

**IMAGE
COMPRESSION
USING WAVELETS**

By

MOHD FADLI FADZIR

FINAL PROJECT REPORT

**Submitted to the Electrical & Electronics Engineering Programme
in Partial Fulfillment of the Requirements
for the Degree
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)**

**Universiti Teknologi Petronas
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan**

© Copyright 2005

by

Mohd Fadli Fadzir, 2005

CERTIFICATION OF APPROVAL

IMAGE COMPRESSION USING WAVELETS

by

Mohd Fadli Bin Fadzir

A project dissertation submitted to the
Electrical & Electronics Engineering Programme
Universiti Teknologi PETRONAS
in partial fulfilment of the requirement for the
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Approved:



Ms. Nasreen Badruddin

Project Supervisor

Nasreen Badruddin

Lecturer,

Electrical & Electronic Engineering

Academic Block No 22

Universiti Teknologi PETRONAS

Bandar Seri Iskandar

31750 Tronoh, Perak Darul Ridzuan, MALAYSIA

UNIVERSITI TEKNOLOGI PETRONAS

TRONOH, PERAK

December 2005

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.



MOHD FADLI BIN FADZIR

ABSTRACT

Image compression enables images for easier data storage and data transmission. One of newest technique used in compressing image is wavelet transform. Wavelet widely used in application such as medical imaging, internet imaging, scanning and printing, mobile and digital cameras. Wavelets are new filter that can keep the information in both time domain and frequency domain. The special about wavelet filter is that the window can be varied by changing the frequency. The objective of the project is to create a simulation model to investigate image compression using wavelets. The investigation will makes comparative study by applying different types of wavelet techniques on different types of images. The MATLAB software is used in doing simulation. As necessary background to do the project, basic concept of image processing, wavelet theory, image compression, and information theory are learned and discussed. The simulation will use several types of wavelets families including Haar, Daubachies, Symlet, Coiflet and Biorthogonal Spline wavelets. The papers will analyze and examine the effect of difference wavelet families, filter order, filter length, decomposition level and image content and quantizer type in compressing image. After doing numerous comparisons of wavelet effects on all test images, the results of the simulation shows that Daubachies wavelets family is having the most outstanding performance compared to other wavelet families. Hence, Daubachies is the best filter to use in doing wavelet image compression.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my project supervisor Miss Nasreen Badruddin for her technical and moral support which made this final year project report possible. Her continuous feedback and support throughout the semester helping the student by giving encouragement, suggestions and ideas throughout the final year project are invaluable. I appreciate the time she spent on reviewing my writing and improving my technical writing skills.

Lastly, I also wish to express my sincere gratitude to my parents, my brother and my sisters for their unconditional love, support and trust for providing me the best of educational opportunities to study in University Technology PETRONAS.

TABLE OF CONTENTS

LIST OF TABLES	viii
LIST OF FIGURES.....	ix
LIST OF ABBREVIATIONS	1
CHAPTER 1 INTRODUCTION	1
1.1 Background of study	1
1.2 Problem Statement	1
1.3 Significant of the project	2
1.4 Objective and scope of study.....	2
1.5 Feasibility of the Project within the Scope and Time frame	3
CHAPTER 2 LITERATURE REVIEW AND THEORIES	5
2.1 Image Processing Overview	5
2.2 JPEG2000 Architecture.....	6
2.3 Multi-resolution analysis (MRA) properties	7
2.4 One dimensional Discreet Wavelet Transforms (orthogonal).....	8
2.5 One dimensional Discreet Wavelet Transforms (Bi-orthogonal).....	8
2.6 Perfect reconstruction condition for wavelet Filter Bank.....	9
2.7 Two dimensional Discreet Wavelet Transforms (Orthogonal)	9
2.8 Performance Measurement.....	12
2.9 Image Quantization	13
2.10 Information Theory	14
2.11 Image Entropy	15
2.12 Run-length encoding	16
2.13 Huffman Code	16
2.14 Arithmetic Coding.....	17
CHAPTER 3 METHODOLOGY / PROJECT WORK	18
3.1 Methodology	18
3.2 MATLAB Implementation Methodology	21
3.2.1 Two Dimensional Discrete Wavelets Transform.....	21
CHAPTER 4 RESULT AND DISCUSSION	24
4.1 JPEG2000 Process.....	24
4.2 Wavelet Basis Properties.....	25

