomposition taking into cognizance the energy variation at different decomposition levels. This method is known as level-dependent threshold or Birge-Massart strategy (Luo and Zhang, 2012).

After the process of image signal decomposition and thresholding, there is the need to reassemble the decomposed components to get back the image signal, with minimal loss of information. This process is called reconstruction, or synthesis. The mathematical manipulation for realizing image signal synthesis is called the inverse discrete wavelet transforms (IDWT). The wavelet reconstruction process consists of upsampling and filtering. Upsampling is the process of lengthening a signal component by inserting zeros between samples and it is the reverse process of down-sampling (Mallat, 2009). It will be recalled that the downsampling of the signal components performed during the decomposition phase introduced a distortion called aliasing. It turns out that the up-sampling process at the reconstruction phase neutralizes the effect of aliasing or distortion introduced by the down-sampling process (Strang and Nguyen, 1996). The filtering part of the reconstruction process is very important and the choice of filters is crucial in achieving perfect reconstruction of the source image signal.

There are different methods for the filtering of image signals. The most well-known approach is the signal spectral analysis filtering using Fourier Techniques and the most recent method

is wavelet multi-resolution analysis (Strang, 1994; Polikar, 2001, Mallat, 2009). Fourier Series (FS) is Joseph Fourier's original work which involved the representation of a periodic function as a finite, weighted sum of sinusoids that are integer multiples of the fundamental frequency of the analysis signal (Polikar 2004). These frequencies are said to be harmonically related or simply harmonics. Continuous Fourier Transform (FT) is an extension of Fourier series to non-periodic functions. Any continuous aperiodic function can be represented as an infinite sum (integral) of sinusoids. Discrete Time Fourier Transform (DTFT) is an extension of FT to discrete sequences where the discrete function is also represented with an infinite sum (integral) of sinusoids. The frequency representation of DTFT is not discrete but continuous. Discrete Fourier Transform (DFT) is an extension of DTFT, where the frequency variable is also discretized. Fast Fourier Transform (FFT) is mathematically identical to DFT, however, it has a more significantly efficient implementation. Given an N-sampled signal sequence, the number of computations for DFT is N^2 whereas the number of computations for FFT is $N \log_2 N$ (Mallat, 2009).

The Short-Time Fourier Transform (STFT), maps a signal into a two-dimensional function of time and frequency. The STFT method can analyze a non-stationary signal in the time domain through a segmented algorithm. STFT processes the signal with a sliding window having constant length in time or space. Through a moving window process, the original signal is broken up into a set of segments, and each segment is processed by the conventional Fast Fourier transform (FFT) algorithm. In the end, all the results in frequency domain are summed up (Giurgiutiu and Lingyu, 2003).

Wavelet analysis is a signal windowing technique with variable-sized regions. It is a new idea in signal processing. Instead of analysing cosine waves that go on forever or get chopped off, the new building blocks are wavelets. These are little waves that start and stop. A long signal is broken into a basis of possible signals, called wavelets. The wavelets come from a single

function w(x) by shifting and scaling (Strang, 1994; Mallat, 2009). Amplitudes or coefficients of transformation are processed and reconstructed to recover a synthesis signal. Redundant coefficients are discarded to achieve compression. One major advantage afforded by wavelets is the ability to perform local analysis, that is, to analyse a localized area of a large signal. Wavelet analysis is capable of revealing aspects of data that other signal analysis techniques miss, aspects like trends, breakdown points, discontinuities in higher derivatives, and self-similarity. This is because it gives a different view of data than those presented by traditional techniques. Wavelet analysis can often compress a signal without appreciable degradation. The summary of the advantages of wavelet analysis and Fourier analysis, as well as, its variants is shown in Table 2.1.

Table 2.1: Comparison between the Properties of Fourier and Wavelet Analyses

Criteria	DFT/FFT	STFT	DWT
1. Applications	Periodic and aperiodic signal pattern recognition and trends	Pattern recognition, trends and compression for non-stationary signal with mild discontinuities	Localization characterization, pattern recognition, trends and compression for non-stationary signal with sharp discontinuities
2. Sliding Window	No windowing	Fixed-size windowing	Variable-size windowing
3. Analysis	Frequency representation/ Spectra analysis	Time-Frequency representation	Multi-resolution analysis
4. Uncertainty Principle	Do not satisfy	Do not satisfy	Satisfy
5. Input signal	Best suited for stationary signals	Best suited for non-stationary signals with mild discontinuities	Best suited for non-stationary signals with sharp discontinuities
6. Number of Computation (N samples)	DFT = N^2 Computations FFT = N $\log_2$ N Computations	STFT = N $\log_2$ N Computations	DWT = N Computations

It is for these obvious reasons that wavelet technique has been adopted as the preferred tool for fingerprint image compression in this research work.

2.2.1.1 *Discrete Wavelet Transform (DWT)*

Discrete wavelet transform (DWT) filter is used for reliable image transformation. The DWT is recognised for its capability of space-frequency decomposition of images, energy compaction of low frequency subbands, and space localization of high frequency subbands (Mallat, 2009; Ashok et al., 2010). DWT decomposes a discrete image signal into bands that vary in spatial frequency and orientation.

In DWT, a space-frequency representation of the digital image signal is obtained using digital filtering techniques. The signal to be analyzed is passed through filters with different cutoff frequencies at different scales. For a discrete sampled signal sequence f(x), where integer x = 0,1,2 ... m-1. There is m number of samples in the sequence.

The scaling function $w_{\emptyset}(j_o, k)$ and wavelet function $w_{\Psi}(j, k)$ are given by (Mallat, 2009):

$$w_{\varphi}(j_o, k) = \frac{1}{\sqrt{m}} \sum_x f(x) \varphi_{j_o,k}(x) \quad (2.1)$$

$$w_{\Psi}(j, k) = \frac{1}{\sqrt{m}} \sum_x f(x) \Psi_{j,k}(x) \quad (2.2)$$

Note that $\frac{1}{\sqrt{m}}$ is a normalizing term. This is to ensure that the energy of the signal in the two transformation domain remains the same.

The inverse discrete wavelet transform (IDWT) is given by (Mallat, 2009):

$$f'(x) = \frac{1}{\sqrt{m}} \sum_k w_{\varphi}(j_o, k)\, \varphi_{j_o,k}(x) + \sum_{j=j_o}^{\infty} \sum_k w_{\Psi}(j, k) \Psi_{j,k}(x) \quad (2.3)$$

Where:

$f(x)$ = Original signal sequence

$f'(x)$ = Reconstructed signal sequence

$\varphi_{j_o,k}(x)$ = Scaling parameter

$\Psi_{j,k}(x)$ = wavelet parameter

Equations (2.1) and (2.2) can be called forward discrete wavelet transform while equation (2.3) is the inverse discrete wavelet transform. Every mathematical transformation involves

the original signal, the transformed signal and the transformation kernel. In this case, the transformation kernels are $w_{\emptyset}(j_o, k)$ and $w_{\Psi}(j, k)$. Note that for a more efficient computational implementation of DWT, fast wavelet transform (FWT) algorithm is used. The number of FWT computations for N-sampled sequence is equal to N (Polikar, 2001; Mallat, 2009).

2.2.1.2 *Wavelet Transform in Two Dimensions*

For analytic transformation of image signal, a two-dimensional (2-D) discrete wavelet transform is used which can easily be extended from a one-dimensional (1-D) wavelet transform. To achieve this, one 2-D scaling function, $\varphi(x, y)$, and three 2-D wavelets: $\Psi^H(x, y)$, $\Psi^V(x, y)$, and $\Psi^D(x, y)$ are required. Each is the product of 1-D scaling function φ and corresponding 1-D wavelets φ as shown (Gonzalez and Woods, 2002):

$$\varphi(x, y) = \varphi(x)\varphi(y) \tag{2.4}$$

$$\Psi^H(x, y) = \Psi(x)\varphi(y) \tag{2.5}$$

$$\Psi^V(x, y) = \varphi(x)\Psi(y) \tag{2.6}$$

$$\Psi^D(x, y) = \Psi(x)\Psi(y) \tag{2.7}$$

Equation 2.4 defines the separable scaling function, $\varphi(x, y)$. Equations (2.5) to (2.7) define the wavelet functions that measure the functional variations of intensity or grayscale for images along different directions: Ψ^H defines variation along columns (horizontal edges); Ψ^V defines variation along rows (vertical edges); Ψ^D defines variation along diagonals.

Given separable 2-D scaling and wavelet functions, 2-D DWT can be defined. First, the scaled and translated or shifted basis functions are defined as follows:

$$\varphi_{j,m,n}(x, y) = 2^{\frac{j}{2}}\varphi\left(2^j x - m, 2^j y - n\right) \tag{2.8}$$

$$\Psi^i{}_{j,m,n}(x, y) = 2^{\frac{j}{2}}\Psi^i\left(2^j x - m, 2^j y - n\right), \quad i = \{H, V, D\} \tag{2.9}$$

Where, i = directional wavelet index

Therefore, 2-D DWT of function f(x, y) of size MxN is given by (Gonzalez and Woods, 2002):

$$w_{\varphi}(j_0, m, n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y)\, \varphi_{j_0,m,n}(x, y) \quad (2.10)$$

$$w^{i}{}_{\Psi}(j, m, n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y)\, \Psi^{i}{}_{j,m,n}(x, y), \;\; i = \{H, V, D\} \quad (2.11)$$

Where,

j_0 = Arbitrary starting scale ($j_0 = 0$)

$w_{\varphi}(j_0, m, n)$ = Approximation coefficients for f(x, y) at scale j_0

$w^{i}{}_{\Psi}(j, m, n)$ = Horizontal, vertical and diagonal details coefficients at scales $j \geq j_0$

$M = N = 2^j, for\ j = 0,1,2, \dots, j-1$

$m, n = 0,1,2, \dots, 2^j - 1$

Given w_{φ} and $w^{i}{}_{\Psi}$, f(x, y) can be obtained from 2-D Inverse DWT as follows (Gonzalez and Woods, 2002):

$$f(x, y) = \frac{1}{\sqrt{MN}} \sum_m \sum_n w_{\varphi}(j_0, m, n)\, \varphi_{j_0,m,n}(x, y) +$$

$$\frac{1}{\sqrt{MN}} \sum_{i=H,D,V} \sum_{j=j_0}^{\infty} \sum_m \sum_n w^{i}{}_{\Psi}(j, m, n)\, \Psi^{i}{}_{j,m,n}(x, y) \quad (2.12)$$

It should be noted that since image signal has two dimensional data structure, 2-D DWT will be implemented for fingerprint image transformation in this research work.

The wavelet transform provides for space-frequency multi-resolution analysis. The first step in the implementation of a wavelet transform is to determine a mother wavelet. The mother wavelet is also known as wavelet prototype or basis (Polikar, 2001). There are many types of mother wavelets and each one has its own application. Once a mother wavelet is decided, it is translated through the signal using the mathematical technique, called convolution (Mallat, 2009). The windowing technique is used by changing the scale of the wavelet. This is basically, the dilation and compression of the wavelet when translated. The idea behind

dilation and compression is that if the wavelet is compressed it represents high frequency and if it is dilated, it has a slow rate of change, that is, low frequency (Polikar, 2001). As wavelet is translated and dilated multiple times through the signal, the wavelet quantifies how well it correlates to the topology of the signal. The correlation of the wavelet receives a value of how well it matches the signal. If there is a high correlation, the transform reports high value at a particular scale and position in time. If there is no correlation, the transform reports low index value of correlation (Polikar, 2001). The term mother wavelet gets its name from two important properties of the wavelet analysis. First, the term wavelet means a small wave and smallness refers to the condition that this window function is of finite length (compactly supported). The wave refers to the condition that this function is oscillatory. Secondly, the term mother implies that the functions with different region of support that are used in the transformation process are derived from one main function, or the mother wavelet. In other words, the mother wavelet is a prototype for generating the other analysis window functions (Strang and Nguyen, 1996).

2.2.1.3 *Wavelet Bases or Prototypes*

Even though, wavelet analysis is the preferred techniques for image signal compression, there are different types of wavelets and their qualities vary according to several criteria. The main criteria include: support of wavelet and scaling functions; the symmetry, the number of vanishing moments; the regularity, the orthogonality or biorthogonality (Mallat, 2009). Some of the wavelet prototypes include: Haar, Daubechies; Coiflet, Symlet, Mexican-Hat, Morlet, Biorthogonal wavelets etc (Misiti et al., 1996, Mallat, 2009).

Haar wavelets are the simplest and first to be invented in the wavelets family. Although, wavelets are advanced topic but the Haar wavelets are elementary (Strang, 1994). Therefore, there is the need to turn to better wavelet bases or prototypes or functions. The new wavelets are more sophisticated. Popular among them are the Daubechies and Coiflet wavelet

functions. Most Daubechies wavelets are not symmetrical. For some, the asymmetry is very pronounced. The regularity property increases with the order of wavelet function. Regularity means that the iterated wavelet filter converges to a continuous function. The Daubechies wavelet analysis is orthogonal. Coiflet wavelet function, on the other hand, was developed to fix the symmetry limitation of Daubechies wavelet. The coiflet wavelet and scaling functions are much more symmetrical than that of Daubechies (Misiti et al., 1996; Islam et al., 2012). The symmetry property of a wavelet matrix is given by its product with its transpose to obtain an identity matrix. This property guarantees a perfect reconstruction in the transformation domain. In principle, a wavelet is constructed as a piecewise function of which its dilated versions and translated versions form an orthonormal basis in square integrable real space, $L^2(\mathbb{R})$ (Mallat, 2009). This leads to the understanding of the connection between wavelet bases and conjugate mirror filters used in multirate filter banks. These filter banks implement a fast orthogonal wavelet transform that requires a computation of only O (N) operations for signal of size N (Mallat, 2009).

Even though, wavelet analysis is the preferred technique for image signal compression, there are different types of wavelets and their qualities vary according to several criteria. The main criteria include (Misiti et al., 1996; Strang, et al., 1996; Mallat, 2009):

i) the support of wavelet and scaling functions;

ii) the symmetry;

iii) the number of vanishing moment;

iv) the regularity; and

v) the orthogonality or biorthogonality.

Both orthogonal wavelets and biorthogonal wavelets can be used in wavelet transform analysis. Daubechies wavelets, Symlets, Coiflets and biorthogonal wavelets are some of the examples in the wavelet family.Their properties are listed in Table 2.2 (Mallat, 2009).

Table 2.2: Information of Some Wavelet Families

Wavelet name	Order	Number of vanishing moments	Size of support	Orthogonality
Daubechies	$N_{\{1 \le N < \infty\}}$	N	2N-1	Orthogonal
Symlets	$N_{\{2 \le N < \infty\}}$	N	2N-1	Orthogonal
Coiflets	$N_{\{1 \le N \le 5\}}$	2N	6N-1	Orthogonal
Biorthogonal wavelets	$N_{\{1 \le N \le 8\}}$ (dec) $N_{\{1 \le N \le 6\}}$ (rec)	N_r	$2N_d+1$ (rec) $2N_r+1$ (dec)	Biorthogonal

Popular among orthogonal wavelets are the Daubechies and Coiflet wavelet functions. Note that dec and rec stand for decomposition and reconstruction respectively. The regularity property of wavelet increases with its order. Regularity means that the iterated wavelet filter converges to a continuous function. Daubechies wavelets lack the property of symmetry. However, Coiflet wavelet was developed to fix the symmetry limitation of Daubechies wavelet. The Coiflet wavelet parameter and scaling parameter are much more symmetrical than that of Daubechies (Wei, 1998, Luo et al., 2012; Dixit and Majumdar, 2013).

2.2.1.4 *Design and Construction of Coiflet Wavelets*

In many image processing applications, one desirable property for filter bank design is linear phase, which corresponds to the symmetry or antisymmetry of the associated wavelets and the symmetry of the associated scaling functions. However, it is well-known that symmetric scaling function is non-existent in most wavelet family (Daubechies, 1992). Therefore, in order to obtain symmetric scaling functions, and still produce wavelet function with linear phase property, Coiflet wavelets were developed to realize these properties. The Coiflet wavelets have the same number of vanishing moments for both the scaling function (low pass filter) and the wavelet function (high pass filter). In addition, Coiflets have been shown to be excellent for the sampling approximation of smooth functions (Daubechies, 1992). In this section, the theoretical framework for designing Coiflet wavelets, their mathematical properties and potentials in image compression application are presented.

Let *h(n)* be the lowpass filter in a two-channel orthonormal wavelet system. The scaling function $\emptyset(t)$ is recursively defined by the dilation equation (Wei, 1998):

$$\emptyset(t) = \sqrt{2}\sum_{n\in\mathbb{Z}} h(n)\emptyset(2t-n) \quad (2.13)$$

And the wavelet $\psi(t)$ is defined as:

$$\psi(t) = \sqrt{2}\sum_{n\in\mathbb{Z}} g(n)\psi(2t-n) \quad (2.14)$$

where the lowpass filter h together with a highpass filter g constitute a pair of conjugate quadrature filters, that is (Wei, 1998):

$$g(n) = -1^n h(n_1 + n_2 - n) \quad (2.15)$$

Where $[n_1, n_2]$ is the support of the filters h and g. The orthonormal condition is given by (Wei, 1998):

$$\delta_k = \sum_{n\in\mathbb{Z}} g(n)\psi(2t-n), \text{ for } k \in \mathbb{Z} \quad (2.16)$$

Where δ_k denotes the Kroneker delta function

A generalized coiflet wavelet of order N (denoted by $\psi_{l,t\hat{}}$) is defined, if for some $t\hat{} \in \mathbb{R}$, the wavelet (denoted by $\psi_{l,t\hat{}}$) and its scaling functions (denoted by $\emptyset_{l,t\hat{}}$) satisfy (Wei, 1998):

$$\int_{\mathbb{R}} (t-t\hat{})^p \emptyset_{l,t\hat{}}(t) = \delta_p \quad (2.17)$$

$$\int_{\mathbb{R}} t^p \psi_{l,t\hat{}}(t) = 0 \quad (2.18)$$

$for\ p = 0,1,\dots,l-1$

Where $t\hat{}$ = center of mass of the scaling function. The vanishing moment conditions in the above definitions are equivalent to (Wei, 1998):

$$\sum_{n\in\mathbb{Z}} n^p h_{l,t\hat{}}(n) = \sqrt{2}t\hat{}^p \quad (2.19)$$

$$\sum_{n\in\mathbb{Z}} (-1)^n n^p h_{l,t\hat{}}(n) = 0 \quad (2.20)$$

$for\ p = 0,1,\dots,l-1$

Coiflet wavelets system is constructed by iteratively solving equations (2.13), (2.14) and (2.15) to obtain the filter coefficients (Wei, 1998). In this research, these filter coefficients are

used for the transformation of image signal to decorrelate its pixel energy. The detail solutions to these equations are outside the scope of this research work. The coefficients for Coiflet wavelets of order L are shown in Appendix I.

2.2.1.5 *Wavelet Threshold Methods*

There are two basic types of wavelet threshold methods namely, hard threshold and soft threshold (Luo and Zhang, 2012). Different variants of these basic methods have been developed. On one hand, some of these variants are implemented either as fixed or global threshold, that is, applying uniform threshold values to all coefficients in the array. On the other hand, some of these threshold variants are implemented by applying variable threshold values at different scales (level-dependent). Several famous threshold schemes which are variants of these basic methods have been proposed (Moghtased-Azar et al., 2011). These include: Minimax threshold estimation by (Donoho and Johnstone, 1998) and Birge-Massart level-dependent threshold strategy (Birge and Massart 1997). The hard threshold removes coefficients below a specified threshold value. This can be implemented by (Mallat, 2009):

$$a_m = \begin{cases} 1, & if|x| \geq T \\ 0, & if|x| < T \end{cases} \quad (2.21)$$

$$\text{Hard Threshold} = a_m(x) = \begin{cases} x, & if|x| \geq T \\ 0, & if|x| < T \end{cases} \quad (2.22)$$

Where: $\mathrm{a_m} = \text{threshold function}$

$\mathrm{x} = \text{coefficient value}$

$\mathrm{T} = \text{Threshold value}$

The soft threshold modifies the wavelet coefficient values above and below the specified range of threshold values. A soft thresholding function is implemented by (Mallat, 2009):

$$0 \leq a_m(x) = \max(1 - \frac{T}{|x|}, 0) \leq 1 \quad (2.23)$$

$$\text{Soft Threshold} = a_m(x) = \begin{cases} x - T, & if|x| \geq T \\ x + T, & if|x| \leq -T \\ 0 \quad , & if|x| \leq T \end{cases} \quad (2.24)$$

Where:

a_m = threshold function

x = coefficient value

T = Threshold value

Typical plots of the hard and soft threshold functions are as depicted in Figure 2.6 (Luo and Zhang, 2012).

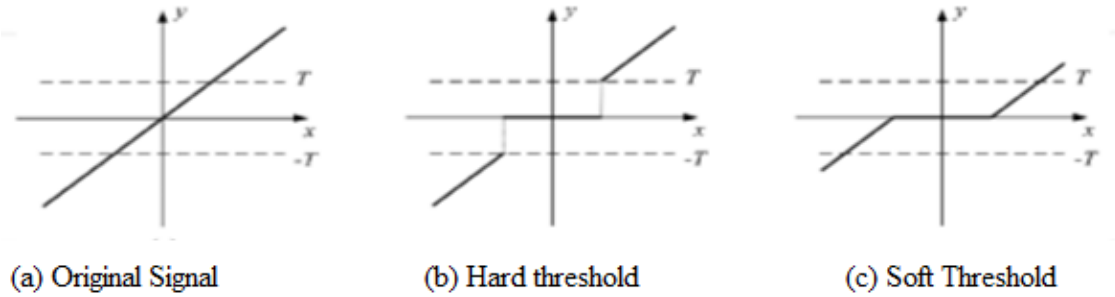


Figure 2.6: Plots of Threshold Functions

The hard thresholding removes the insignificant coefficients in the area where the original signal is regular. Compared with hard thresholding, the magnitude of coefficients with soft thresholding is a little smaller. The soft thresholding estimation reduces the insignificant coefficients at the area of discontinuities, but it also reduces the magnitude of significant coefficients that contribute to the image energy (Luo and Zhang, 2012). Put differently, hard thresholding can be described as the process of setting to zero the elements whose absolute values are lower than the threshold. Soft thresholding is an extension of hard thresholding, first setting to zero the elements whose absolute values are lower than the threshold, and then shrinking the non-zero coefficients.

2.2.1.6 *Level-Dependent Threshold Strategy*

In 2-dimensional (2D) image compression, two variants of hard thresholding strategies are commonly used. These include: fixed or global threshold and level-dependent threshold (Misiti et al., 1996). The global threshold is derived from an equal balance between the percentages of retained energy and number of zeros in a decomposed and energy de-correlated image signal and it is based on the analysis of the level-one detail coefficients. Level-dependent thresholds are more aggressive approach and are derived from Birge-Massart strategy (Birge and Massart, 1997). The Birge-Massart strategy is based on results on adaptive functional estimation in regression or density contexts. It uses level-dependent thresholds obtained by wavelet coefficients selection rule (Misiti et al., 1996; Luo and Zhang, 2012).

Given the decomposition level, j_0, and the sparcity parameter α, the length of coarsest approximation coefficient, m. The level-dependent threshold strategy is defined recursively in terms of the significant coefficient values, k_j as follows (Misiti et al., 1996):

$$k_j = \frac{m}{(j_0+1-j)^{\alpha}} \tag{2.25}$$

Typically, the sparsity parameter α is set to 1.5 for compression applications (Misiti et al., 1996; Birge and Massart, 1997). Level-dependent threshold approach is adopted for this research work because it applies variable threshold values at every level of image decompositions which makes it more efficient in compression algorithm than the other existing thresholding techniques such as the global threshold which applies fixed threshold values at all decomposition levels.

2.2.1.7 *Determination of Wavelet Decomposition Level*

The number of decomposition level in a wavelet transformation process is a crucial factor in image compression. This is because the resolution of the image to be cost-effectively stored or transmitted and the scale of its lowest approximation normally determine the number of decomposition levels in the transformation process. A suitable number of levels is selected

based on the nature of the signal. The number of bits in source image signal is in multiple of 2^n bits where n is the number of levels. Therefore, for 8 bits gray-scale fingerprint image, the decomposition level can be computed as follows: $8 = 2^n = 2^3$; n = 3.

2.2.1.8 *Computation of the Retained Energy (RE) and Number of Zeros (NZ)*

The percentage number of zero, NZ (%) and retained energy, RE (%) are measures of the extent of energy de-correlation of image signal and they are given by (Islam et al., 2012):

$$RE(\%) = \frac{100*(vn(ccd,2))^2}{(vn(original\ signal))^2} \quad (2.26)$$

$$NZ(\%) = \frac{100*(ZCD)}{number\ of\ coefficients} \quad (2.27)$$

Where:

vn= the vector norm,

ccd= the coefficients of the current decomposition and

zcd= the number of zeros of the current decomposition.

In this research work, the Daubechies and Coiflet wavelet bases were applied at different orders and at different decomposition levels for sparse representation of biometric fingerprint image. The performances of the wavelet bases in image approximation will be evaluated on the basis of NZ (%) ad RE (%) estimates.

2.2.2 Quantization

Quantization is the process of mapping a large set of input values to a smaller set, such as rounding values to some unit of precision (Salomon and Motta, 2010). The round-off or truncation error introduced by quantization is referred to as quantization error or quantization distortion and it is the difference between the actual input source value and quantized output value. Quantization is crucial to digital signal processing systems and it forms the core of lossy compression algorithms (Salomon and Motta, 2010; Sayood, 2012). In lossy compression application (Sayood, 2012), it is required to represent each symbol of the source data using a small number of codewords. The number of possible distinct source values is

generally much larger than the number of codewords used to represent them. A system or device which achieves this quantization is known as a quantizer (Sayood, 2012). Put differently, quantization simply means restricting a variable quantity in a continuous set of values to discrete set of values (Salomon, 2007). Whenever image data are captured, they have to be quantized in order to store and transmit them digitally. The natural image with its infinite level of detail is mapped to a finite number, depending on the desired accuracy of the operation. In information technology and digital communications, there is relentless quest for developing efficient quantization techniques for compression that deliver efficient results depending on the particular input data. Therefore, the quantization algorithm has to recognize and retain the important features of the input data and then discard the unimportant ones without any waste of quantization levels (Salomon, 2007).

In lossy image compression application, quantization is used in two ways, namely (Salomon, 2007):

i) If the data to be compressed is represented by large numbers, quantization is used to convert it to small numbers. This is because small numbers take less space than large numbers, thus compression is achieved by quantization;

ii) If the data to be compressed is analog, quantization is used to digitize it into a discrete set of small numbers. The smaller the numbers, the better the compression that is realized but also the greater the information loss.

It is therefore, of paramount importance to achieve a trade-off between the degree of data loss and compression ratio in the design of a lossy compression algorithm. In this research work, the source data for compression is a biometric fingerprint image. The fingerprint image is represented by a set of real numbers and these can either be uniformly quantized or non-uniformly quantized (Salomon, 2007). The quantized sequence where each symbol or pixel value appears in the source data with equal probability are uniformly quantized. However,

when the symbols in the source data are not uniformly distributed, the sequence of quantized values should be realized such that the sequence are distributed in the same fashion as the original source image and are non-uniformly quantized (Salomon, 2007).

In practice, a quantization process comprises two mappings, namely: encoder mapping and decoder mapping. The encoding process divides the range of values generated by the source output into a number of intervals. Each interval is represented by a distinct codeword. The encoder represents all the source output values that fall into a particular interval by the codeword representing that interval. Consequently, since there could be infinitely many distinct sample values that can fall in any given interval, the encoder mapping is irreversible (Salomon and Motta, 2010). This is because the codeword only reveals the interval to which a particular sample value belongs. It does not reveal which of the many values in the interval is the actual sample value (Sayood, 2012).

The task of a decoding process is to generate reconstruction values or representative levels. Since a codeword only represents the entire interval and there is no way of knowing which value in the interval was actually generated by the source, the decoder generates the values that in some sense best represents all the values in the interval. More importantly, the knowledge of the distribution of the source output values is crucial in generating the representative or reconstruction values for the decoding process. In the simplest form of this process, the mid-point value of the interval is often used to represent the interval. In other words, the construction of the intervals is crucial to the design of an encoder while the selection of reconstruction values is pertinent to the design of a decoder in a quantization system. This encoder-decoder pair is called quantizer. Therefore, in the design specification of a quantization system, it is required to know how to (Salomon and Motta, 2010; Sayood, 2012):

i) divide the source input range into intervals;

ii) assign codewords to these intervals; and

iii) find the representative or reconstruction values for these intervals.