4.3 Filter Bank Structure	26
4.4 Selecting Decomposition Level.....	27
4.5 Selecting Quantization Method.....	29
4.5.1 Uniform Quantization Calculations	29
4.5.2 Quantization Comparison	31
4.6 Image Entropy	32
4.7 Data Comparison.....	33
4.7.1 Comparing Between Same Families.....	33
4.7.2 Orthogonal wavelet filter	33
4.7.3 Bi-orthogonal-Spline Wavelets.....	35
4.7.4 Comparing Between Different Wavelets Families	36
4.8 MATLAB RESULTS.....	39
CHAPTER 5 CONCLUSION.....	42
REFERENCES.....	44
APPENDIX	45
Appendix A 2-D DWT MATLAB CODE	46
Appendix B MSE LEVEL 1	68
Appendix C MSE LEVEL 2.....	69
Appendix D MRA DEFINITION.....	70
Appendix E ENTROPY CODING.....	72
Huffman Code	72
Arithmetic Coding.....	75
Appendix F DATA AND EXCEL GRAPH.....	79
Appendix G TEST IMAGES USED	86

LIST OF TABLES



Table 1 : FYP Semester 1 Gantt Chart.....	3
Table 2 : FYP Semester 2 Gantt Chart.....	4
Table 3 : 8X4 image with corresponding Gray level	15
Table 4 : Frequency and probability of gray level 8X4 image.....	15
Table 5 : Wavelet Properties Used[1]	26
Table 6 : Gray Level Frequency table.....	32
Table 7 : Baboon MSE level 1	36
Table 8 : Baboon MSE level 2	37
Table 9 : F16 MSE level 1	37
Table 10 : F16 MSE level 2	37
Table 11 : Suitable filter with its image to use.....	38
Table 12 : Orthogonal and Biorthogonal Basis Properties[3,4]	71
Table 13 : Huffman Code and entropy	74
Table 14 : Probabilities table for arithmetic coding.....	75

LIST OF FIGURES

Figure 1 : Image arranged in matrix.....	5
Figure 2 : JPEG2000 Block diagram.....	6
Figure 3 : MRA space and subspace[3,4]	7
Figure 4 : 1-D Orthogonal DWT filter bank	8
Figure 5 : 1-D Bi-orthogonal DWT filter bank	8
Figure 6 : 2-D Orthogonal Analysis Filter level 1	10
Figure 7 : 2-D Orthogonal Synthesis filter level 1	10
Figure 8 : Decomposition Level 1	11
Figure 9 : Calculation of Huffman code for 3-bit image [6].....	16
Figure 10 : Representation of interval scale up of Arithmetic coding	17
Figure 11 : Methodology.....	18
Figure 12 : 2-D DWT filter banks (analysis and reconstruction).....	21
Figure 13 : Block Diagram of JPEG 2000	24
Figure 14 : Wavelet Families [1].....	25
Figure 15 : 4X4 image.....	27
Figure 16 : Row operation.....	27
Figure 17 : Column operation	28
Figure 18 : Illustration of 6-bit uniform quantization with 64 levels.....	30
Figure 19 : Original Barbara image and its Histogram	31
Figure 20 : Quantized Barbara Image using 6-bit quantizer and Histogram	31
Figure 21 : Histogram of Original Barbara Image	32
Figure 22 : Level 1 decomposition.....	39
Figure 23 : Approximation Coefficients Level 1	39
Figure 24 : Histogram of original picture.....	40
Figure 25 : Histogram of the compressed image.....	41
Figure 26 : Histogram of the reconstructed image.....	41
Figure 27 : Calculation of Huffman code for 3-bit image [6].....	73
Figure 28 : The second symbol scales it again into the range [0.2, 0.26)	76
Figure 29 : Representation of interval scale up of Arithmetic coding	77
Figure 30 : Lena Coiflet MSE level 1	79
Figure 31 : F16 Coiflet MSE level 1	79

Figure 32 : Lena Coiflet MSE level 2	79
Figure 33 : F16 Coiflet MSE level 2	80
Figure 34 : Lena Symlet MSE level 1	80
Figure 35 : Boat Symlet MSE Level 1	80
Figure 36 : Lena Symlet MSE level 2	81
Figure 37 : Boat Symlet MSE Level 2	81
Figure 38 : Lena Daubachies MSE Level 1	81
Figure 39 : Goldhill Daubachies MSE Level 1	82
Figure 40 : Lena Daubachies MSE Level 2	82
Figure 41 : Goldhill Daubachies MSE Level 2	82
Figure 42 : Bior Spline MSE level 1	83
Figure 43 : Peppers Bior Spline MSE level 1	83
Figure 44 : Bior Spline MSE level 2	83
Figure 45 : Peppers Bior Spline MSE level 2	84
Figure 46 : Baboon MSE level 1	84
Figure 47 : F16 MSE level 1	84
Figure 48 : Baboon MSE Level 2	85
Figure 49 : F16 MSE Level 2	85
Figure 50 : House Image	86
Figure 51 : Lena Image	86
Figure 52 : Boat Image	87
Figure 53 : F16 Image	87
Figure 54 : Girl Image	88
Figure 55 : Peppers Image	88
Figure 56 : Barbara Image	89
Figure 57 : Goldhill Image	89
Figure 58 : Baboon Image	90