These design requirements must satisfy the distortion and rate conditions. Distortion is defined as the average squared difference between the quantizer input and output. This is also known as the mean square quantization error (MSQE) or simply mean square error (MSE). Similarly, the rate of the quantizer is the average number of bits required to represent a single quantizer output. Therefore, the goal of a quantization system is to achieve the lowest distortion for a given rate, or the lowest rate for a given distortion level (Sayood, 2012).

Source image data for lossy compression often must be quantized for cost-effective storage through efficient quantization method. The source input image such as the grayscale biometric fingerprint image under consideration in this research work is represented in 8-bits and this must be quantized using less number of bits than the original 8-bits. This process results in some form of distortion which is known as quantization noise. Figure 2.7 shows a simple block diagram of the quantization process and the associated quantization noise.

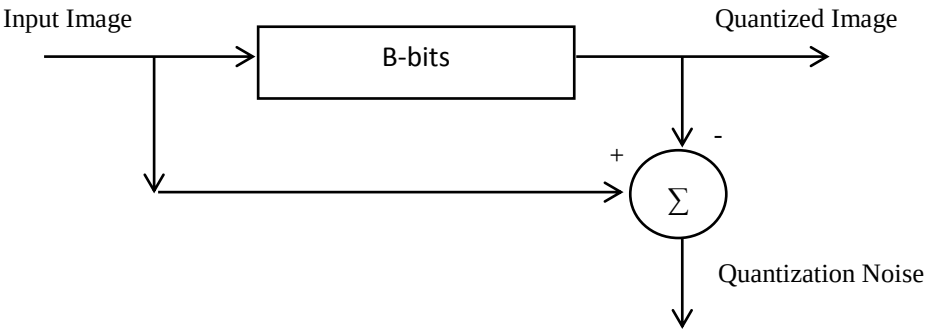


Figure 2.7: Block Diagram of Quantization Process and the Associated Noise (Sayood, 2012)

In simple terms, a quantization process represents an input source of random variables with large symbols values with a quantized output with smaller symbol values. The quantized output is a close approximation of the source input signal. The output symbol values are

predetermined values corresponding to the range of the input values and the number of bits allowed in the quantization process (Thyagarajan, 2011).

The design of a quantization system involves the specification of the decision intervals and corresponding output or representative levels and a mapping rule (Thyagarajan, 2011). Since there are many possibilities to partition the input range, the focus of the research is on the efficient quantization method that minimizes a certain criterion or cost function such as degree of distortion in the process. This Section discusses the quantization methods and relevant theoretical concepts. There are two types of quantization and these include: scalar and vector quantization.

2.2.2.1 *Vector Quantization*

In vector quantization, the source image is partitioned into equal-size blocks or vectors of pixels and the image encoder generates a list of blocks of the same size or lookup table and also known as codebook. Each image block is then compared to all the blocks of the codebook and is matched with the closest one. When the image block is smaller in size to its approximation on the codebook, compression is achieved (Salomon, 2007). It should be noted that the process of generating and maintaining a codebook makes the vector quantization process a complex one (Salomon, 2007). Consequently, the implementation of a compression algorithm based on vector quantization is flawed by its high computational cost and complexity. Therefore, adopting a quantization method with less complexity such as scalar quantization that does not require codebook generation is advantageous in realizing a compression algorithm that is amenable to simple implementation. This is one of the objectives of this research work.

2.2.2.2 *Scalar Quantization*

In scalar quantization, each input symbol of the source image is treated separately in producing the quantized output (Padmaja and Nirupama, 2012). A quantizer can be specified

by its input partitions or intervals and output levels (also called reconstruction values). If the input range is divided into levels of equal spacing, then the quantizer is termed as a uniform quantizer, and if not, it is termed as a non-uniform Quantizer (Padmaja and Nirupama, 2012).

2.2.2.3 *Uniform Scalar Quantization*

In a uniform quantizer, all source intervals or partitions are necessarily of the same size. A uniform quantizer has the following properties (Sayood, 2012):

i) The decision boundaries are spaced evenly;

ii) The reconstruction levels are also spaced evenly, with the spacing as the decision boundaries. The reconstruction levels are the midpoints of the intervals.

The constant spacing in a uniform quantizer's decision boundaries and reconstruction levels is referred to as the step size and it is denoted by delta Δ.

A quantizer that has zero as one of its reconstruction levels is known as a mid-tread quantizer. While a quantizer that does not have zero as one of its reconstruction levels is referred to as a mid-rise quantizer. In mid-rise quantizer, the number of reconstruction level L is even. Whereas in mid-tread quantizer, the number of reconstruction level L is odd. The mid-rise quantizer with even length L is often used for uniform quantization. Additionally, the following assumptions are often made to simplify the design of a uniform quantizer (Sayood, 2012):

i) Input source probability density function (PDF) is symmetric around the origin;

ii) The quantizer is symmetric;

iii) Reconstruction level is the mid-point of decision boundaries

The design of a uniform quantizer entails finding a step size Δ that will minimize the distortion for a given input source and number of decision levels L. Therefore, for a L-level uniform quantizer with a source input that has a uniform probability density function (PDF)

in the interval $[-X_{max}, X_{max}]$, there is the need to divide the input range into L equally sized intervals. The step size Δ is given by (Sayood, 2012):

$$\Delta = \frac{2\,X_{max}}{L} \tag{2.28}$$

Since the reconstruction level is assumed to be the mid-point of the decision boundaries, the distortion is expressed as follows (Sayood, 2012):

$$\sigma_q^2 = 2\sum_{j=1}^{L/2}\int_{(j-1)\Delta}^{j\Delta}\left(x - \frac{2j-1}{2}\Delta\right)^2 \frac{1}{2\,X_{max}}\,dx \tag{2.29}$$

Where:

L = number of decision levels in the quantization process;

$[-X_{max}, X_{max}]$ = Decision intervals or boundaries in the quantization process

Δ = Quantization step-size

Since the quantization error q is x - Q(x) and q lies in the interval $[-\frac{\Delta}{2}, \frac{\Delta}{2}]$. Therefore, the Mean Square Quantization Error (MSQE) is (Sayood, 2012):

$$\sigma_q^2 = \frac{1}{\Delta}\int_{-\Delta/2}^{\Delta/2} q^2\,dq = \frac{\Delta^2}{12} \tag{2.30}$$

If the source variance σ_x^2 in the interval $[-X_{max}, X_{max}]$ is given by (Sayood, 2012):

$$\sigma_x^2 = \frac{(2\,X_{max})^2}{12} \tag{2.31}$$

For the case of fixed-length codes, each code will be made of m bits where m is the number of codewords. The number of reconstruction levels L is given by (Sayood, 2012):

$$L = 2^m \tag{2.32}$$

Therefore, the Signal to Quantization Noise Ratio (SQNR) can be estimated as (Sayood, 2012):

$$\text{SQNR (dB)} = 10\log_{10}\frac{\sigma_x^2}{\sigma_q^2} \tag{2.33}$$

$$= 10\log_{10}\{\frac{(2\,X_{max})^2}{12}\ \text{x}\ \frac{12}{\Delta^2}\}$$

$$= 20\log_{10} 2^m = 6.02m\ dB$$

This means that for every additional bit in the quantizer, there is about 6.0 dB increase in the value of SQNR. This result is often used to obtain an indication of the maximum gain obtainable if the bit rate is increased. However, if the earlier assumptions stated for uniform quantizer are not valid. The above result is not also valid. Quite often in practice, the distributions of the source symbols that are encountered do not have uniform probability density function. Therefore, there is the need to use another approach of quantization for non-uniform sources. This approach is called Lloyd Max quantization. The objective is to determine the step size that will minimize distortion for a given value of L (Sayood, 2012).

2.2.2.4 *Standard Deviation and Variance*

The standard deviation measures the degree of dispersion from the mean of a statistical distribution. The variance is the deviation of each value from the mean of the statistical set. The standard deviation is the square root of the variance and is given by (Stroud, 2007):

$$\text{Standard Deviation, } \sigma = \sqrt{\frac{\sum(x-\bar{x})}{n}} \tag{2.34}$$

Where:

n = number of symbols in the set

x = variable value

$\bar{x} =$ mean of the source distribution

This statistical concept is used in the determination of image pixel distribution which is required for the application of non-uniform quantization schemes.

2.2.2.5 *Lloyd Max Non-Uniform Scalar Quantization*

Lloyd-Max quantization procedure defines an efficient approach to non-uniform quantization process. The basic idea of the Lloyd-Max quantization is to find the decision boundaries and reconstruction levels that minimize the mean square quantization error (MSQE). This approach solves the problem of finding the decision boundaries $\{b_j\}$ and the reconstruction

levels $\{y_j\}$ given N-level quantizer Q(x) on [a, b] so that the MSE given by Equation (2.35) can be minimized (Sayood, 2012).

$$MSQE = \sum_{j=1}^{N} \int_{b_{j-1}}^{b_j} \left(x - y_j\right)^2 f_x(x)dx \tag{2.35}$$

Where:

$f_x(x)$ = The Probability Density Function (PDF) of the source input, X

b_j = Decision boundary

y_j = Reconstruction level

N = Quantization level

Given the PDF of an input source x, the approach to designing the best non-uniform quantizer is to determine the values of the decision boundary and the reconstruction level that minimize the mean square quantization error (MSQE).

Setting the derivative of Equation 2.63 with respect to y_j to zero, and solving for y_j:

$$\frac{\partial(MSQE)}{\partial y_j} = 0$$

$$\sum_{j=1}^{N} \frac{\partial}{\partial y_j} [\int_{b_{j-1}}^{b_j} \left(x - y_j\right)^2 f_x(x)dx] = 0$$

Using the Chain Rule of Differentiation: $\frac{dy}{dx} = \frac{dy}{du} * \frac{du}{dx}$

$$-2 \int_{b_{j-1}}^{b_j} \left(x - y_j\right) f_x(x)dx = 0$$

$$\int_{b_{j-1}}^{b_j} [2y_j f_x(x)dx - 2x f_x(x)dx] = 0$$

$$2 \int_{b_{j-1}}^{b_j} y_j f_x(x)dx - 2 \int_{b_{j-1}}^{b_j} x f_x(x)dx = 0$$

$$2 \int_{b_{j-1}}^{b_j} y_j f_x(x)dx = 2 \int_{b_{j-1}}^{b_j} x f_x(x)dx \qquad j = 1, 2 \dots N$$

$$y_j = \frac{\int_{b_{j-1}}^{b_j} x f_x(x)dx}{\int_{b_{j-1}}^{b_j} f_x(x)dx} \tag{2.36}$$

The reconstruction point for each quantization interval is the centroid of the probability distribution of the interval. Taking the derivative of Equation 2.35 with respect to b_j and setting it equal to zero, an expression for b_j is obtained as follows:

$\frac{\partial(MSQE)}{\partial b_j} = 0$

$\frac{\partial}{\partial b_j}\{\int_{b_{j-1}}^{b_j}(x-y_j)^2 f_x(x)dx + \int_{b_j}^{b_{j+1}}(x-y_j)^2 f_x(x)dx\} = 0$

$(b_j - y_{j-1})^2 f_x(b_j) + (b_j - y_j)^2 f_x(b_j) = 0$

$(b_j - y_{j-1})^2 f_x(b_j) = -(b_j - y_j)^2 f_x(b_j)$

$(b_j - y_{j-1}) = \pm(b_j - y_j)$

$(b_j - y_{j-1}) = -(b_j - y_j)$

$b_j - y_{j-1} = -b_j + y_j$

$2b_j = y_j + y_{j-1}$

$b_j = \frac{y_j + y_{j-1}}{2}$ (2.37)

Lloyd-Max algorithm iteratively solve for the values of b_j and y_j that minimize the MSQE.

The properties of Lloyd-Max quantization system include (Sayood, 2012):

i) The mean values of the input and output of a Lloyd-Max quantizer are equal;

ii) The variance of the output is always less than or equal to the variance of the input of any given Lloyd Max quantizer;

iii) The mean squared quantization error σ_q^2 is given by (Sayood, 2012):

$\sigma_q^2 = \sigma_x^2 - \sum_{j=1}^{L} y_i^2 Prob[b_{j-1} \leq x < b_j]$ (2.38)

Where:

σ_x^2 is the variance of the quantizer input

$\sum_{j=1}^{L} y_i^2 Prob[b_{j-1} \leq x < b_j]$ is the second moment of the quantizer output (or variance if the input has zero mean).

iv) If N is the random variable corresponding to the quantization error, then for a given Lloyd-Max quantizer and X is the source sequence, the expectation of XN is given by (Sayood, 2012):

$$E[XN] = -\sigma_q^2 \tag{2.39}$$

v) For a given Lloyd-Max quantizer, the quantizer output and the quantizer noise are orthogonal (Sayood, 2012)

In summary, the Lloyd-Max algorithm quantizes a source input by first partitioning its symbols into N initial sets or intervals. It then calculates the average point or centroid of each interval. It constructs a new partition by associating each symbol with the closest centroid. The centroids are then re-calculated to obtain new partition. The algorithm iterates these steps until convergence is reached, that is when centroids no longer change. It should be noted that when the source input distribution is uniform, Lloyd-Max quantization can also be used to quantize the source and in this case, the decision intervals are all equal (Thyagarajan, 2011). The Lloyd-Max quantization algorithm is used in this research work.

2.2.3 Entropy Coding

A biometric fingerprint compression algorithm consists of two distinct structural blocks, namely: image encoder and decoder (Gonzalez and Woods, 2002). The encoding process is the forward process of compressing the source image while the decoding process is the reverse process of reconstructing the image signal from the encoded bit streams. The source image encoder is responsible for reducing redundancies in the source image. In the first stage of the source encoding process, the source image is transformed into a domain designed to reduce inter-pixel redundancies. This operation reduces the amount of data required to represent the image which directly results in data compression and the process is reversible (Gonzalez and Woods, 2002). Put differently, the representation of an image by a set of

transformed coefficients makes its inter-pixel redundancies more accessible for reduction to achieve compression. The second stage of the encoding process is the quantization stage where lossy compression is achieved which results in the reduction of the quality of quantized image output. The quantization operation at this stage reduces the psychovisual redundancy of the source image that is, it reduces redundant data that are not perceptible by human eyes and the process is irreversible (Gonzalez and Woods, 2002). In the third stage, of the source encoding process is the entropy coding which creates a fixed-length or variable-length codes to represent the quantizer output and maps the output to some codewords on the basis of the data source's probability distribution.

Source fingerprint image decoding process involves only two stages: a symbolic entropy decoder and an inverse discrete wavelet transform and they are performed in reverse order of the encoding process. Since the quantization process is a non-invertible operation and resulted in irreversible information loss, the idea of inverse quantization is not applicable in a lossy compression process. The final stage of a standard compression process is known as the entropy coding. The entropy coding is a lossless compression method which provides an effective mechanism to minimize coding redundancy, that is, the redundancy due to the coding process (Xing-Yuan et al., 2011). The primary objective of the entropy coding process is to decompose a data set into a sequence of events or symbols, then to encode the symbols using as few bits as possible. The idea is to assign short codewords to more probable symbols and longer codewords to less probable symbols (Howard and Vitter, 1994). Data compression can be achieved whenever some source symbols are more likely to occur than others. Entropy coding efficiently compresses the redundancy present in source input data and make its encoding rate tend to the entropy of the source data (Xing-Yuan et al., 2011). Upon completion of the three-stage compression process, the input source image has been processed to reduce each of the three redundancies present in the image.

2.2.3.1 *Definition of Terms*

For proper understanding of entropy coding process, it is important to define some basic terms such as symbols, alphabets, sequence, code values, etc.

An alphabet is a finite and nonempty set of elements. The length of an alphabet A is referred to as |A|. The elements $\{a_1, \ldots a_m\}$ of an alphabet are called symbols. A series of symbols, S = $\{s_1, s_2, s_3 \ldots, s_m\}$ or symbols s_i from an alphabet A is called sequence (Bodden et al., 2007). Let S = $\{s_1, s_2, s_3 \ldots, s_n\}$, a finite-length sequence with |S| = n over A = $\{a_1, \ldots a_m\}$. Also let $|S|_{a1}$ be the frequency of a_1 in S. Then the probability of a_1 (in S) is given by (Bodden et al., 2007):

$$P_{a_i} = \frac{|S|_{a_i}}{n} \tag{2.40}$$

Given an alphabet, A = $\{a_1, \ldots a_m\}$, A model, M is a function which maps a probability $P_M(a_i)$ to each symbol $a_i \in A$ and this is given by (Bodden et al., 2007):

$$M: A \rightarrow [0, 1): a_i \rightarrow P_M(a_i) \tag{2.41}$$

Where:

a_i = elements or symbols in the alphabet, A.

A = Source Alphabet

M = Model

$P_M(a_i)$ = Probability of symbol

It should be noted that the probability is often estimated as an approximation of the actual probability of the symbol, P_{a_i}.

2.2.3.2 *Entropy*

An information source produces a sequence of symbols, which are coded for storage or transmission and it is important to know how many bits are needed to represent (or code) each symbol of the source on an average (Thyagarajan, 2011). Suppose an information source generates a random sequence of symbols from a finite set of possible symbols

$\{a_1, a_2, \ldots, a_j\}$. The probability of the event that the source will produce symbols a_j is P (a_j). The average information per source output denoted by H(z) is given by (Salomon, 2007, Gonzalez and Woods, 2002):

$H(z) = -\sum_{j=1}^{N} P(a_j) log P(a_j)$, j = 1, 2, ... N (2.42)

Where:

H(z) = the uncertainty or entropy of the source

$P(a_j)$ = Probability that the source will produce a_j

a_j = Sequence of symbols

The entropy defines the average amount of information obtained by observing a single source output. As its magnitude increases, more uncertainty results and thus more information is associated with the source. If the source symbols are equiprobable, the entropy is maximized and the source provides the greatest possible average amount of information per source symbol. In the case of representing a source data as compactly as possible, Shannon's first theorem defines the minimum average codeword length per source symbol that can be achieved in a compression process (Gonzalez and Woods, 2002). According to Shannon theorem, a source data of n symbols can be averagely compressed down to nH bits but not further (Salomon, 2007). In other words, in a source data with n symbols, the entropy of the source depends on the individual probability P (a_j) and is largest when all n probabilities are equal (Salomon, 2007). This fact is used to define the redundancy R in the data. Redundancy R is defined as the difference between the largest possible entropy of a set of symbols and its actual entropy (Salomon, 2007). Thus:

$R = [-\sum_1^n PlogP] - [-\sum_1^n P(a_j) log P(a_j)]$ (2.43)

$R = logi + \sum_1^n P(a_j) log P(a_j)$ (2.44)

Equation 2.44 is the test for fully compressed data that is data with no redundancy. It depicts the condition where the source data represent the highest possible amount of information (Salomon, 2007).

2.2.3.3 *Probability Models*

In a statistical coding approach such as the entropy coding, a probability distribution model is needed to estimate the probability of occurrence of each possible symbol at each point in the coding process (Howard and Vitter, 1994). It only needs to provide an approximation of the probability distribution of the source data. Therefore, to obtain maximum compression of a source image, a good probability model and an efficient way of representing the model is crucial in the design of the compression algorithm. Put differently, to achieve a good image compression, the selection of a model that takes into account the structure or distribution of the source image symbols is of paramount importance.

The main advantage of arithmetic coding for statistical data compression is its inherent separation of coding and distribution modeling (Howard and Vitter, 1994). The separation of coding and modeling is important because it permits adaptability, that is the model structure and probability estimates can be changed adaptively. Therefore, an important method for characterizing a particular source is through the use of probability models. The knowledge of the probability model is important for the design of a number of compression schemes. Uniform, Gaussian, Laplacian, and Gamma distributions are four probability models commonly used in the design of compression systems.

2.2.3.4 *Uniform Distribution*

In a lossless compression, the uniform distribution model is referred to as the ignorance model (Sayood, 2012). This is because when there is no knowledge about the distribution of the source input, except possibly the range of values, the uniform distribution can be used to

model the source. The probability density function, $f_X(x)$ for a random variable uniformly distributed between *a* and *b* is given by (Sayood, 2012):

$$f_X(x) = \begin{cases} 1/b - a & for\ a \le x \le b \\ 0 & otherwise \end{cases} \quad (2.45)$$

Where:

x = random variable

a = minimum symbol value

b = maximum symbol value

2.2.3.5 *Gaussian Distribution*

The Gaussian distribution is one of the most commonly used probability models. The probability density function, $f_X(x)$ for a random variable with a Gaussian distribution is given by (Sayood, 2012):

$$f_X(x) = \frac{1}{\sqrt{\pi\sigma^2}} exp - \frac{(x-\mu)^2}{2\sigma^2} \quad (2.46)$$

Where:

$\sigma^2 =$ Variance of source distribution

$\mu =$ Mean of source distribution

x = random variable or source sequence

2.2.3.6 *Laplacian Distribution*

In image pixels distribution, there is a high degree of correlation among pixels. Therefore, a large number of the pixel-to-pixel differences will have values close to zero. In these situations, a Gaussian distribution is not a very close match to the data. A closer match is the Laplacian distribution, which is peaked at zero. The distribution function for a zero mean random variable with Laplacian distribution and variance is given by (Sayood, 2012):

$$f_X(x) = \frac{1}{\sqrt{2\sigma^2}} exp - \frac{-\sqrt{2}|x|}{\sigma} \quad (2.47)$$

Where:

σ^2 = Variance of source distribution

σ =Standard deviation of source distribution

μ = Mean of source distribution

x = source sequence

2.2.3.7 *Gamma Distribution*

This distribution is more peaked about zero than the Laplacian distribution. The distribution function for a Gamma distributed random variable with zero mean and variance is given by:

$$f_X(x) = \frac{\sqrt[4]{3}}{\sqrt{8\pi\sigma|x|}} exp - \frac{-\sqrt{3}|x|}{2\sigma} \quad (2.48)$$

Where:

σ =Standard deviation of source distribution

μ = Mean of source distribution

x = source sequence

The shapes of these four distributions, assuming a mean of zero and a variance of one, are shown in Figure 2.8.

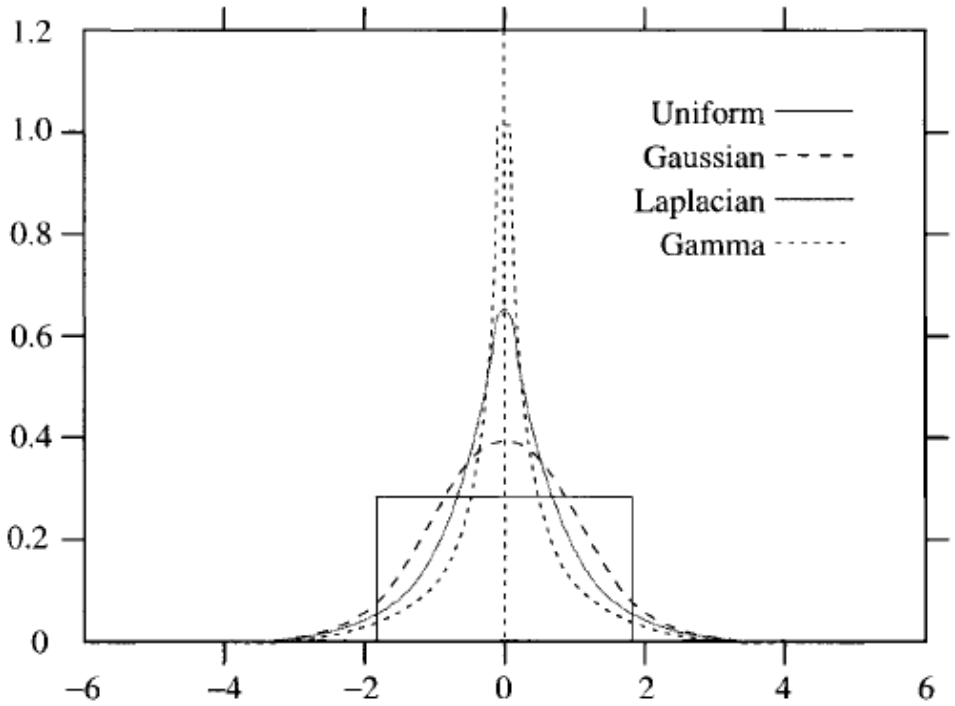


Figure 2.8: Uniform, Gaussian, Laplacian and Gamma Distributions (Sayood, 2012)

It should be noted that, when a source probability model is not readily available, as in the case of the symbol probabilities of real images, the stochastic distribution of such a source can be determined from its histogram (Thyagarajan, 2011). Therefore, in this research work, the probability distribution of the source fingerprint symbols was determined directly from the histogram plot of the source symbols using arithmetic entropy coding technique.

2.2.3.8 *Entropy Coding Methods*

The entropy coding for image compression is a lossless process because it does not result in any loss of data in the image signal (Gonzalez and Woods, 2002). Loss of data or distortion occurs essentially at the quantization stage and as a result of thresholding at the transformation stage. The commonly used entropy coding methods are the Huffman, Shannon-Fano and Arithmetic entropy coding methods.

2.2.3.9 *Huffman Coding*

The first step in Huffman coding which is the encoding step is used to create a series of source data reductions in descending order of probabilities of the occurrence or frequency of

the source data symbols and then combining the lowest probability symbols into a single symbol that replaces them in the next source reduction (Gonzalez and Woods, 2002). This results in a tree with a symbol at every branch from bottom to top. This is done in steps, where at each step, the two symbols with the smallest probabilities of occurrence are selected and added to the top of the partial tree, then deleted from the list and replaced with an auxiliary symbol representing the two symbols under consideration. When the list is reduced to just one auxiliary symbol representing the entire source data, the tree is completed as shown in Figure 2.9. The tree is then reversed to determine the codes of the source symbols (Salomon, 2007). The second step in Huffman's procedure which is the decoding step is aimed at coding each reduced source, starting with the smallest source and working back to the original source. After the code has been created, coding and decoding operations are accomplished in a lookup table manner (Gonzalez and Woods, 2002). However, the implementation of the lookup table adversely impacts on the computational efficiency of the encoding and decoding processes (Gonzalez and Woods, 2002).

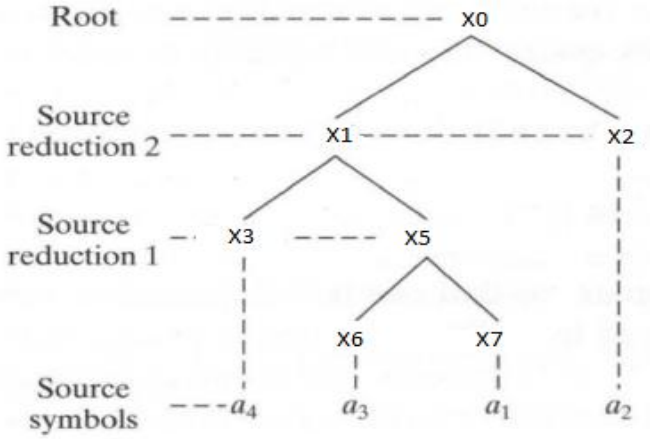


Figure 2.9: A Typical Huffman Binary Tree

2.2.3.10 *Shannon–Fano Coding Scheme*

Shannon-Fano coding is similar to Huffman coding and only differs in the way the binary tree is constructed which is in a 'top-down' manner (Salomon, 2007). The approach involves the development of a sorted list of source symbol's probability of occurrence. The list is

ordered such that the most frequently occurring symbols appear at the left and the least occurring at the right. The list is then divided into two parts, with the total frequency counts of the left part being as close as possible to the total on the right. The left part of the list is then assigned the binary digit 0, and the right part is assigned the digit 1. This means that the codes for the symbols in the first part will all start with 0, and the codes in the second part will all start with 1 so that a tree-like structure of the source's symbols is realized (Salomon, 2007).

2.2.3.11 *Arithmetic Coding*

Unlike the Huffman coding approach, arithmetic coding generates embedded codes that eliminate the need for generation of codebook or lookup table (Gonzalez and Woods, 2002). In this approach, an entire sequence of source image symbols is assigned a single arithmetic code word. The code word itself defines an interval of real numbers between 0 and 1 (Thyagarajan, 2011). As the number of symbols in the source data increases, the interval used to represent it becomes smaller and the number of information units required to represent the interval becomes larger. Each symbol of the source data reduces the size of the interval in accordance with its probability of occurrence (Gonzalez and Woods, 2002; Salomon, 2007). In other words, this method uses the probability of each symbol to narrow the interval as shown in Figure 2.10.

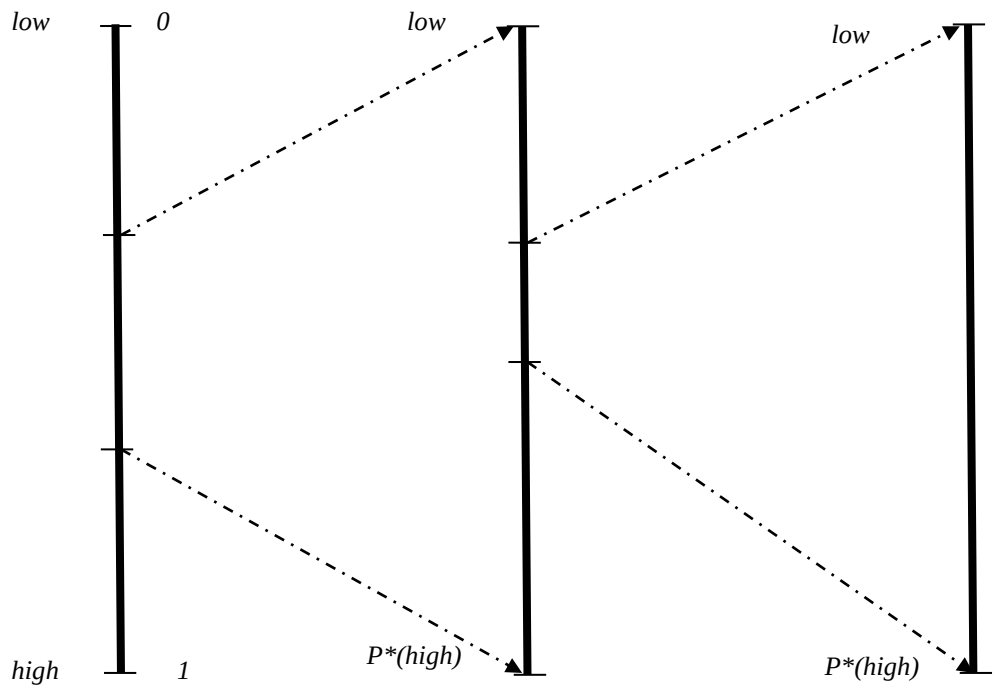


Figure 2.10: Typical Representation of the Arithmetic Coding Process

Arithmetic coding is a form of entropy encoding method which maps the source symbols to a variable-length number of bits. The main idea of this image coding method is to map the encoded source symbols to an interval of real number between 0 and 1 (Gonzalez and Woods, 2002; Salomon, 2007); then divide the interval into several intervals, each value on this interval can uniquely identify the encoded symbols. In arithmetic coding, the greater the probability of occurrence of the source symbol, the wider is the sub-interval, the shorter the corresponding codeword.

In summary, arithmetic coding involves performing arithmetic operations on the probabilities of source symbols as they are encoded. The probabilities are permitted to change from symbol to symbol, so long as the same probability is used in decoding a symbol as was used in encoding it. This allows the code to be adaptive and embedded. In addition, arithmetic coding is efficient even in the case of large sources, where conventional techniques such as Huffman coding become either inefficient or prohibitively complex (Howard and Vitter,

1994). It is for these reasons that arithmetic coding is used for entropy coding in this research work.

2.2.3.12 *Arithmetic Coding and Image Compression*

Compression applications employ a wide variety of techniques and have different degrees of complexity, but share certain processes in common. Figure 2.11 shows a process block for data compression. These processes depend on the data type, and the stages may be in different order or combination. The pre-processing stage often includes transformation and quantization processes. The next stage, source modeling is used to account for variations in the statistical properties of the data. It is responsible for gathering statistics and identifying data contexts that make the source models more accurate and reliable (Said, 2004). What most compression systems have in common is the fact that the final process is the entropy coding, which is the process of losslessly representing information in the most compact form (Said, 2004). In lossless compression scheme, entropy coding is responsible for the entire compression process. However, in lossy compression scheme, it complements what has been accomplished by previous stages such as transformation and quantization stages as it is the case in this research work.

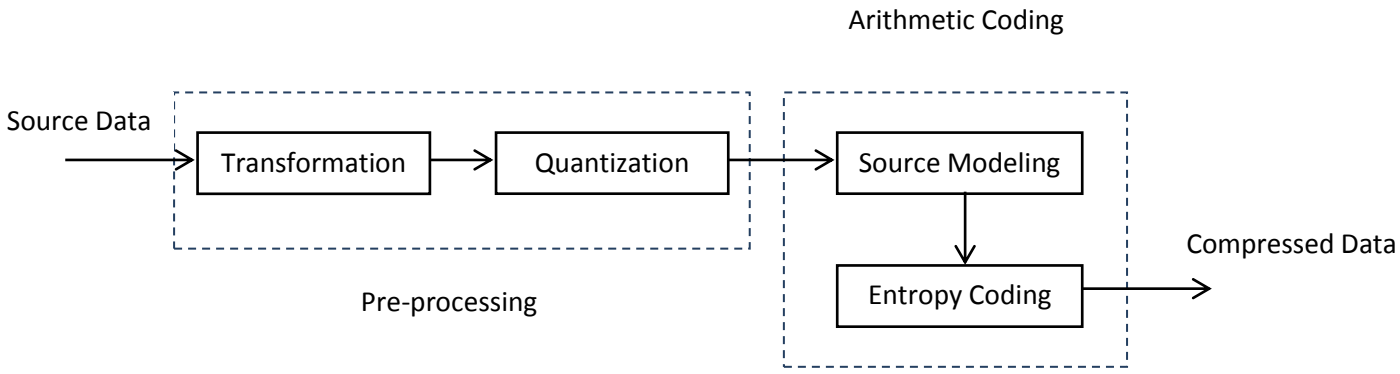


Figure 2.11: Arithmetic Entropy Coding as the Final Stage of a Compression Process

2.2.3.13 *Cumulative Distribution Function (CDF)*

Arithmetic coding essentially entails the determination of the cumulative distribution function (CDF) of the probability of a sequence of symbols. The length of the final subinterval in the arithmetic coding process is equal to the product of the cumulative probabilities of the individual source symbol. The cumulative distribution function (CDF) is given by (Howard and Vitter, 1994):

$$CDF(m_k) = \sum_{k=1}^{m-1} P_k \quad (2.49)$$

Where:

P_k = probability of individual symbol

m_k = Length of symbol

2.2.3.14 *Encoder and Decoder*

Arithmetic coding scheme codes one data symbol at a time, and assigns to each symbol a real-valued number of bits (Said, 2004). Code value or code word representation means coded messages mapped to real numbers in the interval [0, 1).The code value, v of a compressed data sequence is the real number with fractional digits equal to the sequence's symbols (Said, 2004). The code value representation can be used for any coding system and it provides a universal way to represent large amounts of information independent of the set of symbols used for the coding (Said, 2004).

Fundamentally, the arithmetic encoding process consists of creating a sequence of nested intervals in the form (Said, 2004):

$$\emptyset_k(S) = [\alpha_k, \ \beta_k), \quad k = 0, 1, \dots, N \quad (2.50)$$

Where:

$\emptyset_0(S)$ = encoded source sequence

$\alpha_k \ and \ \beta_k$ = real numbers such that $0 \leq \alpha_k \leq \alpha_{k+1}, and \ \beta_{k+1} \leq \beta_k \leq 1$

For simplicity, a new notation for arithmetic coding interval is represented in the form [b, l), where b is called the base or starting point of the interval, and l is the length of the interval. The relationship between the two interval notations is (Said, 2004):

$$[b, l) = [\alpha, \beta), \quad if\ b = \alpha\ and\ l = \beta - \alpha \tag{2.51}$$

The intervals used during the arithmetic coding process are defined by the set of recursive equations (Said, 2004):

$$\emptyset_0(S) = [b, l) = [0, 1) \tag{2.52}$$

$$\emptyset_0(S) = [b_k,\ \ l_k), = [b_{k-1} + \ c(s_k)l_{k-1},\ p(s_k)l_{k-1}),\ \ k = 1, 2, \dots, N \tag{2.53}$$

Where:

$\emptyset_0(S)$ = encoded source sequence

S = data symbol

p = symbol probability

c = cumulative distribution

b = interval base

l = interval length

The properties of intervals guarantee that $0 \leq b_k \leq b_{k+1} < 1, and\ 0 < l_{k+1} \leq l_k \leq 1$. The encoding process defined by Equations 2.51 and 2.52 is called Elias coding (Said, 2004). Equation 2.53 depicts that the size of the symbol's subinterval $[b_k,\ \ l_k)$ is proportional to the estimated probability of occurrence *p* of the symbol in accordance to the cumulative distribution function *c* of the source input. The final interval is then assigned a codeword in bits.

The minimum number of bits required for representing the decoded sequence represented by the code value v' is given by (Said, 2004):

$$B_{min} = [-log_2(l_N)]\ \ bits \tag{2.54}$$

Where:

l_N = interval length

In arithmetic coding, to decode the source sequence, the encoding process is reversed. The decoder input set is determined to restore the original sequence. In other words, the decoded sequence is determined solely by the code value v' used to represent the encoded sequence. Therefore, the decoded sequence is represented as (Said, 2004; Bodden et al, 2007):

$$\hat{S}(v') = \{\hat{s}_1(v'), \hat{s}_2(v'), \dots, \hat{s}_N(v')\} \quad (2.55)$$

In the decoding process, code value v' is the set of codewords used for decoding the correct symbol sequence $\hat{S}(v') = S$ of the encoded source input. The decoding process recovers the data symbols in the same sequence that they were encoded. The numerical solution to the decoding process is determined by defining a sequence of code values $\{\tilde{v}_1, \tilde{v}_2, \dots, \tilde{v}_N\}$. Starting with $\tilde{v}_1 = \tilde{v}$, $\hat{s}_k$and $\tilde{v}_k$ are sequentially determined and $\tilde{v}_{k+1}$ is computed from $\hat{s}_k$ and $\tilde{v}_k$. Equations (2.89) to (2.92) define the numerical solution for decoding process (Said, 2004):

$$\tilde{v}_1 = \tilde{v} \quad (2.56)$$

$$\hat{s}_k(v') = s \; for \; c(s) \leq c(s+1), \;\; k = 1, 2, \dots, N. \quad (2.57)$$

$$\tilde{v}_{k+1} = \frac{\tilde{v}_k - c(\hat{s}_k(v'))}{p(\hat{s}_k(v'))}, \quad k = 1, 2, \dots, N-1. \quad (2.58)$$

$$\tilde{v}_k = \frac{v' - b_{k-1}}{l_{k-1}} \quad (2.59)$$

Where:

$\hat{S}(v') =$ Decoded symbol sequence

$\tilde{v}_k =$ Sequence of decoded code values

$v' =$ Encoded code values of the source input

S = Numerical values of symbols that satisfy the inequality, $c(s) \leq c(s+1)$

In summary, the decoding process recovers the decoded sequence $\hat{S}(v')$ by sequentially determining the sequence of decoded code values $\tilde{v}_k$ from the encoded code values v' of the source input.

2.2.3.15 *Entropy Coding Efficiency*

Source coding efficiency is defined as the ratio (in percentage) of the entropy of a source to the average code length and it is given by (Pu, 2006):

$$E(S,L) = \frac{H(S)}{l'(S,L)} * 100\% \quad (2.60)$$

Where:

$E(S,L) =$ Coding Efficiency

$H(S) =$ Source entropy

$l'(S,L) =$ Average code length

The codeword is the number of bits used to represent a source's symbol in an entropy coding process. Given the length of codewords $L = (l_1, l_2, \dots, l_n)$ resulting from an entropy coding process of a source $S = (s_1, s_2, \dots, s_n)$, the average length of the code is given by (Pu, 2006):

$$l' = \sum_{j=1}^{n} s_j l_j \quad (2.61)$$

Where:

$s_j =$ Symbol's probability of occurrence

$l_j =$ Length of codeword

The entropy provides a theoretical bound of the minimum number of bits that can be used to represent source data. Given the source $S = (s_1, s_2, \dots, s_n)$, the source entropy is given by (Pu, 2006):

$$H(S) = \sum_{j=1}^{n} s_j log_2 s_j \quad (2.62)$$

Where:

$s_j =$ Symbol's probability of occurrence

Information theory states that the best a lossless symbolic compression scheme can achieve is to encode a source with an average number of bits equal to the entropy of the source. However, when the difference between the entropy and the average length of the code increases, the code efficiency decreases. It follows therefore that an entropy coding process is optimal, if the code efficiency reaches 100%. In other words, a code is optimal if the average length of the codewords equals the entropy of the source (Pu, 2006).

2.2.4 Validation and Performance Measures

Two measures will be used to validate the perceptual quality of the compressed fingerprint images, namely: Compression Ratio (CR) and Peak Signal to Noise Ratio (PSNR). The PSNR quality metric will be used to measure the image quality resulting from the compression process.

i) Compression Ratio: The compression ratio (CR) or C_r is used as a performance measure to assess the efficiency of a compression algorithm and this is given by (Xing-Yuan et al, 2011):

$$C_r = \frac{\sum_{i=1}^{N}\sum_{j=1}^{N} r_b(i,j)}{\sum_{i=1}^{N}\sum_{j=1}^{N} r_c(i,j)} \tag{2.63}$$

Where:

r_b = bits per pixel of original image

r_c = bits per pixel of compressed image

ii) The Peak Signal to Noise Ratio (PSNR): It represents a measure of the peak error and is expressed in decibels. The higher the *PSNR value*, the closer the quality of the compressed or reconstructed image to the original source image. Typical values for lossy compression of an image are between 30 and 50 dB and when the *PSNR* is greater

than 40 dB, then the two images are indistinguishable (Strang and Nguyen, 1996). The PSNR is defined by (Strang and Nguyen, 1996):

$$PSNR = 20 \log \left(\frac{2^B - 1}{MSE}\right) \quad (2.64)$$

For an 8-bit grayscale image, B = 8:

$$PSNR = 20 \log \left(\frac{255}{MSE}\right) \quad (2.65)$$

Where:

B = bit depth

MSE = means square error

The Mean Square Error (MSE): It represents the mean squared error between the compressed and the original image. The lower the MSE value, the lower the distortion or degradation incurred in the compression process. For a source image $X_{i,j}$ of size N x M pixels and a reconstructed/processed image $Y_{i,j}$ also of size N x M, the Mean Square Error, MSE, is given by (Strang and Nguyen, 1996):

$$MSE = \frac{1}{MN} \sum_{j}^{N-1} \sum_{i}^{M-1} (X_{i,j} - Y_{i,j})^2 \quad (2.66)$$

Where:

$X_{i,j}$ = Source input or original image sequence

$Y_{i,j}$ = output sequence or compressed image sequence

Equation 2.66 describes the amount of error or distortion that occurs due to the difference between the original input image sequence and the compressed output in the compression system.

2.3 Review of Similar Works

The approach adopted for a critical review of similar research works is based on the compliance with the requirements of a standard compression scheme which involved three processes, namely: transformation, quantization and entropy coding. In addition, the review is based on the performance metrics achieved which includes: compression ratio (CR) and peak signal to noise ratio (PSNR).

Hsin et al. (2007) proposed a hybrid algorithm using SPIHT and EBC (embedded block coding) to encode low frequency and high frequency wavelet coefficients, respectively; the intermediate coding results of low frequency coefficients are used to facilitate the coding operation of high frequency coefficients. Experimental results showed that the coding performance can be significantly improved by the hybrid SPIHT-EBC algorithm. According to Hsin et al (2007), one of the advantages of hybrid SPIHT-EBC coding is that the well-defined hierarchical structure across wavelet subbands and energy clustering within each wavelet subband can be taken into account to facilitate the image compression task. The SPIHT coding method does not perform well in the preservation of an acceptable quality of biometric fingerprint patterns.

Chang et al. (2007) proposed two methods to improve computational efficiency of parallel context formation of EBCOT, that is, sample-parallel pass-type context formations (SPPD) and column-based pass-parallel coding (CBPC) methods. In this technique, DWT based on Daubeches Lifting scheme was used to decompose the image and then uniformly quantized. The resulting code blocks were encoded with the EBCOT methods based arithmetic coder. The EBCOT is a two-tiered architecture. Tier-1 is a context-based adaptive arithmetic coder, which is composed of a context formation (CF) engine and a binary arithmetic coder (BAC). Tier-2 is responsible for rate-distortion optimization and bit stream formation. The proposed methods achieved greater computation efficiency. However, EBCOT algorithm is

computationally complex. In addition, the work did not estimate the compression ratio and PSNR value and as such, there is no basis for determining the extent of compression.

Muhsen et al. (2008) conducted a comparative analysis of a re-quantization codebook method and vector quantization for fingerprint image compression. The transformation and the entropy coding stages were excluded in the implementation and only the quantization method of the compression scheme was used. While the generation of codebook added to the cost of computation, the compression ratio to estimate the performance of the process was not determined. In addition, the work fell short of the requirement of a standard compression scheme.

Rawat and Meher (2009) proposed a hybrid scheme combining Kohonen's Self Organizing Feature Map (SOFM) based Vector Quantization (VQ) coding and Set Partitioning in Hierarchical Trees (SPIHT) coding for effective compression of images. First, the input image is decomposed using biorthogonal wavelet transform. Subsequently, the decomposed image is compressed using SPIHT encoding, which results in bit stream. The resulted bit stream is then fed to the SOFM based VQ coding for further compression. The SOFM algorithm generates a codebook corresponding to the input bit stream and based on the generated codebook compression is achieved. The reconstructed image quality achieved after decoding is of desirable quality. However, the vector quantization method used in this algorithm is very complex and require the generation of codebook which can increase computational cost.

Ashok et al. (2010) proposed an algorithm for fingerprint image compression based on wave atoms decomposition for sparse representation of fingerprint images. In the proposed method, 2-dimentional wave atoms decomposition was applied on the original image in order to efficiently capture coherence of the fingerprint images along and across the oscillations. Wave atoms decomposed coefficients with values close to zero are discarded without

significant degradation in image quality. An appropriate global threshold is used in the image transformation process. The significance map is divided into non-overlapping blocks of 4x4 and they are quantized using K-means vector quantization scheme. The significant coefficients are quantized using a uniform scalar quantizer. Quantized significance map and significant coefficients are encoded using an arithmetic entropy scheme. The model was reported to achieve better peak signal to noise ratio (PSNR) than the wavelet scalar quantization (WSQ) standard. However, the global thresholding strategy used for image de-correlation is not efficient. Level-dependent threshold strategy is more robust and efficient.

Zhao and Wang (2009) introduced a Wavelet-based Contourlet transform (WBCT). The compression algorithm used uniform scalar quantization and arithmetic entropy coding scheme. The 2D wavelet transform was implemented using Biorthogonal wavelet filterbank. The PSNR value obtained for the method was 25.90dB. However, the compression ratio was not estimated and therefore, there is no basis for determining the extent of compression in the process.

Haddad et al. (2009) conducted a comparative analysis of different wavelet bases to determine the most appropriate transform for fingerprint image compression. The Curvelet wavelet and the wave atom were analysed. Quantization and entropy coding processes were not implemented in the scheme. Therefore, the compression process fell short of the requirement of a standard compression scheme.

Kumar et al. (2010) proposed a Biorthogonal wavelet-based fingerprint compression algorithm with SPHIT encoder. The set partitioning in hierarchical trees (SPIHT) is a modified version of the embedded zerotree wavelet method. It used DWT decomposition of image signal using biorthogonal and orthogonal properties of wavelets. The method achieved a compression ratio of 20:1. However, the PSNR value which is a measure of the quality of the compressed image resulting from the algorithm implementation was not estimated for the

compression ratio achieved. There was no basis for determining the quality of the compressed image as against the compression ratio achieved. Additionally, the quantization process was not implemented in the scheme which could further improve the performance of the compression algorithm.

Krishnaiah et al. (2011) proposed 5/3 discrete wavelet transform (DWT) for fingerprint image compression and reconstruction with the SPIHT algorithm. Experimental results were obtained using 9/7 and 5/3 wavelet transforms for different types of fingerprint images. The experimental results showed that the proposed method, that is, 5/3 wavelet transforms approach consistently outperforms 9/7 wavelet transforms approach on fingerprint images for lossless image compression, in terms of compression ratio (CR), Mean square error (MSE), Peak Signal to noise ratio (PSNR), Encoding time, decoding time and transforming time or decomposition time. This is a lossless compression algorithm and the compression ratio that can be achieved lossless technique is less than 5:1.

Mushen et al. (2011) proposed a methodology for lossy fingerprint compression using wavelet and optimal re-quantization approach. 9/7 wavelet transform was used to decompose the image to form coefficients which were optimally re-quantized using generated codebook. The output stream of coding symbols were entropy coded using Huffman and run-length encoding scheme. However, in this scheme the generation of codebook from the vector quantization process required additional computational resources for implementation.

Gangwar (2012) presented a fingerprint image compression technique using the Haar wavelet transform for image decomposition. Elimination of redundancies was presented to achieve reduced computation and storage costs. However, this technique fell below international image compression standard as it excluded vital stages of quantization and entropy coding of a typical image encoder.

Shakhakarmi (2012) performed an experiment that employed different wavelets to deploy cascaded filter banks to achieve fingerprint image decomposition and reconstruction. Significantly, the multiscale analysis of 2D fingerprint image at stage-4, produced a better result for wavelet filters in comparison to the result obtained with Fast Fourier Transform (FFT) and Discrete Cosine Transform (DCT) based filters. The inefficiency of FFT and DCT for image compression has been established as they produced blocking artifacts. A comparative study with these techniques is not rigorous enough for accurate judgement.

Libert et al. (2012) conducted a study to compare the effects of WSQ and JPEG 2000 compression on 500 pixel per inch (ppi) fingerprint imagery at a typical operational compression rate of 0.55 bpp (bits per pixel), corresponding to an effective compression ratio of approximately 15:1. More significantly, the study revealed that while JPEG 2000 exhibited a small advantage over WSQ with respect to PSNR comparison, frequency spectrum comparison shows that WSQ is better tuned to the preservation of fingerprint features than JPEG 2000. However, JPEG 2000 is considerably more stable over multiple compression cycles. WSQ compression ratio is limited to 15:1 while JPEG2000 does not adequately preserve the minutia features of fingerprint at compression ratio higher than 20:1.

Shanavaz and Mythili (2012) presented a technique for evolution of wavelet lifting coefficients for fingerprint image compression to enhance computation efficiency. In this work wavelets were evolved with resized and cropped images. Cropped images performed better in wavelet lifting coefficients. It adopted the SPIHT coder which required the use of codebooks and produced blurring effect in fingerprint images at higher compression ratio.

Islam et al. (2012) investigated Coiflet-type wavelet-based fingerprint image decomposition using wavelet packets. For all Coiflet-type wavelets, different global thresholding values were used at constant decomposition level 3. However, the work did not implement the quantization and entropy coding stages which could further improve performance. In

addition, the compression ratio of the method was not estimated to determine the level of achievable compression. In addition, global threshold strategy was adopted for wavelet analysis and it is less robust compared to level dependent threshold strategy.

Singla et al. (2013) conducted a comparative study of wavelet-based compression on medical images. The performances of Haar, Daubechies, Coiflet and Biorthogonal wavelets transformation implemented with global thresholds were compared. The comparison was based on different parameters to measure the image quality and the methods' compression ratio. The study showed that Coiflets transform produced higher compression rate for ultrasound and mammography images compared to other wavelets using PSNR quality measure. However, the source data used for the analysis were the medical ultra-sound images. Hence, there is no need for the concern for the preservation of biometric features as in the case of a fingerprint image compression.

Selvakumarasamy et al. (2013) conducted a performance analysis of Biorthogonal, Symlet, Daubechies and Coiflet wavelets for lossless compression implementing only the transformation process. A range of compression ratios of 1:0.8 to 1:0.63 were achieved. However, the entropy coding process required for the standard lossless compression system was not implemented. Additionally, lossy compression scheme is a better approach to achieving a much higher compression ratio.

Srivastava et al. (2014) proposed a hybrid quantization scheme which involved uniform quantization of approximation coefficients and non-uniform quantization of the detail coefficients of an image array in a lossy JPEG2000 standard. However, the hybrid scheme was not implemented in the work but only the experimental results of non-uniform quantization of the detail coefficients of colour images, namely; Lenna, Baboon and Pepper images were presented. Hence, there was no consideration of the need to preserve biometric

features as required in the compression of fingerprint images. The work did not also implement the entropy coding of a standard compression scheme.

Rathee and Vij (2014) proposed Discrete Haar Wavelet Transform (DHWT) implemented with global threshold strategy for image compression. The source data used for the analysis was the MATLAB experimental Lena image. Hence, there was no need for the concern for the preservation of biometric features as in the case of a fingerprint image compression. In addition, the work did not implement the quantization and entropy coding stages of a standard lossy compression scheme which can further improve performance. Therefore, the work fell short of the requirement of a standard compression scheme. Besides, there was no performance measurement in terms of compression ratio to estimate the level of achievable compression.

Deepa and Girisha (2014) presented a hybrid wavelet using two orthogonal transform to decompose and compress a color (RGB). However, only the transformation stage of a standard compression scheme was implemented. Quantization and entropy coding stages were not implemented in this work which could further improve the performance of the proposed compression scheme. The resulting compression process was lossless, hence, the low compression ratio.

Table 2.3 summarizes the reviewed similar works done on wavelet based image compression methods with their critical analysis.

Table 2.3: Review Summary of Similar Works Based on Wavelet Compression Methods

Reference	Transformation	Quantization	Coding scheme	Critique
Hsin et al., 2007	Wavelet transform	Not implemented	Context-based EBCOT encoding based on arithmetic coder	High computation cost due to EBCOT. Compression ratio was not estimated to determine the level of achievable compression. The work is not compliant with standard compression scheme requirement.
Chang et al., 2007	DWT based Daubechies wavelet	Uniform scalar quantization	EBCOT encoding based on arithmetic coder	High computation cost of complex algorithm due to EBCOT
Mushen et al., 2008	Not covered	Vector quantization	Not implemented	High computation cost due to codebook resulting from vector quantization. Compression ratio was not estimated to determine the level of achievable compression. The work is not compliant with standard compression scheme requirement.
Rawat et al., 2009	Biorthogonal wavelet transform	SOFM based vector quantization with code book	SPIHT based on Embedded Zerotree encoder	Complex algorithm due to vector quantization
Haddad et al (2009)	Curvelet and wave atom	Not implemented	Not implemented	The compression process fell short of the requirement of a standard compression scheme.
Zhao and Wang (2009)	Contourlet transform	Uniform scalar quantization	Arithmetic coding	The compression ratio was not estimated and therefore, there is no basis for determining the extent of compression in the process.
Kumar et al., 2010	DWT based on Biorthogonal wavelets	Not implemented	SPIHT based on Embedded Zerotree encoder	The compression ratio was not estimated and therefore, there is no basis for determining the extent of compression in the process. Poor preservation of biometric features by SPIHT
Ashok et al., 2010	Wave atom decomposition	Vector quantization	Arithmetic entropy coding	Complex algorithm due to codebook generated from vector quantization process
Krishnaiah et al., 2011	5/3 wavelet transform	Not implemented	SPIHT based on Embedded Zerotree encoder	Low compression ratio due lossless scheme employed.
Muhsen et al., 2011	9/7 Wavelet filter	Vector quantization and re-quantization with codebook	Run-length encoding	High computation cost and complex algorithm due to codebook generation resulting from vector quantization is a challenge
Gangwar, 2012	Haar wavelet	Not implemented	Not implemented	Failed to comply with compression standards
Shanavaz et al., 2012	Daubechies Wavelet lifting	Not implemented	SPIHT based on Embedded Zerotree encoder	No quantization scheme was implemented in the compression method.

Table 2.3: Review Summary of Similar Works Based on Wavelet Compression Methods

Shakhakarmi, 2012	Wavelets multiscale analysis	Not implemented	Not implemented	The work did not implement the quantization and entropy coding stages of a standard lossy compression scheme.
Libert et al., 2012	WSQ Versus JPEG2000	Uniform Scalar Quantization	Arithmetic, Huffman	JPEG2000 achieved poor biometric quality. WSQ is limited in compression ratio
Islam et al., 2012	Coiflet wavelet implemented with global threshold strategy	Not implemented	Not implemented	Compression ratio was not estimated to determine the level of achievable compression. The work is not compliant with standard compression scheme requirement.
Singla and Sharma, 2013	Haar, Coiflet, Daubechies, Biorthogonal wavelets	Not implemented	Not implemented	Wavelets were implemented with global threshold strategy.
Selvakumarasamy, et al (2013)	Biorthogonal, Symlet, Daubechies and Coiflet wavelets for lossless compression	Not implemented	Not implemented	The entropy coding process of a standard lossless compression system was not implemented. Lossy compression scheme is a better approach to achieving a much higher compression ratio.
Srivastava et al (2014)	Daubechies 9/7	Hybrid quantization scheme	Not implemented	The hybrid scheme was not implemented in the work but only the experimental results of non-uniform quantization of the detail coefficients of colour images were presented. Hence, there was no consideration of the need to preserve biometric features as required in the compression of fingerprint images.
Rathee and Vij (2014)	Discrete Haar Wavelet	Not implemented	Not implemented	Wavelets were implemented with global threshold strategy.
Deepa and Girisha (2014)	Orthogonal Hybrid Wavelet Transform	Not implemented	Not implemented	The work did not implement the quantization and entropy coding stages of a standard lossy compression scheme which can further improve performance. The resulting compression process is lossless, hence its low compression ratio.

In conclusion, the reviewed works employed wavelet filter banks such as Haar, Symlet, Wave Atom, Coiflet implemented with global thresholding, Daubechies and their variants for image decomposition and these have been identified to possess limited properties that can achieve improved image compression. Vector quantization and uniform scalar quantization schemes were implemented in the existing compression schemes which accounted for their failure to preserve biometric fingerprint features at compression ratio higher than 15:1. Additionally,

the existing techniques generated codebooks or lookup tables for image coding and this invariably increased the complexity of the algorithms and hence, increased computation or implementation cost. Therefore, the proposed biometric fingerprint image compression algorithm will employ Coiflet wavelet implemented with level-dependent threshold strategy for image transformation as against the global threshold strategy used in the existing wavelet-based image transformation scheme to achieve improved energy de-correlation for a more efficient compression process. In addition, non-uniform quantization scheme will be used at the quantization stage as against the uniform scalar quantization methods employed in the existing schemes. This combination of techniques is the improved approach that is proposed for biometric fingerprint compression in this work.

CHAPTER THREE

MATERIALS AND METHODS

3.1 Introduction

This chapter discusses the various materials and methods employed for this research work.

3.2 Research Materials

For the proposed source coding technique to be developed in this work, raw-bits, uncompressed fingerprint dataset is required. Hence, the need to employ the use of the National Institute of Technology and Standards (NIST) fingerprint datasets. The NIST dataset were used for two reasons:

i) Commercially available fingerprint scanners could not be used to collect rawbits and uncompressed data for this research work because they can only acquire encoded data based on the existing compression algorithm;

ii) The performances of the existing fingerprint compression algorithms were based on the NIST dataset and as such it is reasonable to use the same dataset to assess the performance of the proposed algorithm.

The software analysis tool employed for the experiments in this work is the MATLAB version 2007. The MATLAB toolboxes that were used include; wavelet and image processing toolboxes.

3.3 Research Methods

The research methodologies adopted in this work are in five stages as follows:

Stage One: 2-D Discrete Wavelet Transform (DWT) of source fingerprint image.

Stage Two: Non-uniform scalar quantization using Lloyd-Max procedure

Stage Three: Arithmetic entropy coding to generate image bit-streams

Stage Four: Reconstruction of Fingerprint image from bit-stream

Stage Five: Validation and evaluation compression algorithm using Compression Ratio (CR) and Peak Signal to Noise Ratio (PSNR) performance metrics

3.3.1 2-D Discrete Wavelet Transform (DWT) of Source Fingerprint Image

The first stage of the research methodology is the 2-dimensional discrete wavelet transform (2D-DWT) of source fingerprint image. This stage is further broken down as follows:

i) Acquisition of source fingerprint image data;

ii) Computation of analysis wavelet coefficients up to three-level of decomposition;

iii) Application of level-dependent threshold to achieve sparse representation of source fingerprint image;

iv) Reconstruction of source fingerprint image;

v) Computation of percentage retained energy, RE (%) and number of zero, NZ (%);

vi) Comparative analysis of the results for orthogonal Daubechies and Coiflet wavelets based fingerprint image approximation, as well as, global and level-dependent threshold strategies.

3.3.1.1 *Acquisition of Source Fingerprint Image Data*

Source fingerprint dataset was obtained from the test database of the National Institute of Standards and Technology, USA (NIST, 2013). The NIST fingerprint dataset is the international standard test fingerprint for compression algorithm development. It should be noted that the performance of existing compression standards is based on the NIST fingerprint dataset. The proposed fingerprint compression algorithm is an embedded system that is deployable in a fingerprint acquisition device also known as fingerprint scanner. Therefore, the proposed algorithm is designed to compress fingerprints one at a time at the point of data acquisition. It is designed to be proactive, that is to achieve compression at the point of acquisition. It should be noted therefore that, fingerprint images consist of huge number of pixel values. For instance, an uncompressed fingerprint of 589 x 605 image

resolution from NIST dataset contains 356,545 pixels or symbols. These pixel values are what the proposed algorithm manipulates to achieve compression.

3.3.1.2 *Wavelet Decomposition of Source Fingerprint Image*

A wavelet filter analysis flowchart was developed based on the analysis parameters shown in Table 3.1.The analysis flowchart is depicted in Figure 3.1.

Table 3.1: Wavelet Analysis Filter Design Parameters

S/N	Design Parameters	Values
1.	Wavelet type	Daubechies, Coiflets wavelets
2.	Filter Order or Number of Vanishing Moments	Order 2, 3, 4
3.	Decomposition levels	Levels 1, 2 and 3
4.	Threshold Strategy	Level-dependent/Birge-Massart thresholds

Based on the analysis flowchart in Figure 3.1, MATLAB script was written to implement the various stages of the wavelet analysis of fingerprint image as shown in Appendix I. First, the input variables were defined for the source fingerprint image (cmp00001.pgm). Wavelet decomposition levels (1, 2 and 3), as well as, wavelet bases with their associated vanishing moments (Daubechies: db2, db3, db4 and Coiflet: coif2, coif3, coif4) were also defined. The justification for the three-level decomposition is that since the input image signal is in a multiple of 2^n bits where n is the number of levels. Thus, for 8 bits grayscale fingerprint image, the decomposition level possible is: $8 = 2^n = 2^3$; n = 3. Therefore, at decomposition level three, the 8-bit fingerprint image is completely decomposed.

The 2-D Discrete Wavelet Transform implements the convolution of the Coiflet basis with fingerprint sequence f(x, y) of size MxN. This is given by Equations (3.1) and (3.2) (Gonzalez and Woods, 2002):

$$w_{\varphi}(j_0, m, n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y)\, \varphi_{j_0,m,n}(x, y) \tag{3.1}$$

$$w^{i}{}_{\Psi}(j, m, n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y)\, \Psi^{i}{}_{j,m,n}(x, y), \quad i = \{H, V, D\} \tag{3.2}$$

Where,

j_0 = Arbitrary starting scale ($j_0 = 0$)

$w_\varphi(j_0, m, n)$ = Approximation coefficients for f(x, y) at scale j_0

$w^i{}_\psi(j, m, n)$ = Horizontal, vertical and diagonal details coefficients at scales $j \geq j_0$

$M = N = 2^j, for\ j = 0,1,2, \dots, j-1$

$m, n = 0,1,2, \dots, 2^j - 1$

Having obtained the approximation components, w_φ and detail components, $w^i{}_\psi$, the image sequence f(x, y) is reconstructed from 2-D Inverse DWT expressed by Equation 3.3(Gonzalez and Woods, 2002):

$$f(x,y) = \frac{1}{\sqrt{MN}} \sum_m \sum_n w_\varphi(j_0, m, n)\, \varphi_{j_0,m,n}(x,y) +$$

$$\frac{1}{\sqrt{MN}} \sum_{i=H,D,V} \sum_{j=j_0}^{\infty} \sum_m \sum_n w^i{}_\psi(j, m, n)\, \Psi^i{}_{j,m,n}(x,y) \qquad (3.3)$$

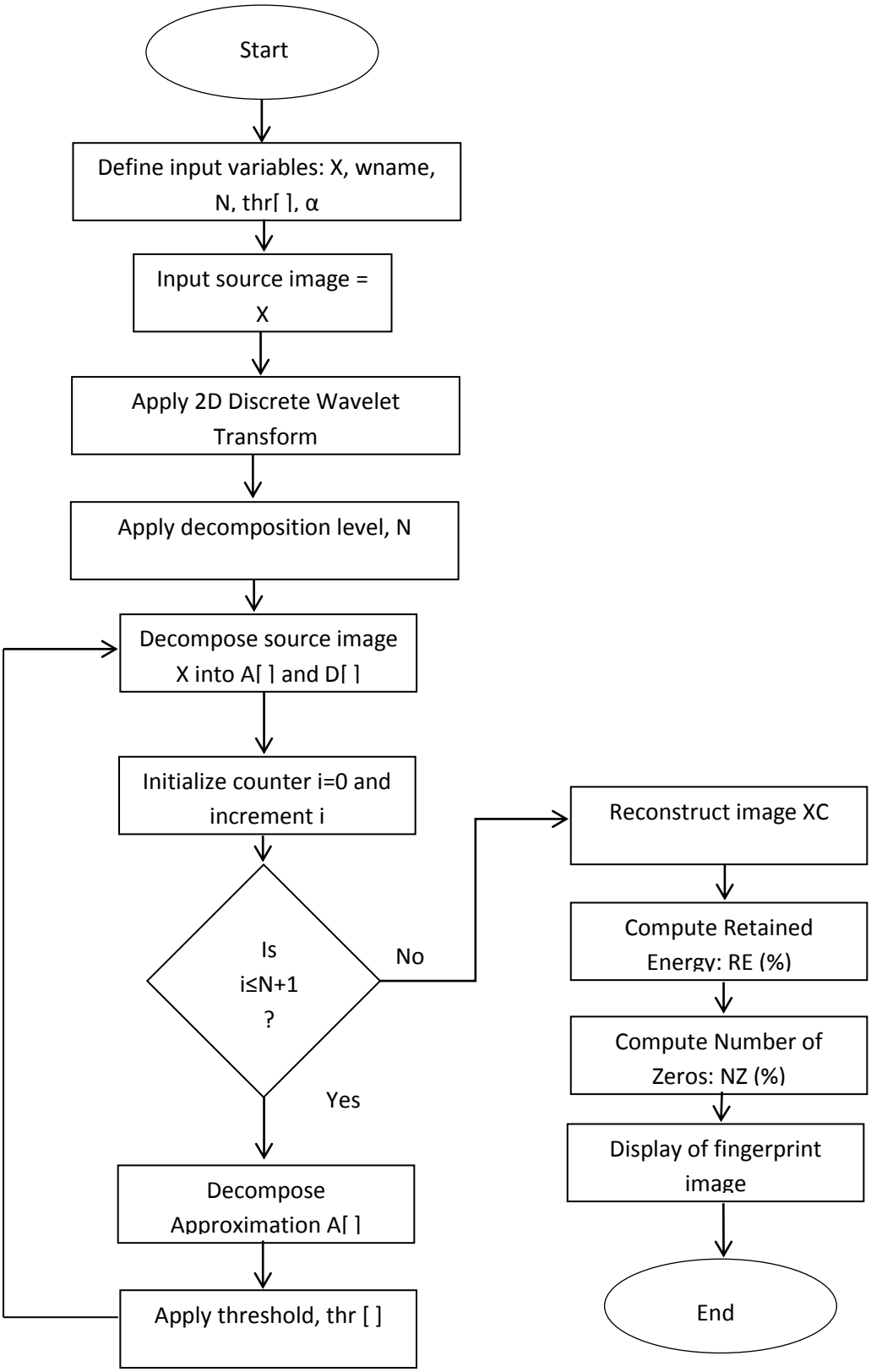


Figure 3.1: Flowchart for MATLAB Implementation of 2-D DWT Analysis of Fingerprint Image

The source fingerprint image (filename: cmp00001.pgm) was decomposed iteratively, with successive approximations components being decomposed in turn, so that one signal is broken down into many lower-resolution components.

Since the wavelet decomposition process is iterative, suitable number of levels were selected based on the nature of the fingerprint image signal. The decomposition level used in this comparative study include: levels 1, 2 and 3. As a result of this 3-level decomposition, approximation and detail components of the fingerprint image were generated at the three levels. The image decomposition was carried out independently using orthogonal Daubechies and Coiflet wavelet bases to evaluate the performance of the two bases in biometric fingerprint image approximation for compression algorithm. Using the Mallat decomposition algorithm, coefficients that are spatially related across scale (or frequency) can be sparsely represented and approximated. The decomposition started from level one and continued through iterative process until it reached level three.

3.3.1.3 *Energy De-Correlation of Transformed Image*

Level-dependent threshold strategy was applied to modify the resulting array of coefficients in the transformation to de-correlate the energy distribution of the fingerprint image.

The Birge-Massart strategy used in this work is based on results on adaptive functional estimation in regression or density contexts. It uses level-dependent thresholds obtained by wavelet coefficients selection rule. Given the decomposition level, j_0, and the sparcity parameter α, the length of coarsest approximation coefficient, m. The level-dependent threshold strategy is defined recursively in terms of the significant coefficient values, k_j as shown in Equation (3.4).

$$k_j = \frac{m}{(j_0+1-j)^{\alpha}} \tag{3.4}$$

Typically, the sparsity parameter α is set to 1.5 for compression applications (Misiti et al., 1996; Birge and Massart, 1997). At this stage, the significant coefficients were retained while the insignificant coefficients were truncated to zero based on the applied thresholds.

3.3.1.4 *Reconstruction of Fingerprint Image*

The transformed, thresholded and approximated fingerprint image was reconstructed from the resulting array of coefficients using the inverse 2D-discrete wavelet transform (IDWT). The analysis and synthesis wavelet filter pair achieved perfect reconstruction of the original image signal. The realization of perfect reconstruction is imperative in the process because wavelet-based algorithm is expected to achieve an efficient lossy compression in which the resulting image degradation is not visually perceptible but can only be measured quantitatively. The visual output of fingerprint image following the analysis, that is, display of original, reconstructed and compressed images from the MATLAB image processing toolbox are depicted for db2 and db4, as well as, coif2 and coif4 wavelet bases in Figures 3.2 to 3.5. From these outputs, there was no visually perceptible difference in the quality of the images. However, through the application of level-dependent thresholds, coefficients below the set threshold values have been truncated to zero and this loss of information were measured quantitatively.

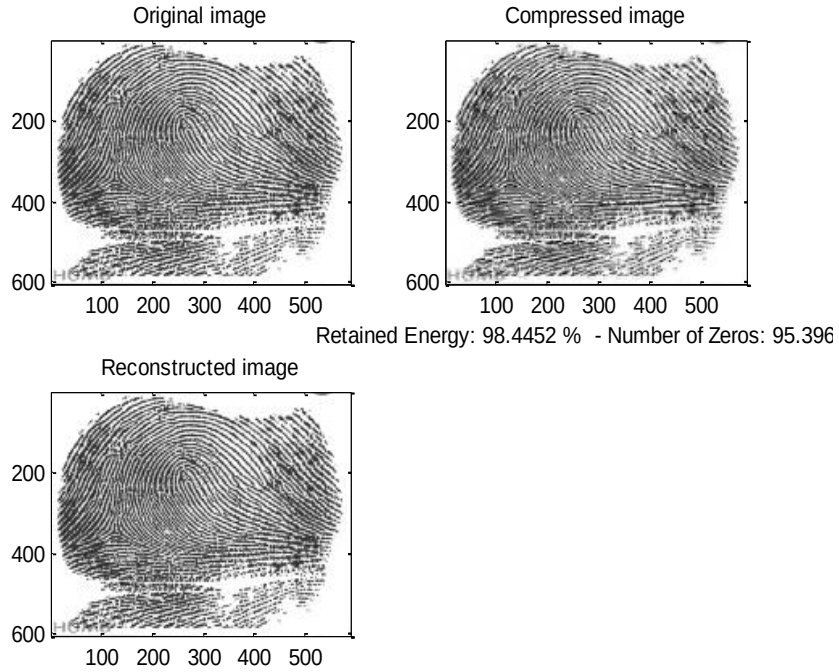


Figure 3.2: Estimation of RE (%) and NZ (%) in Wavelet Approximated Fingerprint Image using Daubechies Order 2 (db2) at Decomposition Level Three

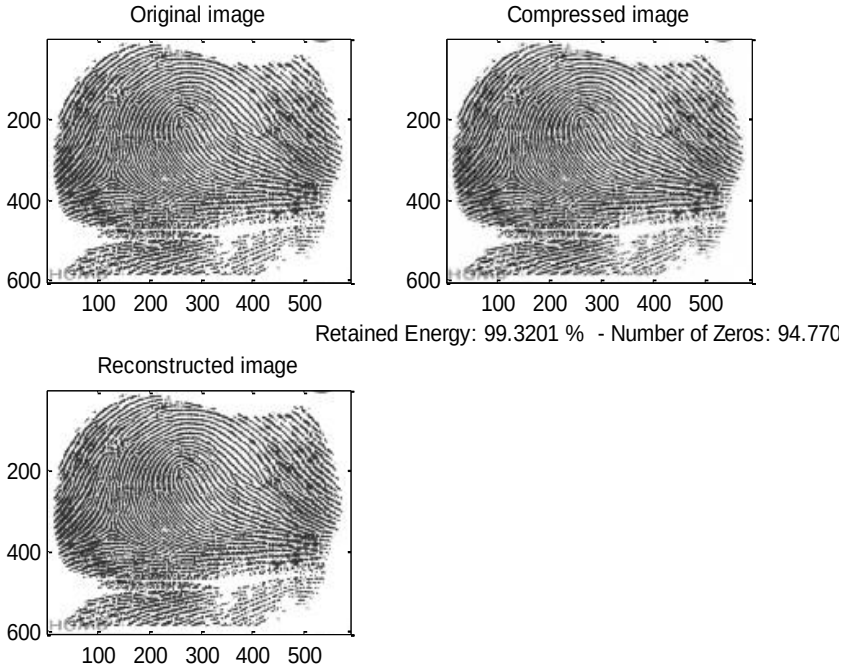


Figure 3.3: Estimation of RE (%) and NZ (%) in Wavelet Approximated Fingerprint Image using Coiflet Order 2 (coif2) at Decomposition Level Three

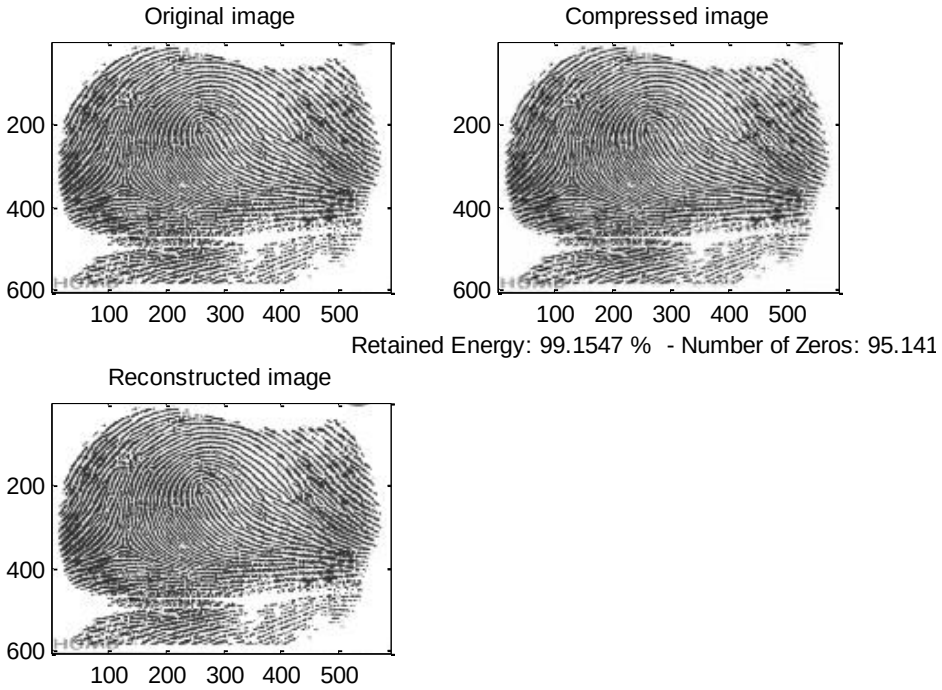


Figure 3.4: Estimation of RE (%) and NZ (%) in Wavelet Approximated Fingerprint Image using Daubechies Order 4 (db4) at Decomposition Level Three

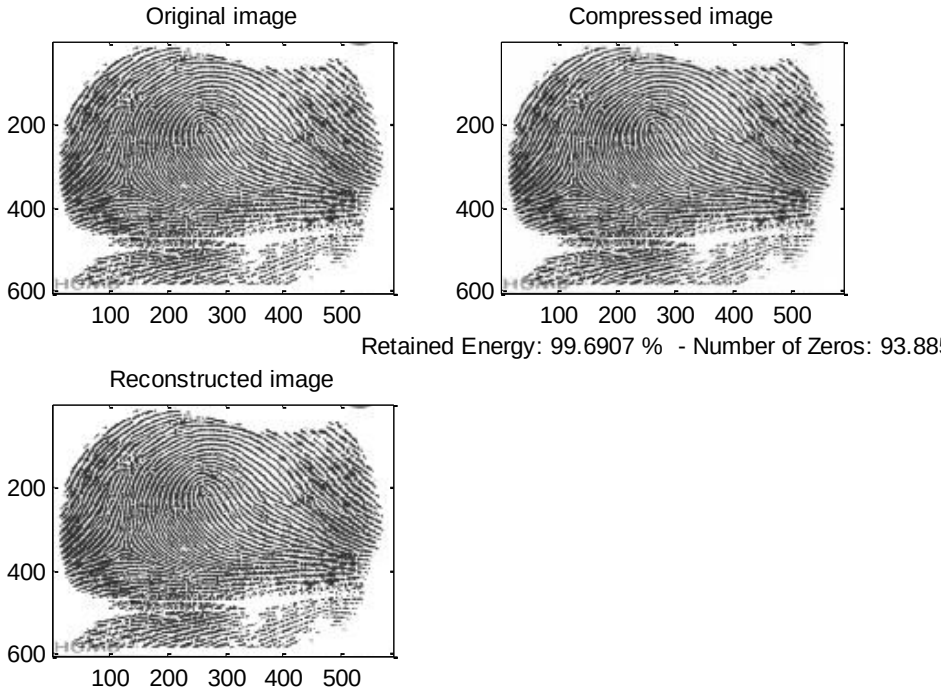


Figure 3.5: Estimation of RE (%) and NZ (%) in Wavelet Approximated Fingerprint Image using Coiflet Order 4 (coif4) at Decomposition Level Three

3.3.1.5 *Comparative Analysis of the Results for Daubechies and Coiflet Wavelets*

A comparative analysis of the results was carried out to determine the performance of each of the wavelets under investigation on the basis of percentage retained energy RE (%) and number of zeros NZ (%).

3.3.2 Non-Uniform Scalar Quantization using Lloyd-Max Procedure

The second stage of the proposed research methodology is non-uniform scalar quantization of transformed source fingerprint images. This stage is further broken down as follows:

i) Plot of histogram of source fingerprint pixel distribution;

ii) Non-uniform quantization of transformed source coefficients using Lloyd-Max algorithm;

iii) Computation of the MSQE and SQNR values to compare the performance of uniform and non-uniform quantization methods as well as dither-based non-uniform quantization and Lloyd-Max non-uniform quantization schemes.

3.3.2.1 *Histogram Plot of Source Image*

Each source fingerprint image is made up of an array (a matrix) of pixel values. The total number of pixel values in the array of transformed image coefficients were estimated and used to plot the histogram of the transformed image to determine the probability of occurrence of each image pixel or symbol. From the information generated from the histogram, the distribution of the input source symbols was determined based on the probability density function (PDF) of the image pixel distribution. Figure 3.6(b) shows the histogram plot of the 8-bit source fingerprint input and Figure 3.6(a) shows the histogram plot of 5-bit quantized fingerprint output at quantization level 32. The histogram plots were obtained using MATLAB image analysis tool.

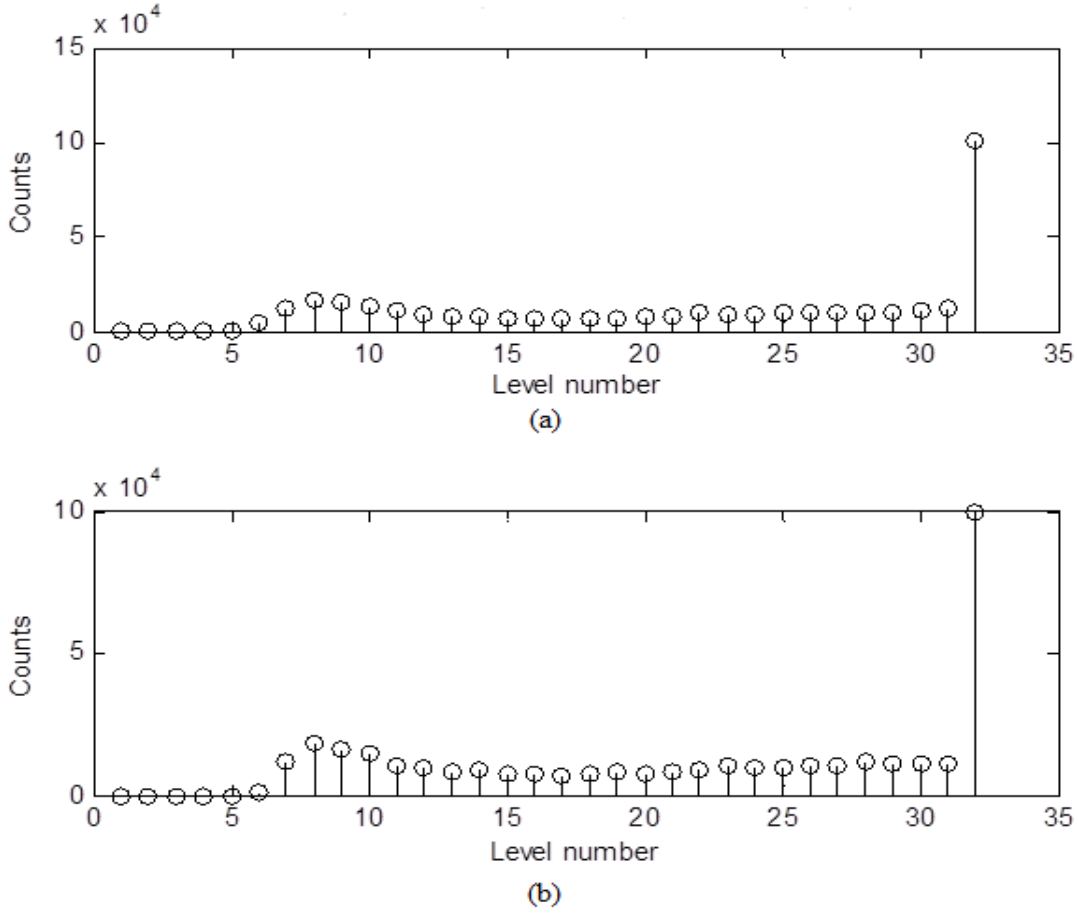


Figure 3.6: Histogram Plot of 5 bpp Quantized Fingerprint Output (a) and 8 bpp Source Fingerprint Input (b)

The histogram plots represent the pixel frequency distribution of the source fingerprint input and the quantized output. The two plots are nearly identical and the change is due to the difference between the 8-bits input image and the 5-bits quantized image. From the histogram plot the source fingerprint pixel distribution is not uniform and consequent upon this, uniform quantization method did not yield the best result. The MATLAB analysis plot of the quantization levels against the decision regions or partitions for the uniform and non-uniform quantization process are as shown in Figures 3.7 and 3.8 respectively. It will be noticed from Figure 3.7 which represents the uniform quantization process that the process used equal quantization step sizes all through the source distribution to quantize the fingerprint image. Whereas in Figure 3.8 the pixel values within the range of 0 to 50 are coarsely quantized with

large step-size differently from the values in the range of 50 to 250 intensity values which are finely quantized with smaller step size for the non-uniform quantization process. This is significant because the non-uniform quantization method used large quantization step-size on the range of pixels with insignificant values and lower step-size to quantize the range of pixels with significant values. The method achieved compression by discarding the insignificant pixels and retaining the significant ones.

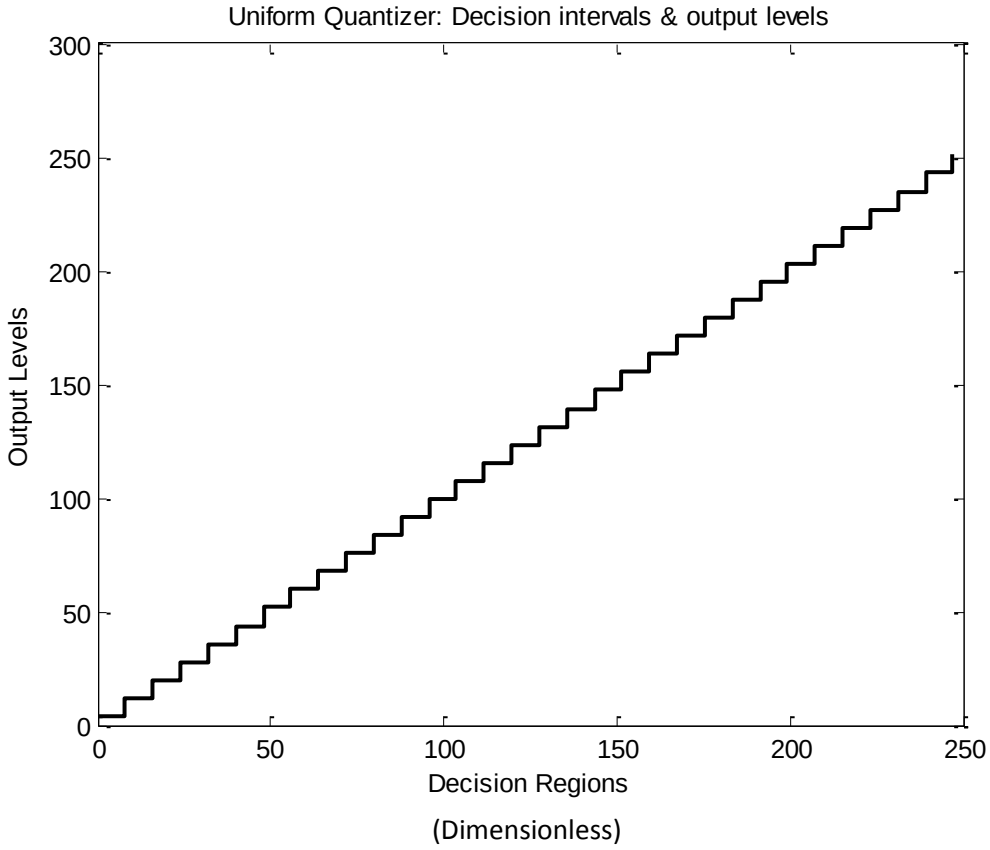


Figure 3.7: Plot of the Uniform Quantization Output Levels at 5 bpp

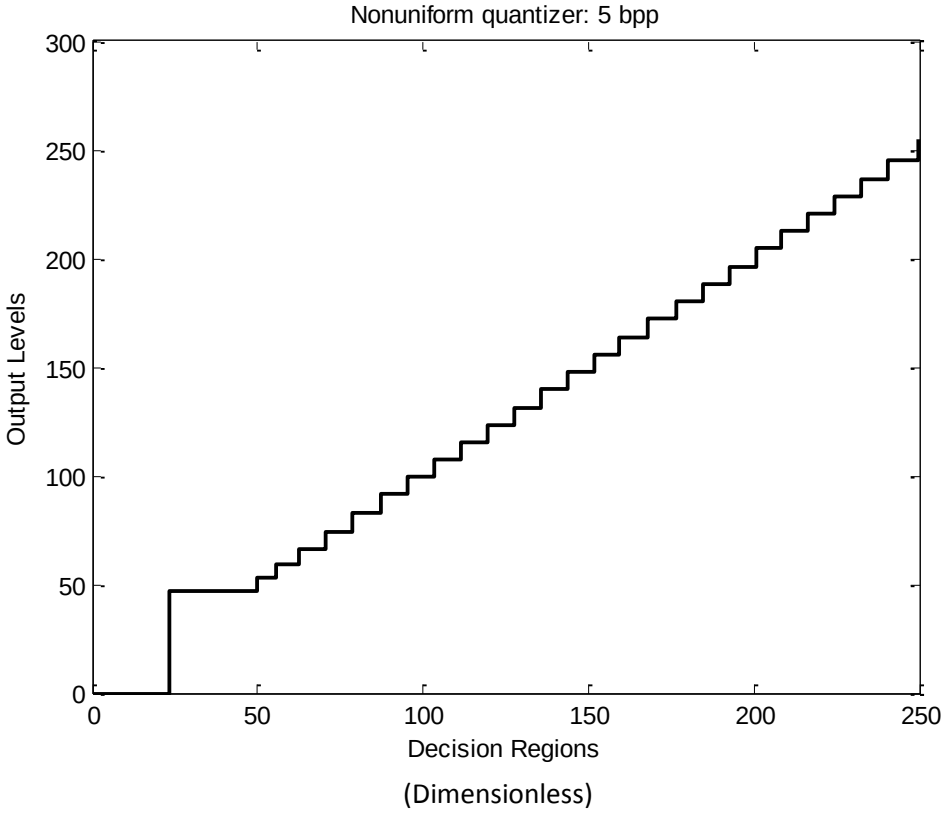


Figure 3.8: Plot of the Non-Uniform Quantization Output Levels at 5 bpp

3.3.2.2 *Non-Uniform Quantization of Transformed Source Fingerprint Image*

The source fingerprints used were 8-bit grayscale images and this means that 256 possible values of the image elements, that is 0 (black) to 255 (white). Lloyd-Max algorithm groups the pixel values of the source image into a number of partitions and measures the similarity between values in each partition. The algorithm starts by partitioning the source input set of values. It then calculates the average values, or centroid of each partition. It constructs a new partition by associating each value with the closest centroid. Then the centroids are recalculated for the new partitions and the algorithm repeats until convergence which is obtained when the centroids no longer change.

The Lloyd-Max quantization procedures involved the following steps:

i) Divide symbols (possible values) into M sets and the resulting partition is called initial set;

ii) The objective of Lloyd-Max algorithm is to minimize a distance metric within each set;

iii) Applied to the source fingerprint image, the algorithm minimizes the error between the corresponding value (reconstruction level) of an interval to its border (the thresholds);

iv) The thresholds are iteratively moved so that the partition changes iteratively, until there are no further changes or until convergence is reached.

The flowchart in Figure 3.9 was developed and represents the various steps in the implementation of the Lloyd-Max quantization algorithm for lossy compression. The design parameters for the lossy compression algorithm are as shown in Table 3.2.

Table 3.2: Design Parameters for Lossy Compression Based on Lloyd Max Algorithm

S/N	**Design Parameters**	**Values**
1.	Wavelet Transformation	Coiflets wavelets
2.	Bits Per pixel (bpp)	1 – 7 bits
3.	Quantization Levels, M	$M = 2^b$ (where b = (bpp))
4.	Quality Metrics	SQNR and MSQE

In Table 3.2, the quantization level, M was initialized based on the number of bits per pixel (bpp) used in the quantization process. The number of bpp used were 1 bpp to 7 bpp and this was because the original grayscale fingerprint images were represented using 8-bits. Therefore, in order to realize compression, the number of bpp lower than 8-bits should be used for quantization and this was done with the aim of achieving a tradeoff between image quality and the degree of degradation in the compression process. Since the quantization level $M = 2^b$, where b is the number of bits used per pixel for the quantization, therefore M is initialized to 2, 4, 8, 16, 32, 64 and 128 (for b = 1, 2, 3, 4, 5, 6, 7).

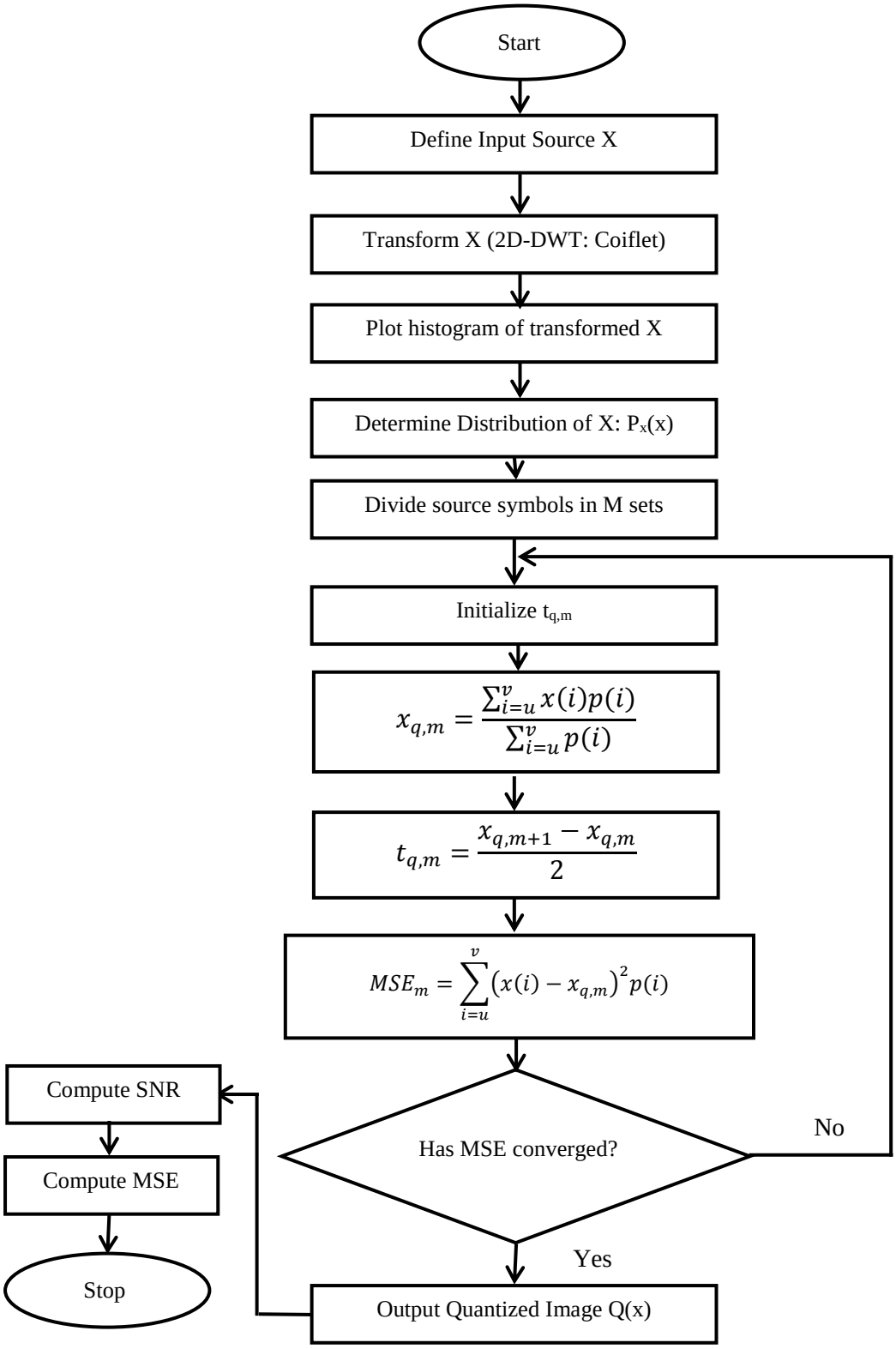


Figure 3.9: Flowchart of the Lloyd-Max Quantization Procedures

Based on the analysis flowchart in Figure 3.9, MATLAB scripts were written to implement the various stages of the fingerprint image quantization process as shown in Appendix II. The algorithm is detailed as follows:

1. Choose an initial set of M representative levels $x_{q,m}$. Pdf is then divided into M intervals;
2. Apply necessary centroid or threshold condition using Equation (3.5). This calculates (new) thresholds for each interval (centroid of $x_{q,m}$ and $x_{q,m+1}$);

$$t_{q,m} = \frac{x_{q,m+1} - x_{q,m}}{2} \tag{3.5}$$

3. Apply necessary minimum MSQE condition using Equation (3.6);

$$x_{q,m} = \frac{\sum_{i=u}^{v} x(i)p(i)}{\sum_{i=u}^{v} p(i)} \tag{3.6}$$

$x_{q,m}$ of each interval m=1, …, M is calculated to minimize MSQE.

The Mean Square Quantization Error of interval m, using pdf p(i) is given by:

$$MSE_m = \sum_{i=u}^{v} \left(x(i) - x_{q,m}\right)^2 p(i) \tag{3.7}$$

Where: i ranges over indices (u to v) in actual interval m;

x(u) = $t_{q,m}$ (lower threshold of interval m);

x(v) = $t_{q,m+1}$ (upper threshold of interval m);

x(i) = current value in interval m;

$x_{q,m}$= representative level of internal;

p(i) = P(x(i)): probability of symbol x(i).

4. The process is iterated. Step 3 and 4 are repeated until no further decrease in total MSQE;
5. Apply quantization on source image to output the quantized image.

After the iterative Lloyd-Max quantization procedure, the quality of the quantized output was estimated.

In addition, for dither-based non-uniform quantization process, random noise or dither signal was applied as a non-subtractive dither to modify the non-uniform quantization process in order to diffuse the quantization error. This is expected to further minimize the MSQE in the quantization process provided the biometric fingerprint image signal is suitable for this scheme. The details of the MATLAB implementation codes are shown in Appendix II.

3.3.2.3 *Computation of MSQE and SQNR*

The minimized Mean Square Quantization Error (MSQE) and Signal to Quantization Noise Ratio (SQNR) were determined for both uniform and non-uniform quantization processes from 1 bpp to 7 bpp. In addition, these performance measures were also determined for dither-based non-uniform quantization and Lloyd-Max quantization schemes to estimate the degree of distortion as against the image quality for the different quantization levels.

3.3.3 Arithmetic entropy coding to generate image bit-streams

The third stage of the proposed research methodology is arithmetic entropy coding of quantized fingerprint source symbols. This stage is further broken down as follows:

i) Comparative analysis of Shanon Fano, Huffman and Arithmetic Coding;

ii) Arithmetic entropy coding of quantized fingerprint source symbols;

3.3.3.1 *Comparative Analysis of Entropy Coding Schemes*

A comparative analysis of Shannon-Fano, Huffman and Arithmetic coding methods was carried out in terms of their coding efficiency with a view to choosing the most efficient for use in this work. The source entropy for fingerprint image (cmp00001.pgm) was determined (in Bits/symbol). In addition, the average code lengths (in Bits) were also determined for the three coding schemes. Then the source coding efficiencies for Shannon Fano, Huffman and Arithmetic coding schemes were determined from Equation (3.8) as shown in Table 3.3.

$$Coding\ Efficiency, E(S,L) = \frac{Entropy, H(S)}{Average\ code\ length\ l'(S,L)} * 100\% \quad (3.8)$$

Table 3.3: Coding Efficiency for Shannon-Fano, Huffman and Arithmetic Coding Schemes

Parameters	Shannon-Fano Coding	Huffman Coding	Arithmetic Coding
Source Entropy (Bits/Symbol)	6.48	6.48	6.48
Average Code Length (Bits)	13.96	8.42	7.14
Coding Efficiency (%)	46.40	76.95	90.68

The results in Table 3.3 show that the coding efficiencies for Shannon Fano, Huffman and arithmetic coding schemes were 46.40%, 76.95% and 90.68% respectively. This means that, for the source fingerprint image considered (that is, cmp00001.pgm), the arithmetic coding is the most efficient scheme. Therefore, the choice of Arithmetic coding is justified for source fingerprint image encoding in this work.

3.3.3.2 *Arithmetic Entropy Coding of Quantized Source*

Arithmetic coding assigns a sequence of bits to the source fingerprint image which is an array of symbols. The number of bits used to encode each symbol varies according to the probability of occurrence of the assigned symbol. The main idea of arithmetic coding is to assign each symbol to an interval. This singular interval is subdivided into several intervals with their sizes proportional to the probability of occurrence of the assigned symbols. The subinterval from the coded symbol is then taken as the interval for the next symbol. The length of the final subinterval is equal to the product of the probabilities of the individual symbol, which is the probability of the particular sequence of symbols in the source input. The output is the interval of the last symbol. The arithmetic encoding procedure entails the sequential computing of the lower and upper boundaries of the interval in which the current symbol lies. Supposed the symbol sequence of the fingerprint image to be coded is: $\{a_1, \ldots a_m\}$. The encoding procedure is as stated below (Thyagarajan, 2011):

i) The lower and upper boundaries of the encoding interval are initialized to 0 and 1 respectively, that is, $I_L(0) = 0$ and $I_U(0) = 1$;

ii) The initial interval (I_L, I_U) is further subdivided based on the probability of occurrence of the assigned symbols in accordance with cumulative distribution of the source input;

iii) The lower and upper boundaries of the final interval are computed as follows:

$$I_L(k) = I_L(k-1) + \big(I_U(k-1) - I_L(k-1)\big) * CDF(m_k - 1)$$

$$I_U(k) = I_L(k-1) + \big(I_U(k-1) - I_L(k-1)\big) * CDF(m_k)$$

$Where$:

$I_L = Lower\ boundary\ of\ the\ final\ interval$

$I_U = Upper\ boundary\ of\ the\ final\ interval$

$K = Source\ Alphabet\ size$

$m = Symbol\ Sequence\ length$

iv) The encoded tag, T_e is determined as the midpoint of the final interval as follows (Thyagarajan, 2011):

$$T_e = \frac{I_U(k) + I_L(k)}{2} \tag{3.9}$$

The arithmetic decoding procedure is as stated below (Thyagarajan, 2011):

i) The intial boundary values remain initialized to 0 and 1, that is, $I_L(0) = 0$ and $I_U(0) = 1$; Set K = 1;

ii) While K ≤ m, repeat the following steps:

iii) Compute the decoded tag value:

$T_d = (T_e - I_L(k-1))/(I_U(k-1) - I_L(k-1))$; where T_e is the encoded tag value corresponding to the encoded symbol sequence;

iv) Determine m_k such that $CDF(m_k - 1) \leq T < CDF(m_k)$;

v) Update the boundaries of the current interval using the equations as follows:

$$I_L(k) = I_L(k-1) + \big(I_U(k-1) - I_L(k-1)\big) * CDF(m_k - 1)$$

$$I_U(k) = I_L(k-1) + \big(I_U(k-1) - I_L(k-1)\big) * CDF(m_k)$$

vi) Then increment k.

3.3.4 Reconstruction of Source Fingerprint Image from Compressed Bit-Stream

The fourth stage of the proposed research methodology is the reconstruction of source fingerprint image from compressed bit-stream. In order to reconstruct the compressed fingerprint image from the encoded bit-streams, the processes of transformation, quantization and entropy encoding were reversed with arithmetic entropy decoding and inverse discrete wavelet transform (IDWT). The forward and reverse processes are depicted in the program flowchart as shown in Figure 3.10 and the algorithm was implemented by writing a MATLAB script using the image processing toolbox as shown in Appendices II.

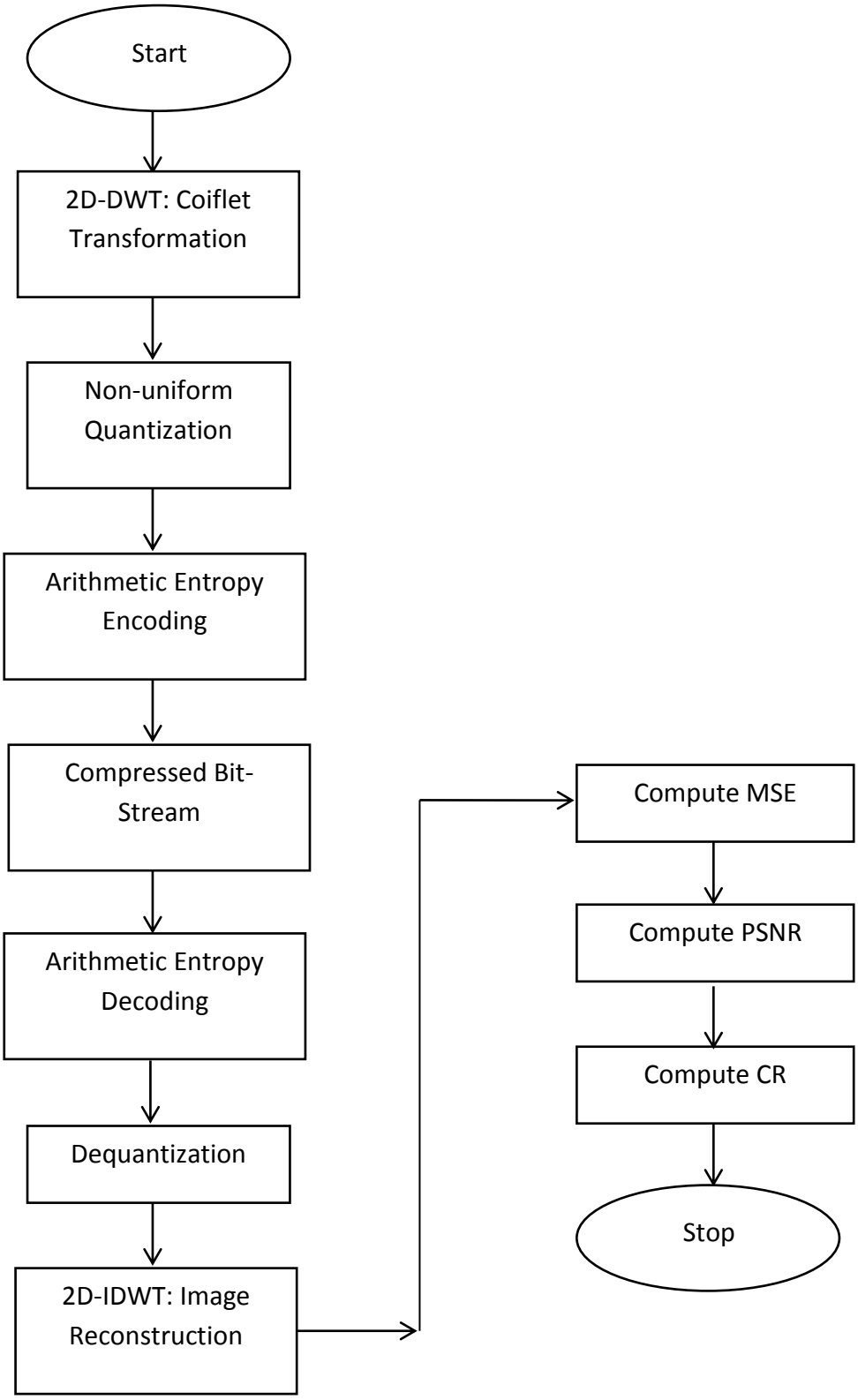


Figure 3.10: Program Flowchart of the Proposed Lossy Compression Algorithm for Biometric Fingerprint Image

3.3.5 Validation and Evaluation of the Performance of Compression Algorithm

The fifth stage of the research methodology is the validation and evaluation of the performance of the proposed compression. In order to evaluate the performance of the developed biometric fingerprint compression algorithm, Peak Signal to Noise Ratio (PSNR) and Compression Ratio (CR) were used as performance metrics. PSNR was used as a measure of the resultant image quality after compression and the Compression Ratio was used as a measure of the degree of compression achievable. In the performance analysis, a trade-off was achieved between the achievable compression ratio and the allowable degradation which is a function of achievable PSNR in the compression process. Compression ratio values were computed with the intent of maintaining compressed fingerprint image quality consistent with the requirement of preserving essential fingerprint features such as the core, ridge endings and bifurcations for biometric application purpose.

The compression process involved the manipulation of the source fingerprint's symbols (pixels) by retaining the significant symbols and discarding the insignificant ones. The significant symbols contribute more to the energy of the image whereas the insignificant symbols contribute little or nothing to the energy of the image. The source fingerprint images were transformed using the Coiflet wavelets at a decomposition level of three as shown in Figures 3.11.

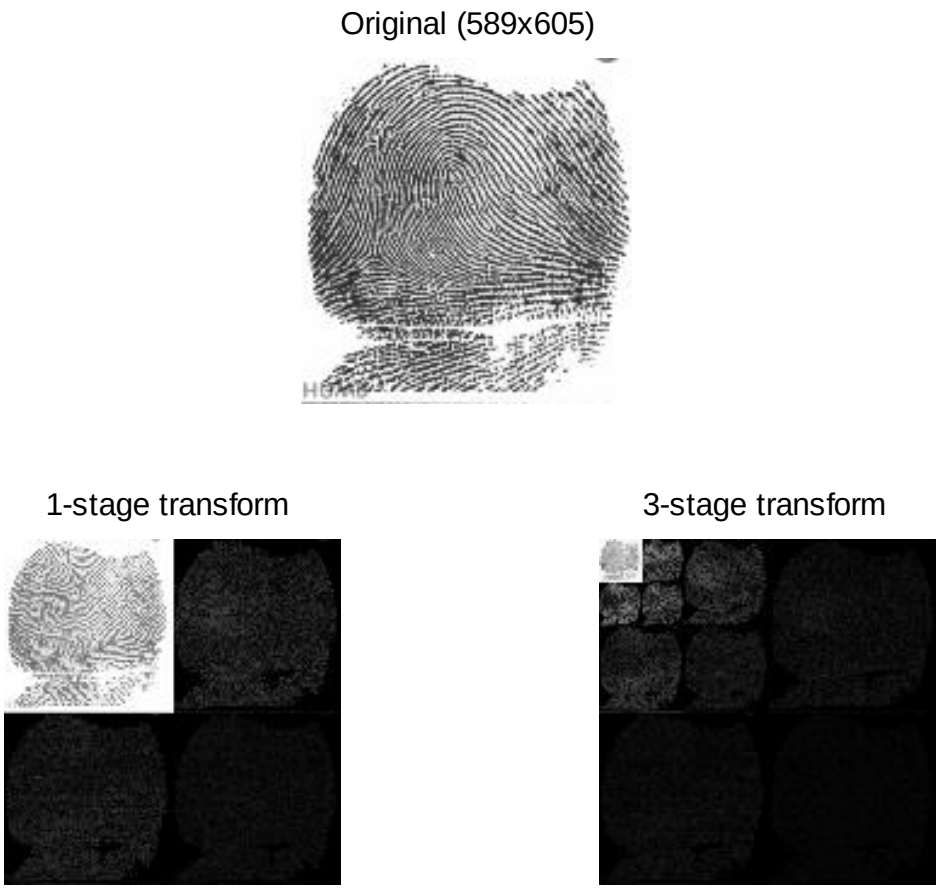


Figure 3.11: 3-Level Decomposition of Fingerprint Image in the Compression Process

In addition, the source images were non-uniformly quantized and entropy coded. Compression ratios of 15:1, 20:1, 40:1 and 80:1 were applied as follows:

i) For a compression ratio of 15:1, a fraction of $\frac{1}{16}$ of the total number of symbols in the source image were retained while a fraction of $\frac{15}{16}$ of the total number were discarded and the output stored as compressed image;

ii) For a compression ratio of 20:1, a fraction of $\frac{1}{21}$ of the total number of symbols in the source image were retained while a fraction of $\frac{20}{21}$ of the total number were discarded and the output stored as compressed image;

iii) For a compression ratio of 40:1, a fraction of $\frac{1}{41}$ of the total number of symbols in the source image were retained while a fraction of $\frac{40}{41}$ of the total number were discarded and the output stored as compressed image;

iv) For a compression ratio of 80:1, a fraction of $\frac{1}{81}$ of the total number of symbols in the source image were retained while a fraction of $\frac{80}{81}$ of the total number were discarded and the output stored as compressed image.

Additionally, the corresponding values of the PSNR as a measure of image quality after compression were estimated as shown in Figures 3.12 to 3.14.

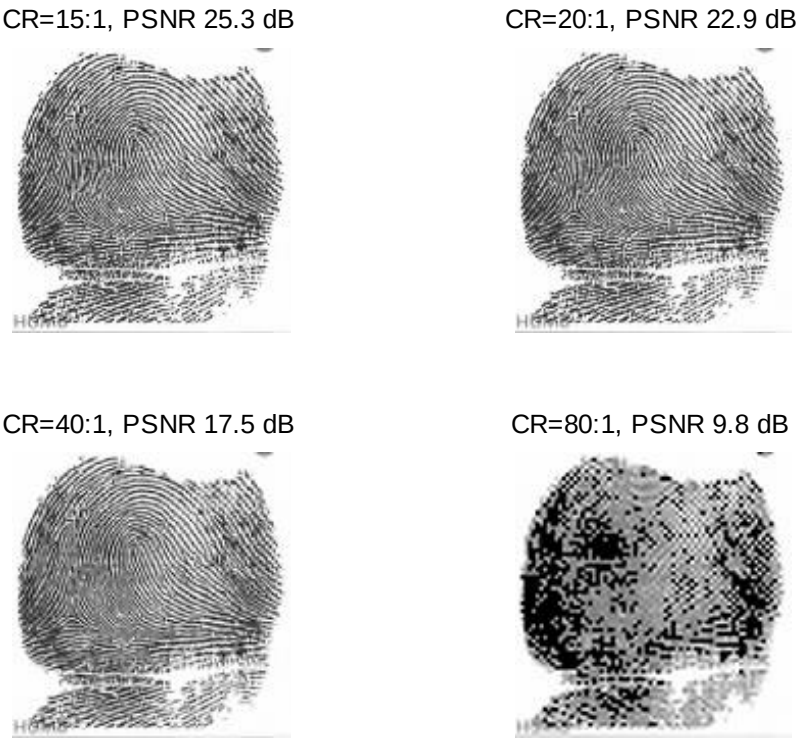
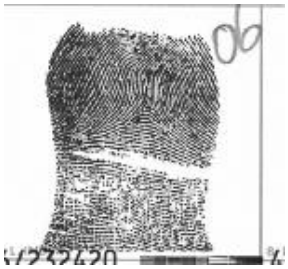


Figure 3.12: Compression of Fingerprint Image with Values of Compression Ratio and their Corresponding PSNR Values

CR=15:1, PSNR 32.1 dB



CR=20:1, PSNR 29.1 dB



CR=40:1, PSNR 22.4 dB



CR=80:1, PSNR 11.5 dB

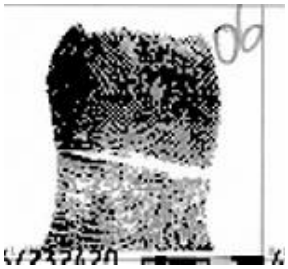
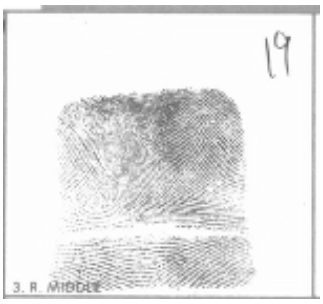


Figure 3.13: Compression of Single Fingerprint Image with Values of Compression Ratio and their Corresponding PSNR Values

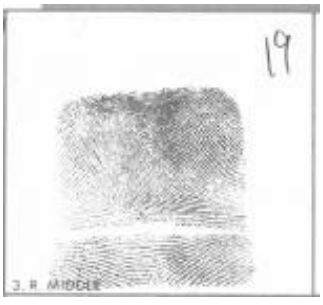
CR=15:1, PSNR 36.1 dB



CR=20:1, PSNR 32.9 dB



CR=40:1, PSNR 26.0 dB



CR=80:1, PSNR 5.9 dB



Figure 3.14: Compression of Fingerprint Image with Values of Compression Ratio and their Corresponding PSNR Values

Figures 3.12 to 3.14 show the compressed fingerprint images for compression ratios (CRs) of 15:1, 20:1, 40:1 and 80:1 as well as the respective PSNR values. The compressed image output produced the highest PSNR value at CR 15:1 which indicated that the highest image quality is achieved at this level of compression. Moreover, as the level of compression was raised to CR of 20:1 through 80:1, the quality of the compressed image output, which is a function of the PSNR obtained for the process, began to decrease. This was evident in the distortion observed in the biometric features of the fingerprint images and these features were completely lost at CR of 80:1. This trend was repeated in the compression process for all the experimental fingerprint image data.

CHAPTER FOUR

RESULTS AND DISCUSSIONS

4.1 Introduction

The results of the various experimental analyses carried out in this work are presented in this chapter. These include the results of the comparative analysis of wavelet bases implemented with level-dependent and global threshold strategies, uniform and non-uniform quantization techniques, as well as the performance measurement of the proposed lossy compression scheme.

4.2 Results of Performance Analysis of Wavelet Bases

Results of the performance analysis of biometric fingerprint image transformation using Coiflet and Daubechies wavelets bases for image approximation are shown in Tables 4.1 and 4.2 respectively. The analysis involved fingerprint image decomposition and thresholding. The analysis was conducted on the basis of the measures of percentage retained energy, RE (%) and the percentage number of zero NZ (%) resulting from the wavelet transformation process.

Table 4.1: Results of RE (%) and NZ (%) for Coiflet Wavelets

Decomp. Level	Wavelet Bases					
	coif2		coif3		coif4	
	NZ (%)	RE (%)	NZ (%)	RE (%)	NZ (%)	RE (%)
1	51.13	99.99	51.13	99.99	51.13	99.99
2	83.77	99.91	83.25	99.94	82.89	99.95
3	94.77	99.32	94.34	99.57	93.88	99.69

Table 4.2: Results of RE (%) and NZ (%) for Daubechies Wavelets

Decomp. Level	Wavelet Bases					
	db2		db3		db4	
	NZ (%)	RE (%)	NZ (%)	RE (%)	NZ (%)	RE (%)
1	51.13	99.98	51.13	99.99	51.13	99.99
2	84.38	99.74	84.15	99.86	84.07	99.89
3	95.34	98.45	95.21	98.91	95.14	99.15

Results of the comparative analysis of global threshold and level-dependent threshold strategies in terms of percentage retained energy, RE (%) as applied to the transformation of fingerprint image using Coiflet wavelet are shown in Table 4.3.

Table 4.3: Results of the Comparative Analysis of Global Threshold and Level-Dependent Threshold Strategies

Decomp. Level	Wavelet Bases					
	coif2		coif3		coif4	
	Global Threshold	Level-Dependent	Global Threshold	Level-Dependent	Global Threshold	Level-Dependent
	RE (%)	RE (%)	RE (%)	RE (%)	RE (%)	RE (%)
1	98.15	99.99	99.89	99.99	99.91	99.99
2	98.15	99.91	98.55	99.94	98.73	99.95
3	97.32	99.32	97.42	99.57	97.42	99.69

The global threshold strategy was used in the work carried out by Islam et al (2012). The global threshold method applied fixed threshold values to all coefficients in the fingerprint image array at all decomposition level. In the case of the level dependent threshold, variable threshold values were applied at different decomposition levels.

4.3 Results of Comparative Analysis of Quantization Schemes

The results of comparative analysis of uniform quantization and Lloyd Max non-uniform quantization methods for source fingerprint images in terms of the MSQE and SQNR performance metrics are as shown in Table 4.4.

Table 4.4: Comparison of the Non-Uniform Quantization and Uniform Quantization Methods on the Basis of Quality and Distortion Measures

No. of Bits (Bits/Pixels)	**Non-uniform Quantization**		**Uniform Quantization**	
	MSQE	**SQNR (dB)**	**MSQE**	**SQNR (dB)**
1	8527.2	-1.9204	1873.0	6.2924
2	667.9327	9.1403	551.6564	10.8670
3	64.416	19.2977	130.2040	17.0903
4	15.4413	25.5008	32.5056	22.9594
5	3.915	31.4603	8.2034	28.9540
6	0.9247	37.7277	2.1361	34.8221
7	0.1896	44.6083	0.6304	40.1349

These results represent the MSQE and SQNR values which are the measures of the image distortion and quality respectively in the quantization process. Table 4.5 shows the results of the MSQE and SQNR values obtained from the comparative analysis of dither-based non-uniform quantization scheme and Lloyd-Max non uniform quantization scheme for the number of bits per pixel (bpp) of 1 to 7.

Table 4.5: Comparison of the Dither-Based Non-Uniform Quantization and Lloyd-Max Non-Uniform Quantization Methods on the Basis of Quality and Distortion Measures

No. of Bits (Bits/Pixels)	**Dither-Based Non-uniform Quantization**		**Lloyd-Max Non-uniform Quantization**	
	MSQE	**SQNR (dB)**	**MSQE**	**SQNR (dB)**
1	8416.7	-1.8637	8527.2	-1.9204
2	672.9086	9.1082	667.9327	9.1403
3	66.5738	19.1546	64.416	19.2977
4	17.0952	25.0590	15.4413	25.5008
5	5.4085	30.0569	3.915	31.4603
6	2.4012	33.5835	0.9247	37.7277
7	1.7006	35.0818	0.1896	44.6083

4.4 Results of Performance Evaluation of the Proposed Compression Algorithm

The performance evaluation results of the proposed source coding for fingerprint compression on the basis of compression ratio (CR) and peak signal to noise ratio (PSNR) are as shown in Tables 4.6 and 4.7.

Table 4.6 tabulates the different CR values used in the experimental process with their corresponding PSNR values for two experimental source data (cmp00001.pgm and cmp00002.pgm). The CR values were obtained by the level-dependent threshold process of retaining significant image pixel and truncating to zero the insignificant image pixels at different compression ratios of 15:1, 20:1, 40:1 and 80:1 and then measure the compressed image quality output using the PSNR quality metric.

Table 4.6: Performance Analysis of the Proposed Compression Algorithm on Fingerprint Images at Different Compression Ratios

Source Images	**Compression Ratio (CR)**	**PSNR (dB)**
Cmp00001.pgm	15:1	25.30
Cmp00001.pgm	20:1	22.90
Cmp00001.pgm	40:1	17.50
Cmp00001.pgm	80:1	9.80
Cmp00002.pgm	15:1	32.10
Cmp00002.pgm	20:1	29.10
Cmp00002.pgm	40:1	22.40
Cmp00002.pgm	80:1	11.50

Table 4.7 tabulates the compression ratio of 20:1 for all the NIST experimental fingerprint images used. In addition, the corresponding values of the PSNR which is the measure of the resultant image quality in the compression process were also tabulated.

Table 4.7: PSNR Values Obtained for a Compression Ratio of 20:1

Source Images	Compression Ratio (CR)	PSNR (dB)
Cmp00001.pgm	20:1	22.90
Cmp00002.pgm	20:1	29.10
Cmp00003.pgm	20:1	32.90
Cmp00004.pgm	20:1	33.30
Cmp00005.pgm	20:1	35.50
Cmp00006.pgm	20:1	28.00
Cmp00007.pgm	20:1	25.60
Cmp00008.pgm	20:1	28.80
Cmp00009.pgm	20:1	25.10
Cmp00010.pgm	20:1	25.40
Cmp00011.pgm	20:1	25.40
Cmp00012.pgm	20:1	24.90
Cmp00013.pgm	20:1	24.90
Cmp00014.pgm	20:1	25.70
Cmp00015.pgm	20:1	30.40
Cmp00016.pgm	20:1	31.70
Cmp00017.pgm	20:1	26.00
a001.pgm	20:1	27.10
a002.pgm	20:1	25.50
a018.pgm	20:1	30.10
a039.pgm	20:1	24.90
a070.pgm	20:1	26.50
a076.pgm	20:1	27.60
a089.pgm	20:1	24.10
a107.pgm	20:1	27.10
a129.pgm	20:1	27.10
a165.pgm	20:1	26.30
b124.pgm	20:1	25.50
b157.pgm	20:1	26.30
b186.pgm	20:1	27.10

4.5 Discussion of Results

The results of the comparative analysis of wavelet bases, quantization schemes and performance measurement of the proposed lossy compression algorithm were discussed in this chapter.

4.6 Performance Analysis of Wavelet Bases

The comparative performance analysis of Coiflet and Daubechies wavelet bases was carried out in terms of the percentage retained energy RE (%) and percentage number of zeros, NZ (%) to determine their efficiency in image transformation. At wavelet decomposition level one, the two wavelet bases have the same values for RE (%) and NZ (%) for the three wavelet orders investigated (orders 2, 3 and 4). However, decomposition level three is the level of interest as it is the level at which 8 bits source fingerprint image is completely decomposed. At level three, the RE (%) values for Coiflets ranged from 99.32% to 99.69% whereas the RE (%) values for Daubechies ranged from 98.45% to 99.15%. This result is significant in the sense that the Coiflet wavelets exhibited higher energy retention capacity than Daubechies wavelet. This is due to the fact that Coiflet wavelet has the property of symmetry and Daubechies wavelet does not. This means that the capacity of Coiflet wavelet for higher image energy retention will enable the realization of compression algorithm with a lower image distortion than Daubechies. Consequently, compressed image with higher perceptual quality is possible with Coiflet wavelet image approximation than that of Daubechies wavelet.

Additionally, it is evident in the results that as the number of vanishing moment or wavelet order of the investigated wavelets and the decomposition level was increased, the improvement in RE (%) obtained for Coiflet wavelet became more evident than that of Daubechies wavelet. The results also revealed higher values of NZ (%) for Daubechies wavelet than Coiflet wavelet with the value range of 95.14% to 95.34% for Daubechies and 93.88 to 94.77% for Coiflets. These results showed that more insignificant coefficients can be truncated to zero with Daubechies wavelet than Coiflet and this possibility makes the lossy character of the proposed compression algorithm using Daubechies more pronounced. Hence, the distortion rate is higher in Daubechies wavelet. The bar chart plots of the RE (%) and NZ

(%) results for decomposition level three for Daubechies and Coiflet wavelet bases are shown as in Figures 4.1 and 4.2.

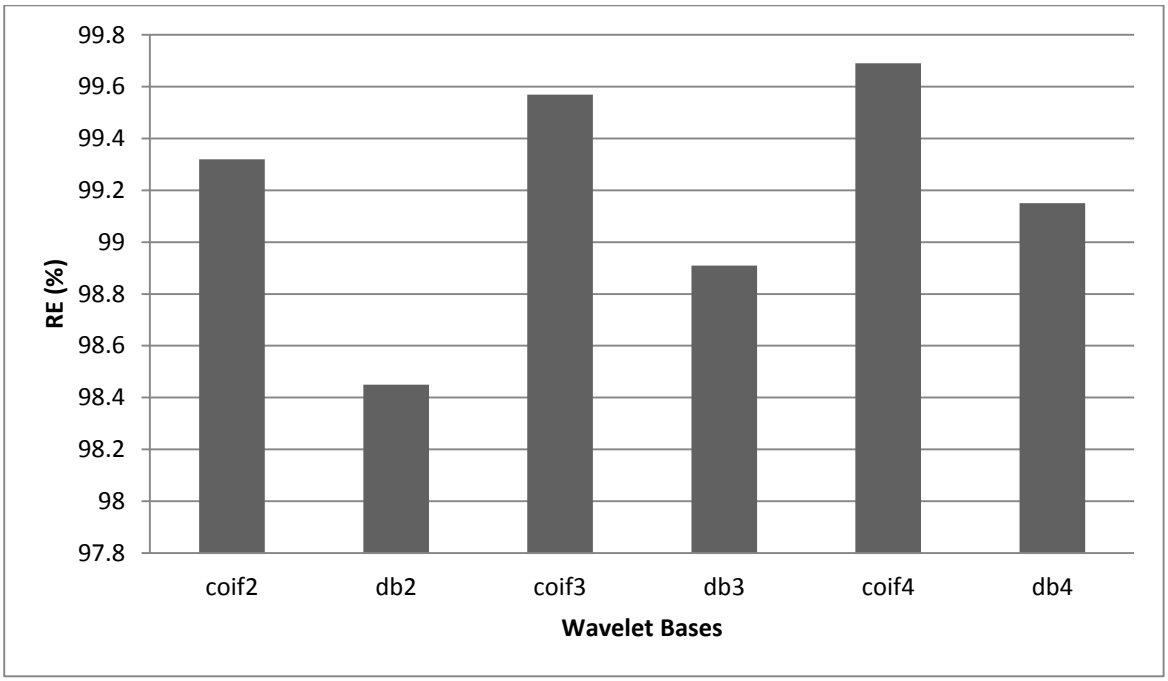


Figure 4.1: Estimation of RE (%) in Wavelet Approximated Fingerprint Image using Daubechies and Coiflet Wavelet Bases at Decomposition Level Three

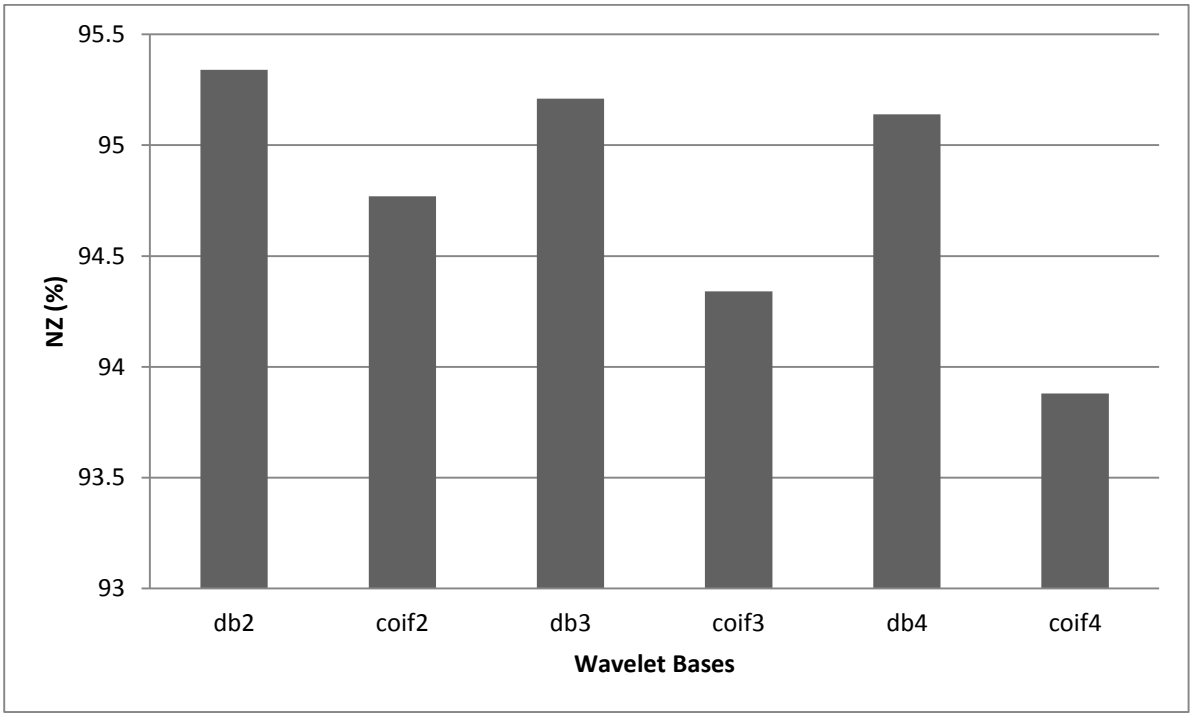


Figure 4.2: Estimation of NZ (%) in Wavelet Approximated Fingerprint Image using Daubechies and Coiflet Wavelet Bases at Decomposition Level Three

4.7 Comparative Analysis of Global Threshold and Level-Dependent Threshold Strategies

The result of the comparative analysis of global threshold and level-dependent threshold strategies in terms of percentage retained energy as applied to the transformation of fingerprint image using Coiflet wavelet was carried out to determine the most efficient threshold technique. The global threshold strategy was used in the work carried out by Islam et al (2012). The global threshold method applied fixed threshold values to all coefficients in the fingerprint image array at all decomposition level. In the case of the level dependent threshold, variable threshold values were applied at different decomposition levels. From the analysis results as evident in Figure 4.3, the level dependent threshold performed better than the global threshold method as it produced higher percentage retained energy at the three decomposition level used in the experiment. The significance of this result is that the quality of transformed fingerprint image obtained from level dependent threshold method is higher than that of global threshold method. Therefore, level dependent threshold strategy is used in this work for improved performance.

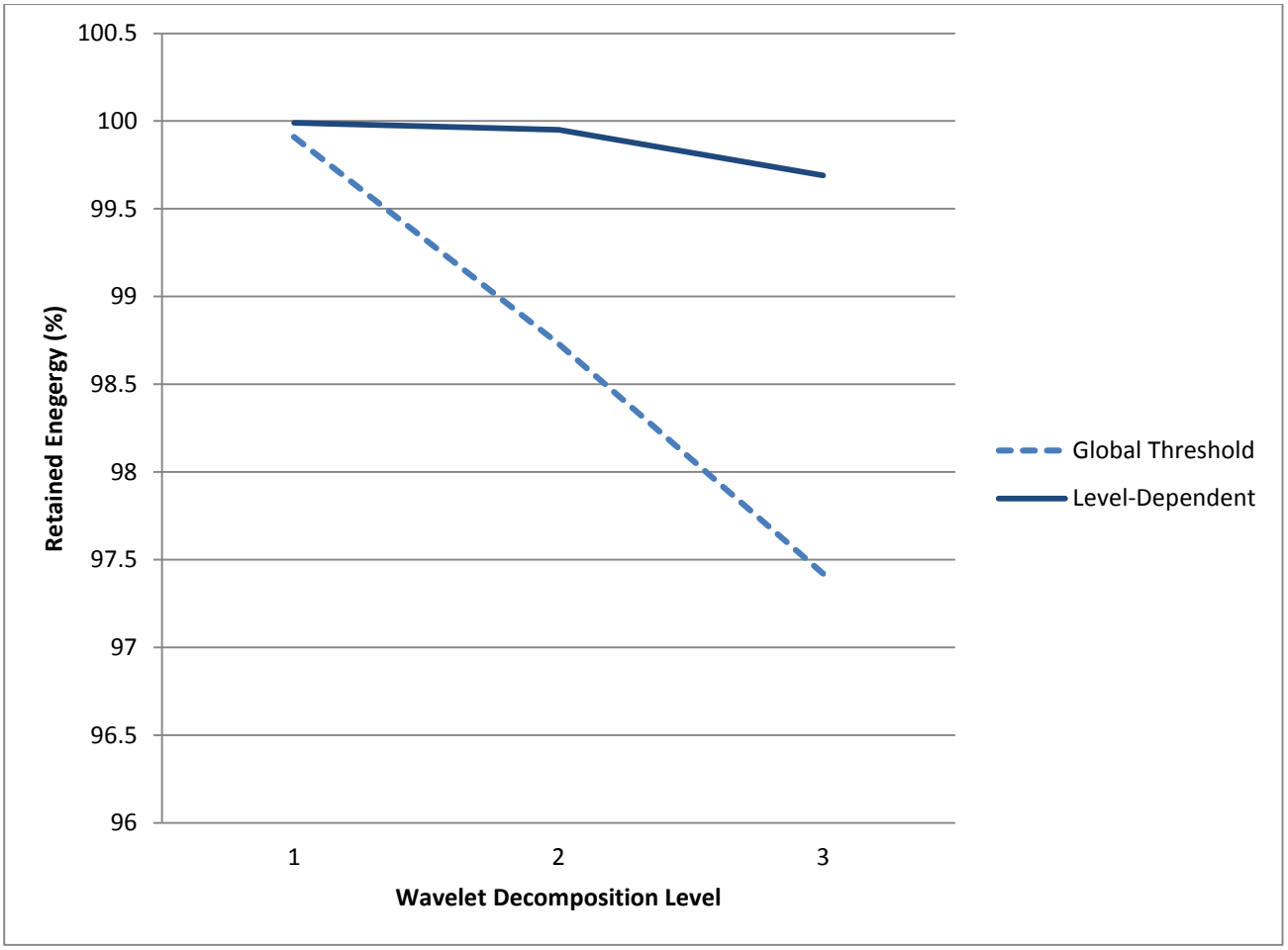


Figure 4.3 Plot of the Efficiency of Global Threshold and Level Dependent Threshold Strategies in Terms of Percentage Retained Energy

4.8 Comparison of the Non-Uniform Quantization and Uniform Quantization Methods

The comparative analysis of non-uniform quantization and uniform quantization schemes was carried out to determine the most efficient quantization scheme for the proposed lossy compression algorithm. The analysis was done in terms of the mean square quantization error (MSQE) and signal to quantization noise ratio (SQNR) which are the measures of the image distortion and quality respectively. It was observed on the one hand that the SQNR values for non-uniform quantization increased from 19.2977 dB for 3 bpp to 44.6083 dB for 7 bpp whereas for the same range (3 bpp to 7 bpp) for uniform quantization, SQNR values increased from 17.0903 dB to 40.1349 dB. On the other hand, the MSQE values for non-

uniform quantization decreased from 64.414 for 3 bpp to 0.1896 for 7 bpp while for the same range (3 bpp to 7 bpp) for uniform quantization, MSQE values decreased from 130.2040 for 3 bpp to 0.6304 for 7 bpp. These results indicated that the higher the number of bits used in the quantization process the higher the quality and the less the distortion in the resulting quantized images. Perhaps, more significantly, from 3 bpp and 7 bpp quantization, the results revealed that the non-uniform quantization consistently performed better than the uniform quantization method and this is depicted in the plots for SQNR and MSQE as shown in Figures 4.4 and 4.5. The MSQE values plotted against the number of bits per pixel (bpp) in the uniform and non-uniform quantization processes is shown in Figure 4.4 and it was revealed that MSQE values for non-uniform method was lower than that of the uniform method but as the number of quantization bits approaches 8-bits, the MSQE values for both methods are almost equal.

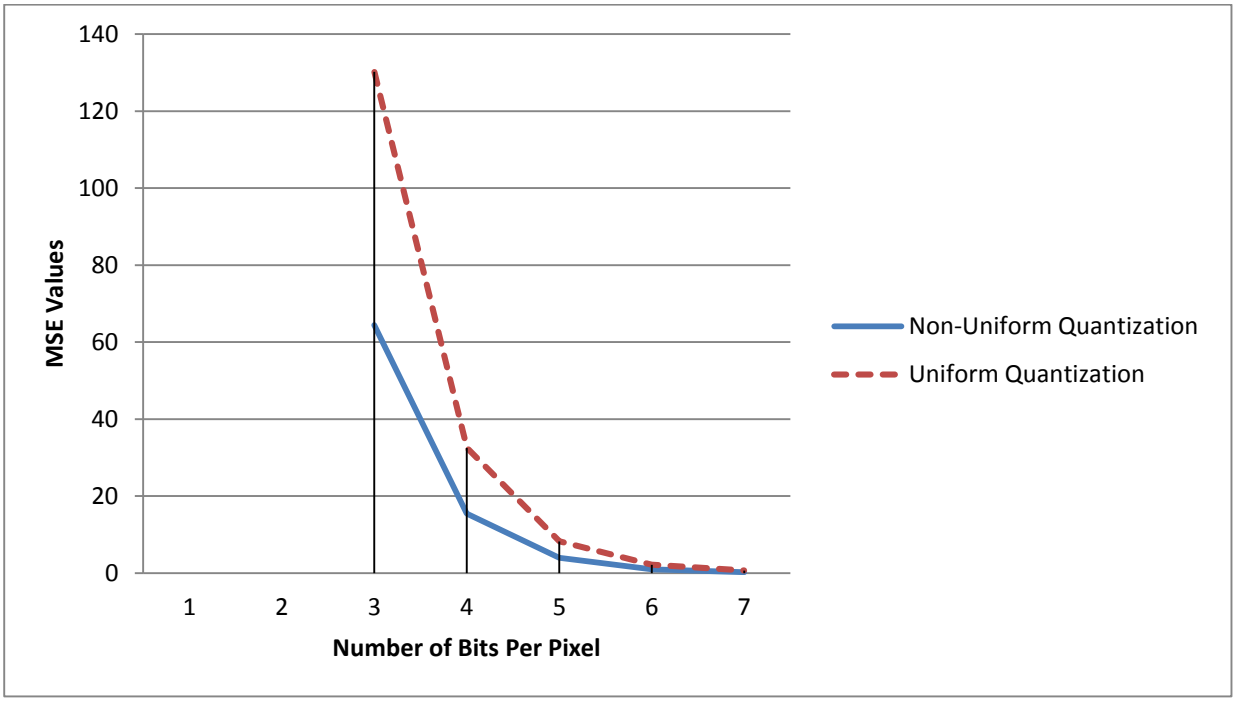


Figure 4.4: Plot of MSQE Values for Non-Uniform and Uniform Quantization

Figure 4.5 shows the SQNR values plotted against the number of bits per pixel (bpp) in the uniform and non-uniform quantization processes and it was revealed that SQNR values for non-uniform method were higher than that of the uniform method.

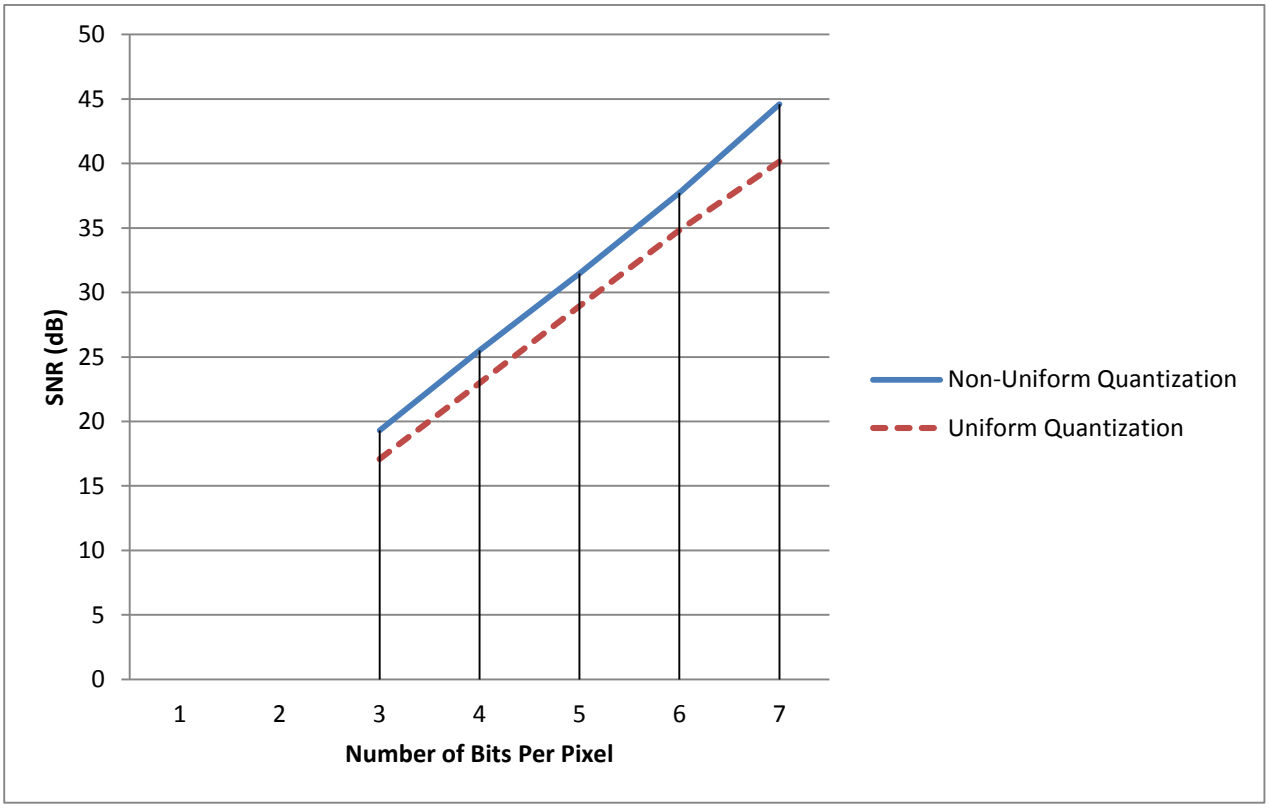


Figure 4.5: Plot of SQNR in Decibels (dB) for Non-Uniform and Uniform Quantization

A comparative analysis between two non-uniform quantization schemes, namely; Lloyd Max non-uniform quantization and dither-based non-uniform quantization schemes was also carried out in terms of the MSQE and PSQNR. The MSQE values obtained for dither-based non-uniform quantization scheme were higher than the ones obtained for Lloyd-Max scheme for the number of bits per pixel (bpp) from 2 to 7. However, for 1 bpp, the MSQE value obtained for dither-based quantization process was lower than that of Lloyd-Max method. Perhaps more significant is the fact that the lower the bit number, the more the efficiency of the dither-based quantization process. In Figure 4.6, it was observed that as the number of bits used in the quantization process decreases, the MSQE values obtained for dither-based quantization scheme got very close to the MSQE values for the Lloyd-Max quantization

scheme. This means that the performance of both schemes became very close as the number of bits for quantization process was decreased.

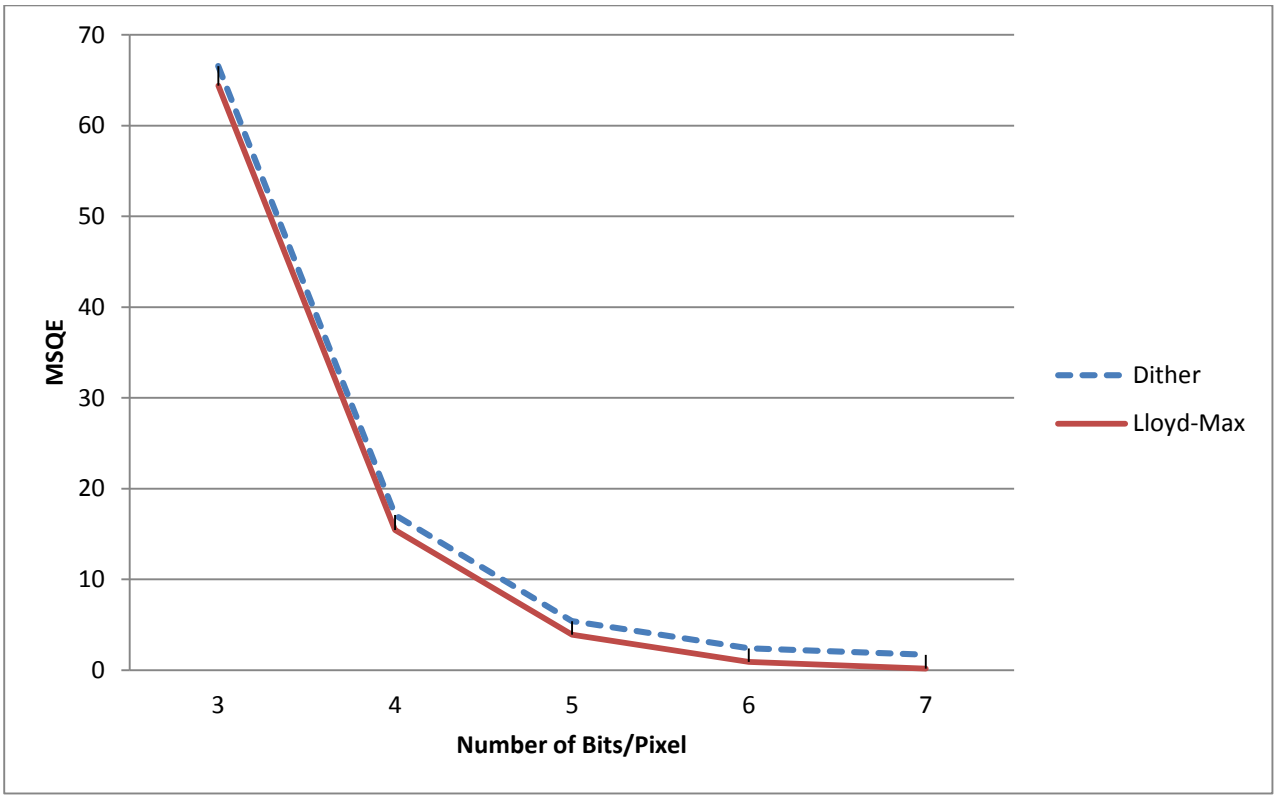


Figure 4.6: Plot of MSQE Values for Dither-based Non-Uniform Quantization and Lloyd-Max Non-Uniform Quantization Schemes

From the analysis of results shown in Figure 4.7, the SQNR values for dither quantization scheme increased from 9.1082 dB to 35.0818 dB between 2 bpp and 7 bpp as opposed to higher SQNR values obtained for Lloyd Max scheme which increased from 9.1403 dB to 44.6083 dB for the same range of bit numbers. Whereas, for 1 bpp, the SQNR value obtained for dither scheme is -1.8637 dB and this value is higher than the value obtained for Lloyd-Max scheme which is -1.9204 dB. This means that the performance of dither-based non-uniform quantization on biometric fingerprint image is not as efficient as the Lloyd-Max approach for higher bpp values but as the bit number used in the quantization process decreases, the performance of dither-based scheme began to improve.

In Figure 4.7, the gap between the SQNR values for dither-based quantization scheme and Lloyd-Max scheme gradually decreased as the number of bits was reduced from 7 bpp until the SQNR values became equal for both schemes at 3 bpp. Therefore, it stands to reason that discrete images with higher bit representation such as 16-bits and 24 bits are best suited for dither-modified quantization scheme because more number of bits will be available for manipulation in the process. However, Lloyd-Max non-uniform quantization is the preferred choice for 8-bit gray-scale biometric fingerprint image lossy compression based on the comparative analysis results.

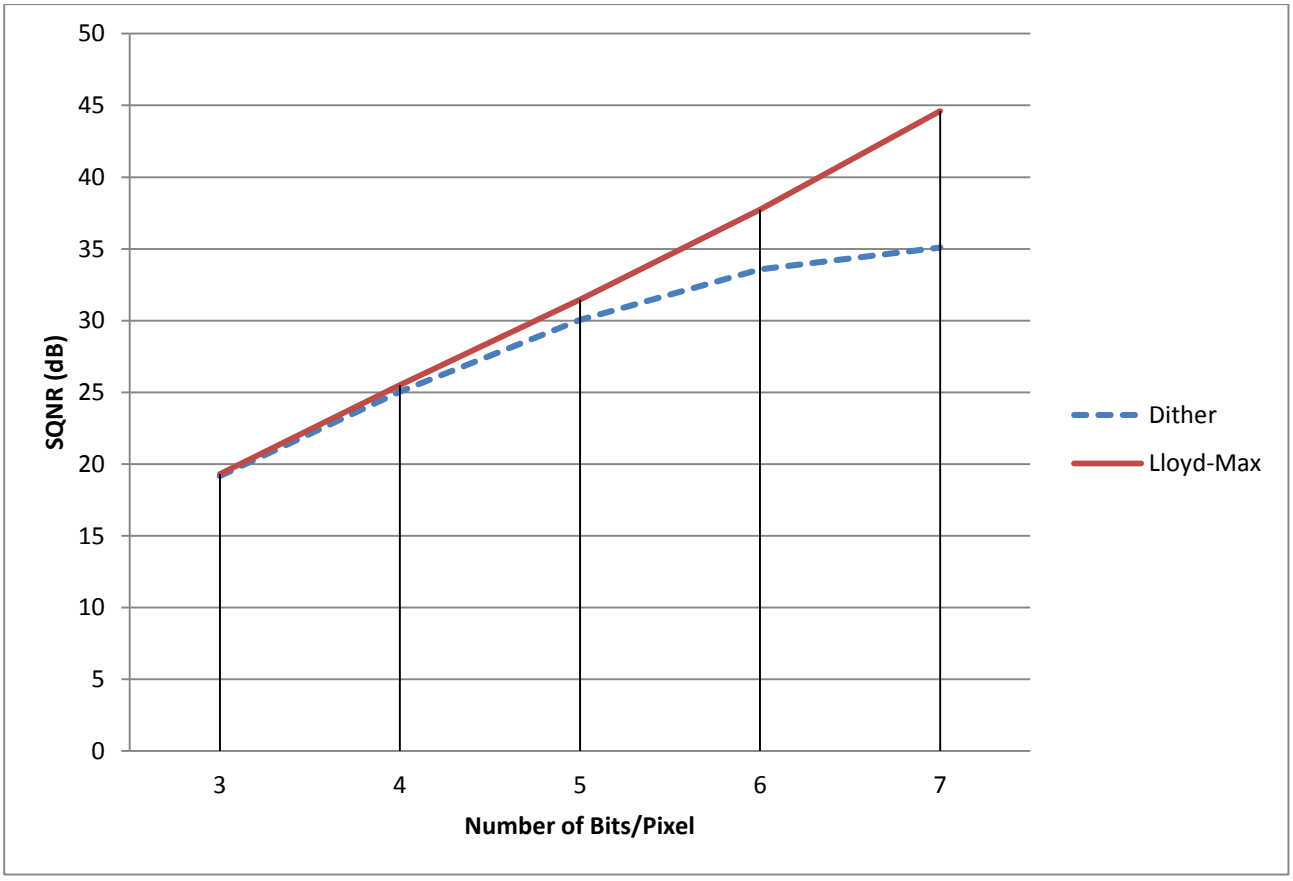


Figure 4.7: Plot of SQNR in Decibels (dB) for Dither-based Non-Uniform Quantization and Lloyd-Max Non-Uniform Quantization Schemes

4.9 Performance Evaluation of the Compression Algorithm

The performance evaluation results of the proposed fingerprint compression algorithm was carried out in terms of compression ratio (CR) and peak signal to noise ratio (PSNR). From

the results of the performance evaluation of the proposed lossy compression algorithm, it was observed that as the compression ratio increased from CR of 12:1 to 80:1, for the fingerprint image sample cmp00001.pgm, the corresponding values of PSNR decreased from 25.3 dB for CR of 15:1 to 9.80 dB for CR of 80:1 for the fingerprint image sample cmp00002.pgm, the corresponding values of PSNR decreased from 32.10 dB for CR of 15:1 to 11.50 dB for CR of 80:1. These results followed the same trend for the other fingerprint images used in the experiment. This is evidently illustrated in the bar chart plot of CR against PSNR shown in Figures 4.8 and 4.9.

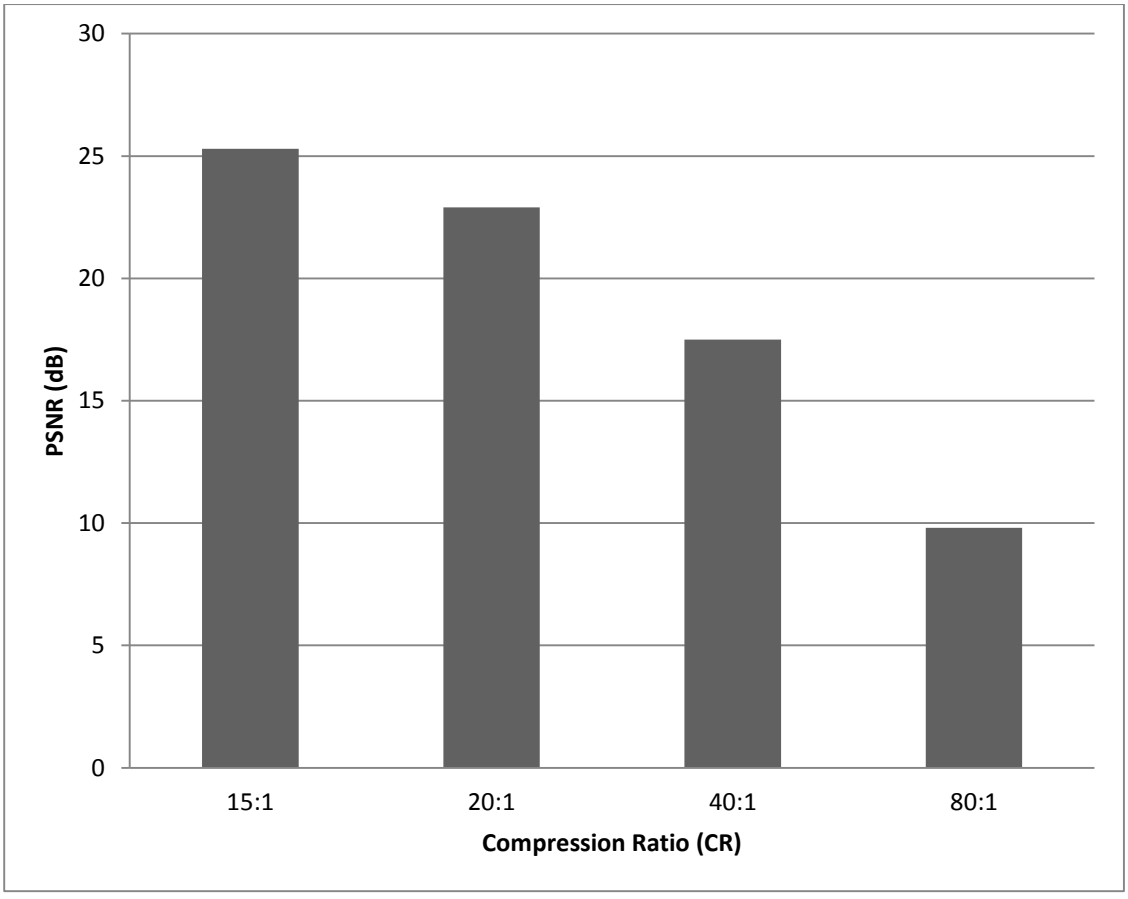


Figure 4.8: Bar Chart Plot of PNSR Values against their Corresponding Compression Ratio (CR) for Fingerprint Image (cmp00001.pgm)

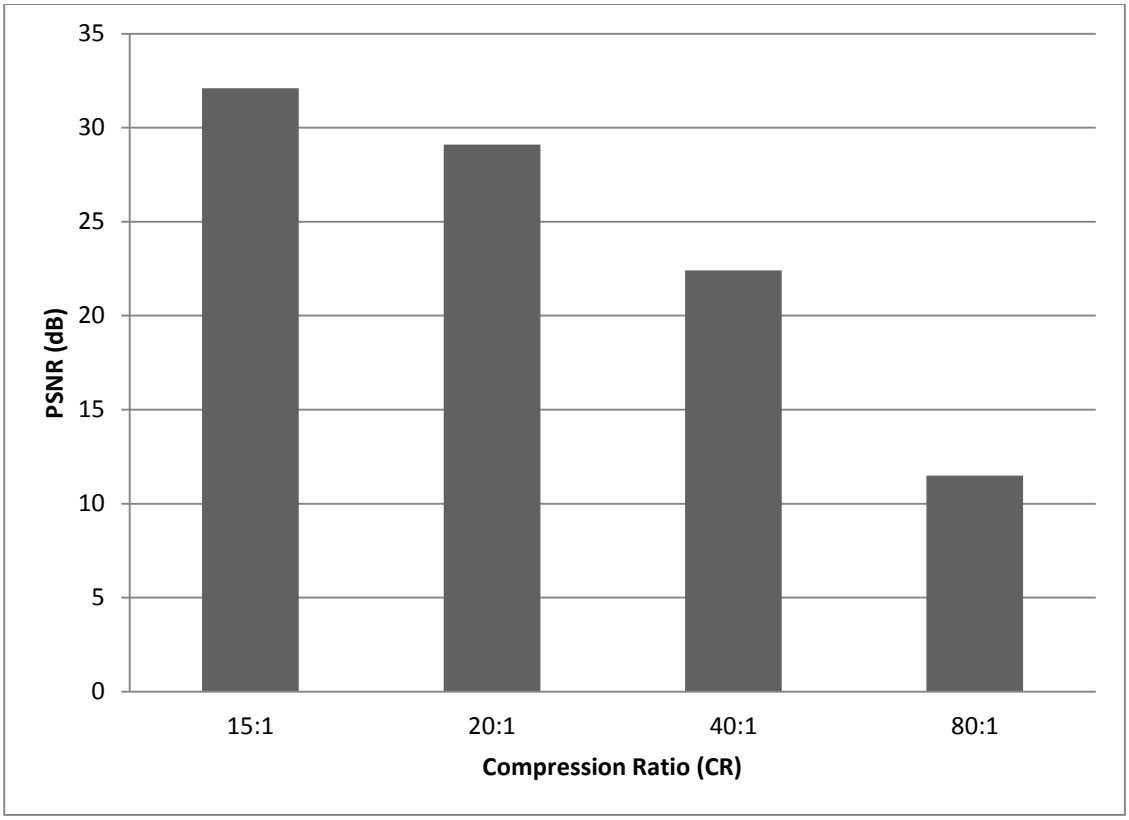


Figure 4.9: Bar Chart Plot of PNSR Values against their Corresponding Compression Ratio (CR) for Fingerprint Image (cmp00002.pgm)

The significance of these results is that as the CR value increased beyond 20:1, the fingerprint images began to lose their biometric features due to increased degradation which resulted from the compression process. However, at compression ratio of 20:1, the compressed fingerprint images still retained their biometric attributes because of the improvement achieved by the use of non-uniform quantization scheme instead of the widely used uniform scalar quantization scheme and this was visually evident from the compressed image output. The minimum PSNR value of 22.90 dB and a maximum PSNR value of 35.5 dB were obtained. This is contrary to the results of the comparative analysis carried out by Libert et al (2012) which revealed that the existing fingerprint compression standard, that is, Wavelet Scalar Quantization (WSQ) produced unsatisfactory results beyond a compression ratio of 15:1 and at CR of 20:1, the biometric features of the compressed images were completely

lost. In addition, the results obtained from this research work show that there is an improvement over the existing fingerprint compression scheme as reported in the work by Ashok et al (2010) in which WSQ produced PSNR values which ranged from 22.52 dB to 29.27 dB for a compression ratio of 15:1.

It was observed that there was variation in the values of PSNR values obtained from the compression of the NIST fingerprint dataset at the same compression ratio. This was due to different level of source fingerprint image quality as the experimental images were acquired at different levels of image quality which was visually evident in the compressed output display. In addition, the presence of background noise also accounted for the variation in the measurement of overall image quality in the compression process. Generally, the overall performance of the proposed compression algorithm achieved an improvement in terms of compression ratio of 20:1 over the existing WSQ fingerprint compression algorithm which achieved a compression ratio of 15:1.

4.10 Contribution to Knowledge

This research work presents an improved approach to biometric fingerprint image compression. The significance of this work is in its innovative application of the concepts of signal processing and information theory to achieve improved fingerprint image compression in terms of compression ratio higher than what is achievable in the existing fingerprint compression standard, that is, Wavelet Scalar Quantization (WSQ) standard. The contributions to academic knowledge by this work include:

i) The attainment of improved compression ratio of 20:1 compared to 15:1 hitherto achieved by WSQ standards which was made possible by trading off some of the added quality realized by using the improved source coding approach;

ii) Improved energy retention efficiency from 98.45% to 99.32% in image transformation process through the replacement of Daubechies wavelet basis in the existing

compression schemes with Coiflet wavelet basis implemented with level dependent thresholds. This approach achieved a better preservation of the energy of the compressed image. Hence, a better image quality that preserves the biometric features of fingerprints. It was shown through experimental analysis that improved energy retention can be achieved in image transformation by the approach adopted in this research work;

iii) Improved image quantization efficiency from 40.13 dB to 44.60 dB by employing non-uniform quantization scheme as against the uniform scheme used by the existing compression schemes. Non-uniform quantization produced lesser distortion or degradation in the quantized image signal. It was also shown by comparative analysis that quantization error can be better minimized through non-uniform scalar quantization approach adopted in this work as opposed to the uniform scalar quantization used in the existing compression algorithm.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

In this research work, an improved approach to source coding for lossy compression of biometric fingerprint images was developed using a novel approach. The innovative approach employed in the development of the improved algorithm involved:

(i) the transformation of biometric fingerprint image with Coiflet wavelet implemented with level dependent threshold strategy instead of the global threshold strategy commonly used ;

(ii) the use of non-uniform scheme for the quantization of transformed fingerprint images to minimize image signal distortion as against the widely used uniform scalar quantization scheme; and

(iii) the use of arithmetic entropy coding to reduce coding redundancy in the compression process.

Comparative analysis of Coiflet wavelets implemented with level dependent thresholds and Daubechies wavelets were conducted on the basis of percentage retained energy, RE (%). The RE (%) values for Coiflet wavelet ranged from 99.32% to 99.69% as opposed to the values for Daubechies wavelet which ranged from 98.45% to 99.15%. These results revealed that the Coiflet wavelet bases performed better than the Daubechies wavelet and that the level-dependent threshold strategy performed better than the global threshold technique. Hence, the choice of Coiflet wavelet basis with level-dependent threshold method for image transformation in the proposed compression algorithm was justified. The performance analysis of uniform and non-uniform scalar quantization schemes for biometric fingerprint image compression was conducted. The non-uniform quantization method based on Lloyd-Max approach performed better than the uniform quantization method used in the existing

fingerprint compression standard. The Signal to Quantization Noise Ratio (SQNR) values for non-uniform quantization increased from 19.2977 dB for 3 bit per pixel (bpp) to 44.6083 dB for 7 bpp whereas for the same range (3 bpp to 7 bpp) for uniform quantization, SQNR values increased from 17.0903 dB to 40.1349 dB. Therefore, based on the performance results, it was concluded that the future of the development of lossy compression techniques for fingerprint image lies with the use non-uniform scalar quantization scheme.

The overall performance of the compression algorithm was evaluated using Peak Signal to Noise Ratio (PSNR) and Compression Ratio (CR) as performance metrics. PSNR was used as a measure of compressed image quality and it was estimated as a function of the mean square error (MSE) which is a measure of the image distortion that resulted from the compression process. The Compression Ratio was used as a measure of the degree of compression achievable to evaluate the performance of the proposed compression algorithm against the achievable compression ratio for the existing fingerprint compression schemes. In the system performance analysis, a trade-off was made between the achievable compression ratio and the allowable degradation which is a function of achievable PSNR in the compression process. Put differently, the additional improvements achieved by the use of non-uniform quantization scheme and the application of level-dependent threshold method for wavelet transformation were sacrificed to achieve additional compression in the process. Compression ratio values were computed with the intent of maintaining compressed fingerprint image quality consistent with the requirement to preserve the biometric features in the fingerprint images. From the performance analysis results, the overall performance of the proposed compression algorithm achieved an improvement in terms of compression ratio of 20:1 over the existing algorithms which achieved a compression ratio of 15:1 for biometric application. This overall improvement was largely due to the improvement achieved at the transformation stage by employing Coiflet wavelet basis for image energy decorrelation as

opposed to Daubechies wavelet in the existing system. In addition, the improvement achieved at the quantization stage through the use of non-uniform quantization as opposed to uniform scalar quantization used in the existing compression algorithm also contributed to the overall performance.

This research work encountered the following challenges:

(i) the non-availability of raw-bits and uncompressed fingerprint image acquisition device. As a result, uncompressed fingerprint dataset were obtained from the database of the National Institute of Standards and Technology (NIST), USA for experimental analysis.

(ii) the presence of background noise in some of the experimental fingerprint images obtained from NIST which accounted for the differences in the overall Peak Signal to Noise Ratio (PSNR) values obtained for fingerprint images at the same compression ratio.

5.2 Recommendations

Recommendations for further research work are as follows:

i) The development of an improved approach to channel coding for efficient management of transmission bandwidth in biometric fingerprint application;

ii) Estimation of computation efficiency of source coding of fingerprint images in terms of processing time.

Further research in these areas is pertinent because transmission bandwidth and computation resources are limited. Hence, the need for efficient methods to reduce transmission bandwidth and computation costs is also important.

REFERENCES

Ashok, J., Shailaja, T. V., Somayajula, S. P. K. (2010) Wave atoms decomposition based fingerprint image compression, IJCSNS, Vol. 10, No. 9, pp. 57-61

Birge, L. and Massart, P. (1997) From model selection to adaptive estimation; In Festschrift for Lucien Le Cam: Research papers in probability and statistics, 55-88, Springer-Verlag, New York: Retrieved from http://www.stat.yale.edu/~pollard/ Books/LeCamFest/BirgeMassart.pdf on 05/02/2014

Bodden, E, Clasen and Kneis, J. (2007) Arithmetic coding revealed, Sable Technical, No. 2007-5; Retrieved from www.sable.mcgill.ca on 12/07/2014

Chang, C. C., Chen, S. G. and Chiang, J. C. (2007) Efficient encoder design for JPEG2000 EBCOT context formulation, 15th European Signal Processing Conference, EUSIPCO, Poznan, pp. 644-648

CJIS (2000) WSQ Gray-scale fingerprint image compression specification, Criminal Justice, Information Services Division, FBI, USA: Accessed from https://www.fbibiospecs.org/docs/WSQ_Gray-scale_Specification_Version _3_1_ Final.pdf on 22/08/2011

Daubechies, I. (1992) *Ten Lectures on Wavelets*, Society for Industrial and Applied Mathematics (SIAM), Philadephia, Pennsylvania, pp121-261

Deepa, T. and Girisha, H. (2014) Image compression using Hybrid wavelet Transform and their Performance Comparison; International Journal of Modern Engineering Research (IJMER), Vol. 4, Issue 6, pp6-12; Retrieved from http://www.ijmer.com/papers/Vol4 _Issue6/Version-2/IJMER-46020612.pdf on 09/12/2014

Dixit, A. and Majumdar, S. (2013) Comparative analysis of coiflet and daubechies wavelets using global threshold for image de-noising; International Journal of Advances in Engineering & Technology, IJAET, Vol. 6, Issue 5, pp. 2247-2252: Retrieved from http://www.e-ijaet.org/media/32I17-IJAET1117356_v6_iss5_2247-2252.pdf on 21/03/2014

Donoho, D. and Johnstone, I. (1998) Minimax estimation via wavelet shrinkage; The Annals of Statistics, Vol. 26, No. 3, pp879-921: Retrieved from http://projecteuclid.org:80 /Dienst/getRecord?id=euclid.aos/1024691081/ on 05/01/2014

Gangwar, D. K. (2012) Compression and development analysis on an image using Haar wavelet transform, IJATER, Vol. 2 No. 3, pp225-229

George J. P., Abhilash S. K., Raja K. B., (2012). Transform Domain Fingerprint Identification Based on DTCWT, International Journal of Advanced Computer Science and Applications (IJACSA), *3*(1), pp190-195.

Giurgiutiu, V. and Lingyu Yu, L. (2003) Comparison of Short-time Fourier Transform and Wavelet Transform of Transient and Tone Burst Wave Propagation Signals For Structural Health Monitoring, Proceedings of the 4th International Workshop on Structural Health Monitoring, Stanford University; pp122-134

Gonzalez, R. C. and Woods, R. E. (2002*) Digital Image Processing*, 2rd Edition, Prentice Hall, Upper Saddle River, New Jersey, pp. 411-486

Haddad, Z., Beghdadi, A. Serir, A. and Mokraoui, A. (2009) A new fingerprint image compression based on wave atoms transform; IEEE Transaction Signal Processing; Vol. 6 No. 9, pp.89-94

Howard, P. G. and Vitter, J. S. (1994) Arithmetic coding for data compression, Proceedings of the IEEE Transaction on Information Theory, Vol. 82, No. 6, pp. 857-865

Hsin, H. C., Lien, J. J. and Sung, T. Y. (2007) A hybrid SPIHT-EBC image coder, International Journal of Computer Science, Vol. 34, No. 1, pp15-26

Islam, M. R. et al (2012) Performance analysis of Coiflet-type wavelets for a fingerprint image compression by using wavelet and wavelet packet transform, International Journal of Computer Science & Engineering Survey(IJCSES), Vol. 3, No. 2, pp. 79-87; Retrieved from http://airccse.org/journal/ijcses/papers/ 3212ijcses09.pdf on 29/02/2013

Iqbal, F. (2004) Wavelet transform based image compression on FPGA, MSc Thesis, University of Florida, USA: Retrieved from http://diginole.lib.fsu.edu/cgi/viewcontent.cgi?article=2179&context=etd on 07/06/2013

Kazi, M. M., Mane, A. V., Manza, R. R. and Kale, K. V. (2011) Comparison of fingerprint enhancement techniques through mean square error and peak-signal to noise ratio, IJCSE, Vol.3No1, pp. 266-270

Kumar, C., Shekhar, C., Das, S. and Bhandari, S. (2010) Compression algorithm by using wavelets, International Journal of Engineering Science and Technology, Vol. 2, No. 10, pp. 4978-4982

Libert, J. M., Orandi, S. and Grantham J. D. (2012) Comparison of the WSQ and JPEG 2000 Image Compression Algorithms On 500 ppi Fingerprint Imagery, NIST Interagency Report 7781, Accessed from http://www.nist.gov/customcf/get_pdf.cfm?pub_id=910658 on 23/04 2013

Luo, G. and Zhang, D. (2012) Wavelet Denoising: Advances in Wavelet Theory and Their Applications in Engineering, Physics and Technology, DumitruBaleanu (Ed.), ISBN: 978-953-51-0494-0, InTech, Retrieved from: http://www.intechopen.com/books/advances-in-wavelet-theory-and-their-applications-in-engineering-physics-and-technology/wavelet-denoising on 05/02/2014

McAndrew, A. (2004) An introduction to digital image processing with MATLAB, School of Computer Science and Mathematics, Victoria University of Technology: Retrieved from http://visl.technion.ac.il/labs/anat/An%20Introduction%20To%20Digital%20Image%20Processing%20With%20Matlab.pdf on 30/01/2014

Mallat, S. (2009) A wavelet tour of signal processing: The sparse way, 3rd Edition, Elsevier Inc., Burlington MA, USA: pp.122-212

Misiti, M., Misiti, Y., Oppenheim, G. and Poggi, J. M. (1996) Wavelet Toolbox for use with MATLAB; User Guide, The MathWork, USA: pp.56-123

Muhsen, Z., Dababneh, M., Al Nsour, A. (2008) Requantization codebook using fingerprint; Science Publications, Journal of Computer Science 4 (11): pp959-962

Muhsen, Z., Dababneh, M., Al Nsour, A. (2011) Wavelet and optimal re-quantization methodology for lossy fingerprint compression, International Arab Journal of Information Technology, Vol. 8, No. 4, pp. 383-387

NIST (2011) WSQ Fingerprint Image Compression Encoder/Decoder Compliance Guidelines, National Institute of Standards and Technology, Accessed from http://nigos.nist.gov:8080/wsq/reference_images_v2.0_pgm.tar on 21/03/2013

Onyshczak, R. and Youssef, A. (2000) Fingerprint Image Compression and the Wavelet Scalar Quantization Specification, Technical Report, National Institute of Standards and Technology, U.S.A., pp. 1-32

Padmaja, G.M. and Nirupama, P. (2012) Analysis of Various Image Compression Techniques, ARPA Journal of Science and Technology, Vol. 2 No. 4, pp 371-376

Peric, Z., Nikolic, J. and Pokrajac, D. (2007) Estimation of the support for Laplacian source scalar quantizers; Journal of Electrical Engineering, Vol. 58, No. 1, pp. 47-51: Retrieved from http://iris.elf.stuba.sk/JEEEC/data/pdf/1_107-07.pdf on 15/03/2014

Polikar R. (2001) The engineer's ultimate guide to wavelet analysis: The wavelet tutorial, Rowan University: Accessed from http://users.rowan.edu/~polikar/WAVELETS/WTtutorial.html on 11/05/2011

Polikar R. (2004) Processing of biological signals, Rowan University, Retrieved from http://engineering.rowan.edu/~polikar/CLASSES/ECE404 on 11/05/2011

Pu, I. M. (2006) *Fundamental Data Compression*; Butterworth-Heinemann, Burlington, MA; pp189-206

Rajkumar, R., Meetei, T. C., Begum, S. A., Hemachandra, K. (2012) Fingerprint enhancement and its features purification, Journal of Computing, Vol. 4 no. 5, May 2012, pp. 78-82

Rathee, M. and Vij, A. (2014) Image compression Using Discrete Haar Wavelet Transforms; International Journal of Engineering and Innovative Technology (IJEIT) Volume 3, Issue 12, pp47-55; Retrieved from http://ijeit.com/Vol%203/Issue%2012/IJEIT 1412201406_09.pdf on 05/12/2014

Rawat, C. S. D. and Meher, S. (2009) A hybrid coding scheme combining SPIHT and SOFM based vector quantization for effective image compression, European Journal of Scientific Research, Vol. 38, No. 3, pp. 425-440

Said, A., and Pearlman, W. (1996) A new fast and efficient image codec based on set partitioning, IEEE Trans. Circuits Syst., Vol. 6, pp. 243-250

Said, A. (2004) *Introduction to Arithmetic Coding - Theory and Practice*; Imaging Systems Laboratory; Academic Press; HP Laboratories Palo Alto; pp1-45

Salomon, D. and Motta, G. (2010) *Handbook of data compression*, 5th Edition, Springer-Verlag, London: pp.52-112

Salomon, D. (2007) *Data Compression: The Complete Reference*; 4th Edition, Springer-Verlag, London: pp. 117-154

Sayood, K. (2012) *Introduction to data compression*, Third Edition, Morgan Kaufmann - Elsevier, Massachusetts, USA: pp. 35-97

Selvakumarasamy, K., Radhikadevi, Ramya and Sagunthala (2013) Performance analysis of Biorthogonal wavelets for fingerprint image compression; AMS College of Engineering; Vol.1, No.4, pp.64-70

Shakhakarmi, N. (2012) Quantitative multiscale analysis using different wavelets in 1D voice signal and 2D image, IJCSI International Journal of Computer Science, Vol 9, No. 2, Accessed from http://www.IJCSI.org on 22/04/ 2013

Shanavaz, K.T. and Mythili, P. (2012) Faster techniques to evolve wavelet coefficients for better fingerprint image compression, International Journal of Electronics, Vol. 100 No 5, accessed from http://www.tandfonline.com/doi/pdf/10.1080/00207217. 2012.720944 on 22/04/ 2013

Shapiro, J. M. (1993) Embedded image coding using zerotrees of wavelet coefficients, IEEE Transactions on Signal Processing, Vol. 41, No. 12, December, 1993, pp. 3445-3462

Selvakumarasamy, K. et al (2013) Performance Analysis of biorthogonal wavelets for fingerprint image compression, JECE, Vol. 1, No 4, pp2320-8945

Shakhakarmi, N. (2012) Quantitative multiscale analysis using different wavelets in 1D voice signal and 2D image, IJCSI International Journal of Computer Science, Vol 9, No. 2, Accessed from http://www.IJCSI.org on 22/04/2013

Shanavaz, K.T. and Mythili, P. (2012) Faster techniques to evolve wavelet coefficients for better fingerprint image compression, International Journal of Electronics, Vol. 100 No 5, accessed from http://www.tandfonline.com/doi/pdf/10.1080/00207217.2012.720944 on 22/04/2013

Singla, N. and Sharma, S (2013) A Review on Wavelet based Compression using Medical Images; International Journal of Innovative Research in Computer and Communication Engineering, IJIRCCE; Vol. 1, Issue 8; pp 1649-1653: Retrieved from http://ijircce.com/upload/2013/october/7AReview.pdf on 23/03/2014

Srivastava, M., Yashu, Y., Singh, S. K. and Panigrahi, P. K. (2011) Multisegmentation through wavelets: Comparing the efficacy of Daubechies vs Coiflets; Conference Proceedings on Signal Processing and Real Time Operating System, March 25-26, 2011: Retrieved from http://arxiv.org/ftp/arxiv/papers/1207/1207.5007.pdf on 23/03/2014

Srivastava, M., Singh, S. K. and Panigrahi, P. K. (2014) An adaptive statistical non-Suniform quantizer for detail wavelet components in lossy JPEG2000 image compression; Proceedings of the International Conference on Pattern Recognition Applications and Methods, pp1-26; Retrieved from http://arxiv.org/pdf/1305.1986.pdf on 05/01/2014

Strang, G. (1994) Wavelets, American Scientist 82, Technical Report, pp250-255; Accessed from *www-math.mit.edu/~gs/papers/amsci.ps.gz* on 17/04/2012

Strang, G. and Nguyen, T. (1996) Wavelets and filter banks, Wellesley-Cambridge Press, Massachusetts, USA, pp1-28, 221-259, 262-275

Stroud, K. A. and Booth, D. J. (2007) *Engineering Mathematics*, Siixth Edition, Palgrave Macmillan, New York, pp234-241

Sweldens, W. (1996) The lifting scheme: A construction of second generation wavelets, SIAM Journal on Mathematical Analysis, pp1-42

Taubman, D. (2000) High performance scalable image compression with EBCOT, IEEE Trans on Image Processing, Vol. 9, No. 7, pp. 1158-1170

Thyagarajan, K. S. (2011) *Still image and video compression with MATLAB*, John Wiley and Sons Inc, New Jersey; pp4-16, 131-145, 259-282

Usevitch, B. E. (2001) A tutorial on modern lossy wavelet image compression: Foundations of JPEG 2000, IEEE Signal Processing Magazine, pp. 22-35

Villegas O. V, Domínguez, H. O., Sanchez, V. G., Maynez, L. O. and Orozco, H. M. (2006) Biometric Human Identification of Hand Geometry Features Using Discrete Wavelet Transform. In Olkkonen, H. (ed), Discrete Wavelet Transforms – Biomedical Applications, pp 251-266.

Wei, D. (1998) Coiflet-type wavelets: Theory, design, and applications, PhD Desertation, University of Texas, Austin: Retrieved from http://esrdc.mit.edu/library/ ESRDC_library/Wei_Dong_PhD_Thesis.pdf on 10/06/2014

Xing-Yuan, W., Jiao-Jiao, Y. and Yong-Lei, Z. (2011) An efficient adaptive arithmetic coding image compression technology, Chinese Physical Society and IOP, Vol. 20, No. 10, pp. 104203-1 – 104203-7.

Zhao, S. And Wang, X. (2009) Fingerprint image compression based on directional filter banks and TCQ; IEEE Computer Society; Second International Workshop on Knowledge Discovery and Data Mining; pp660-663

APPENDIX I

The Coefficients for Orthogonal Coiflets of order L = 2K, K = 1 to 5 (Daubechies, 1992)

	n	$h_n/\sqrt{2}$
$K=1$	−2	−.051429728471
	−1	.238929728471
	0	.602859456942
	1	.272140543058
	2	−.051429972847
	3	−.011070271529
$K=2$	−4	.011587596739
	−3	−.029320137980
	−2	−.047639590310
	−1	.273021046535
	0	.574682393857
	1	.294867193696
	2	−.054085607092
	3	−.042026480461
	4	.016744410163
	5	.003967883613
	6	−.001289203356
	7	−.000509505399
$K=3$	−6	−.002682418671
	−5	.005503126709
	−4	.016583560479
	−3	−.046507764479
	−2	−.043220763560
	−1	.286503335274
	0	.561285256870
	1	.302983571773
	2	−.050770140755
	3	−.058196250762
	4	.024434094321
	5	.011229240962
	6	−.006369601011
	7	−.001820458916
	8	.000790205101
	9	.000329665174
	10	−.000050192775
	11	−.000024465734
$K=4$	−8	.000630961046
	−7	−.001152224852
	−6	−.005194524026
	−5	.011362459244
	−4	.018867235378
	−3	−.057464234429
	−2	−.039652648517
	−1	.293667390895

	n	$h_n/\sqrt{2}$
$K=4$	0	.553126452562
	1	.307157326198
	2	−.047112738865
	3	−.068038127051
	4	.027813640153
	5	.017735837438
	6	−.010756318517
	7	−.004001012886
	8	.002652665946
	9	.000895594529
	10	−.000416500571
	11	−.000183829769
	12	.000044080354
	13	.000022082857
	14	−.000002304942
	15	−.000001262175
$K=5$	−10	−.0001499638
	−9	.0002535612
	−8	.0015402457
	−7	−.0029411108
	−6	−.0071637819
	−5	.0165520664
	−4	.0199178043
	−3	−.0649972628
	−2	−.0368000736
	−1	.2980923235
	0	.5475054294
	1	.3097068490
	2	−.0438660508
	3	−.0746522389
	4	.0291958795
	5	.0231107770
	6	−.0139736879
	7	−.0064800900
	8	.0047830014
	9	.0017206547
	10	−.0011758222
	11	−.0004512270
	12	.0002137298
	13	.0000993776
	14	−.0000292321
	15	−.0000150720
	16	.0000026408
	17	.0000014593
	18	−.0000001184
	19	−.0000000673

APPENDIX II

MATLAB CODES IMPLEMENTATION

Biometric Fingerprint Image Compression using wavelet

```
function waveletcoif_demo1
% FingerprintImage Compression System demo %%%lc
% Source image is transformed, quantized and entropy coded..
% Significant image symbols are retained and insignificant ones are
% discarded using different compression ratios ...

clc
clear all
close all

% Input fingerprint image file ...
ImageFile = input('Enter a source image: ');

% Load the fingerprint image
%X = double(imread(ImageFile))/255;
X = double(imread(ImageFile));
N = size(X);
figure; colormap(gray(256));
subplot(2,1,1);
image(X);
axis image; axis off;
title(sprintf('Original (%dx%d)',N(2),N(1)));
drawnow;
%X = RGBToYCbCr(X);      % Use YCbCr colorspace for processing
subplot(2,2,3);
Y = waveletcdf97(X(:,:,1),1);  % 1-stage transform of the intensity channel
imagesc(abs(Y).^0.5);
axis image; axis off;
title('1-stage transform');
drawnow;
subplot(2,2,4);
Y = waveletcdf97(X(:,:,1),3);  % 3-stage transform
imagesc(abs(Y).^0.5);
axis image; axis off;
title('3-stage transform');
drawnow;
L = 3;
Y = waveletcdf97(X,L);
% figure;colormap(gray(256));
% drawnow;
% subplot(1,2,1);
%
% R = waveletcdf97(keep(Y,1/5),-L);
% %image(YCbCrToRGB(R));
%
% image(R);
% axis image; axis off;
% title(sprintf('CR=5:1, PSNR %.1f dB',psnr(X,R)));
% drawnow;
%
% subplot(1,2,2);
% R = waveletcdf97(keep(Y,1/12),-L);
% %image(YCbCrToRGB(R));
```

```
% image(R);
% axis image; axis off;
% title(sprintf('CR=12:1, PSNR %.1f dB',psnr(X,R)));
% drawnow;
% return;



% Apply non-uniform quantization and arithmetic entropy coding



figure;colormap(gray(256));
drawnow;
subplot(2,2,1);
R = waveletcdf97(keep(Y,1/15),-L);
%image(YCbCrToRGB(R));
image(R);
axis image; axis off;
title(sprintf('CR=15:1, PSNR %.1f dB',psnr(X,R)));
drawnow;

subplot(2,2,2);
R = waveletcdf97(keep(Y,1/20),-L);
%image(YCbCrToRGB(R));
image(R);
axis image; axis off;
title(sprintf('CR=20:1, PSNR %.1f dB',psnr(X,R)));
drawnow;
subplot(2,2,3);
R = waveletcdf97(keep(Y,1/40),-L);
%image(YCbCrToRGB(R));
image(R);
axis image; axis off;
title(sprintf('CR=40:1, PSNR %.1f dB',psnr(X,R)));
drawnow;

subplot(2,2,4);
R = waveletcdf97(keep(Y,1/80),-L);
%image(YCbCrToRGB(R));
image(R);
axis image; axis off;
title(sprintf('CR=80:1, PSNR %.1f dB',psnr(X,R)));
drawnow;
return;



% Apply non-uniform quantization and arithmetic entropy coding



function X = keep(X,Ratio)
% Set all but the largest elements of X to zero

if size(X,3) > 1
  NumElements = prod(size(X));
  % Number of components to keep in the Y channel
  NumY = floor(NumElements*Ratio*0.92);
  % Number of components to keep in the Cb channel
  NumCb = floor(NumElements*Ratio*0.04);
  % Number of components to keep in the Cr channel
```

```
  NumCr = floor(NumElements*Ratio*0.04);

  X(:,:,1) = keep(X(:,:,1),NumY);
  X(:,:,2) = keep(X(:,:,2),NumCb);
  X(:,:,3) = keep(X(:,:,3),NumCr);
  return;
end

if Ratio > 1
  Num = Ratio;
else
  Num = floor(prod(size(X))*Ratio);
end

[tmp,i] = sort(abs(X(:)));
X(i(1:end-Num)) = 0;
return;



function p = psnr(A,B)
% Compute the PSNR value of B relative to A
p = -10*log10(mean((A(:) - B(:)).^2) / (max(A(:)) - min(A(:)))^2);
return;
```

Lloyd Max Non-uniform Quantization

```
% Design of a Non-uniform Quantizer for a given input image
% Starts with an initial set of output levels and
% iteratively builds a final set that results in the
% least mean square error between the input and
% reconstructed image
clear
close

A = imread('cmp00001.pgm');
 [Height,Width,Depth] = size(A);
if Depth == 1
f = double(A);
else
f = double(A(:,:,1));
end
%
%figure,imshow(A(60:275,302:585,1));% cropped for Barbara image
% figure,imshow(A(:,:,1));
figure,imshow(uint8(f));
title('Original 8 bpp image')
%
sigmaf = std2(f); % std dev of input image
B = 7; % bits of quantization
snr = zeros(B,1); % signal-to-quantization noise in dB
err = 0.1e-01; % accuracy to terminate iteration

% for b = 5:5

for b = 5:5
L = 2^b; % number of output levels
%C = linspace(0.5,255,L); % initial output levels
C = linspace(2,253,L);
PartitionRegion = zeros(L,Height,Width);
% partitions consisting of input pixels
```

```
RegionCount = zeros(L,1); % number of elements in each partition
RegionSum = zeros(L,1); % sum of pixels in a given partition region
Distortion = zeros(L,1); % minimum distortion in a partition region
TotalDistortion = 1.0e+5; % total minimum distortion
%TD = abs(TotalDistortion-sum(Distortion)); % initial total distortion
% alternate convergence measure
TD = (TotalDistortion-sum(Distortion))/TotalDistortion;
% start iteration
It = 1; % iteration number
while TD >= err
for r = 1:Height
for c = 1:Width
d1 = abs(f(r,c)-C(1));
J = 1;
 for k = 2:L
d2 = abs(f(r,c)-C(k));
if d2<d1
d1 = d2;
J = k;
end
end
RegionCount(J) = RegionCount(J) + 1;
RegionSum(J) = RegionSum(J) + f(r,c);
PartitionRegion(J,r,c) = f(r,c);
Distortion(J) = Distortion(J)+(f(r,c)-C(J))*(f(r,c)-C(J));
end
end
Index = logical(RegionCount == 0);% check for empty regions
RegionCount(Index) = 1;% set empty regions to count 1
Distortion = Distortion ./RegionCount;% average minimum distortions
%TD = abs(TotalDistortion-sum(Distortion));
% alternate convergence measure
TD = (TotalDistortion-sum(Distortion))/TotalDistortion;
C = RegionSum ./RegionCount; % refined output levels
TotalDistortion = sum(Distortion);% total minimum distortion
It = It + 1;
end
clear Index
sprintf('Number of iterations = %d; \t bpp = %d',It-1,b)
%sprintf('SNR = %4.2f dB',10*log10(sigmaf*sigmaf/TotalDistortion))
DR = zeros(L+1,1);
DR(L+1) = 255;
f1 = zeros(size(f));
for k = 2:L
DR(k) = (C(k-1)+C(k))/2;
end
% plot output levels against decision regions
% if b == 5
if b == 5
figure,stairs(DR(1:L),C,'k','LineWidth',2)
title(['Nonuniform quantizer: ' num2str(b) ' bpp'])
xlabel('Decision Regions')
ylabel('Output Levels')
figure,subplot(2,1,1),stem(RegionCount/(It-1),'k')
title(['Comparison of output & input histograms at ' num2str(b) ' bpp'])
ylabel('Counts')
[Counts,Bins] = imhist(uint8(A),32);
subplot(2,1,2),stem(Counts,'k')
xlabel('Level number'), ylabel('Counts')
end
```

```
% decode using the codebook generated
for r = 1:Height
for c = 1:Width
%d1 = abs(f(r,c)-C(1));
d1 = abs(f(r,c)-C(1));
for k = 2:L
%d2 = abs(f(r,c)-C(k));
d2 = abs(f(r,c)-C(k));
if d2<d1
d1 = d2;
J = k;
end
end
f1(r,c) = C(J);
end
end
% decode using decision regions & codebook
%{
for k = 1:L
[x,y] = find(f>DR(k) & f<=DR(k+1));
if ?isempty(x)
for j = 1:length(x)
f1(x(j),y(j)) = C(k);
end
end
clear x y
end
%}

%if b == 3 || b == 5
if b == 7

%figure,imshow(uint8(f1(60:275,302:585)));% display requantized image
figure,imshow(uint8(f1))
title(['Requantized image at ' num2str(b) ' bpp'])
end
mse = sum(sum((f-f1).*(f-f1)))/(Height*Width);
snr(b) = 10*log10(sigmaf*sigmaf/mse);
end
figure,plot(1:B,snr,'k','LineWidth',2)
title('SNR Vs Bits/pixel of a nonuniform quantizer')
xlabel('# bits')
ylabel('SNR(dB)')
```

Arithmetic coding Algorithm implementation

```
% This algorithm encodes image source symbol sequence ‘‘m’’ using arithmetic coding.

clear

close

m = imread(cmp00001.pgm);% symbol sequence to be coded

p = Hist(m);% probability vector of input alphabet

if any(find(p<0))

error('Negative value for probability\n')

end

if abs(sum(p)-1) >1.0e-10
```

```
error('Not a valid probability set\n')
end
%TB is the significant number of bits of the Tag
TB = ceil(-log2(prod(p(m))))+1;
sprintf('Precision of Tag = %g bits\n',TB)
sprintf('Entropy = %4.3f;\tAvg. Code Length = %4.3f\n',sum(-p.*log2(p)),TB/length(m))
sprintf('Efficiency = %4.2f\n',(100*sum(-p.*log2(p))/TB*length(m)))
% compute the cumulative probability function (CDF) of p
CDF = zeros(1,length(p));
for i = 1:length(p)
CDF(i) = sum(p(1:i));
end
%
% Compute the Tag corresponding to the symbol sequence
% by calling the function ‘‘ArithmeticEncode’’
Tag = ArithmeticEncode(CDF, m);
% generate truncated code for the tag
AC = uint16(0);% contains the arithmetic code
k = 1;
T = Tag;
while k <= TB
T = 2*T;
if T >= 1
AC = bitshift(AC,1);
AC = AC + uint16(1);
T = T -1;
else
AC = bitshift(AC,1);
end
k = k + 1;
end
% Now decode the symbol sequence
T1 = 0.0;
AC1 = AC;
TB1 = TB + 1;
for k = 1:TB
T1 = double(bitand(AC1,1))*(2^(k-TB1)) + T1;
AC1 = bitshift(AC1,-1);
end
sprintf('Tag = %10.8f\t\tTruncated Tag = %10.8f\t\tArithmetic code = %g\n', Tag,T1,AC)
```

```
%
D = ArithmeticDecode(CDF,T1,length(m));
sprintf('\nEncoded Symbol Sequence :\n')
sprintf('%g\t',m)
sprintf('\nDecoded Symbol Sequence :\n')
sprintf('%g\t',D)
function Tag = ArithmeticEncode(CDF,m)
Tag = ArithmeticEncode(p);
% Arithmetic coding procedure
% p is the probability vector of the source alphabet
% m is the symbol sequence to be coded
% CDF is the cumulative distribution
Low = 0; High = 1;% initial values of lower & upper intervals
%l and u are the next lower & upper intervals
k = 1;
while k <= length(m)
if m(k) == 1
Lower = Low;
else
Lower = Low + (High-Low)*CDF(m(k)-1);
end
Upper = Low + (High-Low)*CDF(m(k));
Tag = (Upper+Lower)/2;
Low = Lower; High = Upper;
k = k + 1;
end
function DecodedSymbol = ArithmeticDecode(CDF,Tag,L)
DecodedSymbol = ArithmeticDecode(m);
% Given the CDF, TAG, and the length of the
% symbol sequence, the function returns the
% symbol sequence in ‘‘DecodedSymbol’’
%
Low = 0; High = 1;% initialize lower & upper intervals
DecodedSymbol = int16(zeros(1,L));
k = 1;
while k <= L
T = (Tag -- Low)/(High-Low);
for j = 1:length(CDF)
if j == 1
if T >= 0 && T<CDF(j)
```

```
DecodedSymbol(k) = j;
break
end
else
if T >= CDF(j-1) && T<CDF(j)
DecodedSymbol(k) = j;
break
end
end
end
if DecodedSymbol(k) == 1
Lower = Low;
else
Lower = Low + (High-Low)*CDF(DecodedSymbol(k)-1);
end
Higher = Low + (High-Low)*CDF(DecodedSymbol(k));
Low = Lower; High = Higher;
k = k + 1;
end
```