LIST OF ABBREVIATIONS

$f(x)$	A signal or function
$f(x) \in V_j$	Scaling function or basis function
$f(x-k)$	k determine the position of $f(x)$ along x axis
$\{\psi_{j,k}(t)\}$	Expansion Set of wavelet function
$\langle f(x), \psi(x) \rangle$	Inner product of $f(x)$ and $\psi(x)$
V_j	Subspace level j
$span\{ \}$	Function space referred to closed span of $\{ \}$
$V_j = span\{\phi_{j,k}\}$	V_j is the span of $\phi_{j,k}$ over k for any j
$L^2 \mathbb{R}$	Set of measurable, square integrable functions
h_k	Scaling function vector
g_k	Wavelet function vector
$V_0 = \phi(x)$	Scaling function subspace
$W_0 = \psi(x)$	Wavelet function subspace
$\psi(x)$	Wavelet function
\oplus	Denotes the union of space/set
$V_1 = V_0 \oplus W_0$	Decomposition of subspace V_1
$h_k(n) = h(n)$	Scaling function coefficient
$g_k(n) = g(n)$	Wavelet function coefficient
	Down sample / decimation
	Up sample / interpolation
$H(z)$	Scaling function coefficient in z-transform

$G(z)$	Wavelet function coefficient in z-transform
LL	Approximation coefficients
HL	Vertical detail coefficients
LH	Diagonal detail coefficients
HH	Horizontal detail coefficient
DWT	Discrete Wavelets Transform
2-D	2 dimensional
SNR	Signal to Noise Ratio
MSE	Mean Square Error
PSNR	Peak Signal To Noise Ratio
CR	Compression Ratio
ADC	Analog To Digital Converter
M	Quantization Level
b	Step size
f_{\max}	Maximum value of gray level
f_{\min}	Minimum value of gray level
$\hat{E}[x, y]$	Reconstruction Level
$H(x)$	Information Content or Entropy
$p(x)$	Probability of gray level
R	Redundancy
I_{\max}	Quantization Bit
t_n	Sequence of threshold
P	Quantization Level
R_n	Dynamic range of uniform quantization
r_n	Reproduction level

CHAPTER 1

INTRODUCTION

1.1 Background of study

Digital imaging has significant contributions to industrial application and scientific applications. Uncompressed images require big amount of memory and storage. Compression allows easier storage and transmission of digital images. Many techniques can be used to compress digital images. Before wavelet transform, popular technique used in image compression are Discreet Cosine Transform (DCT) and Short Time Fourier Transform (STFT). Wavelet transform is new technique replacing Short Time Fourier Transform and Discreet Cosine Transform because it is proven better in image compression. Wavelets widely used in applications such as medical imaging, internet imaging, scanning and printing, mobile and digital cameras.

1.2 Problem Statement

The primary concern of this project is to conduct research in wavelets and simulate the performance of wavelet image compressions on different test images. Knowledge of basic image processing and wavelet theory is studied first before starting programming. The research will emphasize on the wavelet theory such as filter bank, discreet wavelet transform, image representation and analysis of the performance of image compression using wavelet based on different type of families.

1.3 Significant of the project

JPEG2000 is the standard used to compress all still images. The wavelet transform technique is found to be the most comprehensive technique for the image compression. Engineers and researchers especially in signal processing area are doing numerous researches to find the most suitable wavelet family to use. Hopefully, the project will help the engineers in the field of image compression improving their research.

1.4 Objective and scope of study

The aim of the project:

- Create a simulation model to investigate wavelet compression.
- Apply different wavelet techniques on different images.
- Measure the performance using different wavelet technique.
- Collect and analyze the results on these different techniques on different image.

The scope of study will consist of 5 main components:

- Image processing - Understand basic theory of image processing-image sampling, image representation and color image
- Wavelet theory - Understand the multi resolution analysis, wavelet discrete transform in one dimensional and two dimensional and mathematical concept.
- Image Compression – study basic of data compression such as information theory, entropy, Run length coding, Huffman coding, Arithmetic coding and scalar quantization.
- Simulation on software - Software that will be used in programming is MATLAB version 7. Student need to study the image processing toolbox and wavelet toolbox.
- No hardware implementation - The image compression algorithm will be simulated on software only. No hardware implementation using FPGA or VLSI.
- Test images that will be used through out the project are the standard images publicly available images used in image compression research.

1.5 Feasibility of the Project within the Scope and Time frame

The Final Year projects are divided into two main parts. The First part consists of literature review and the familiarization with MATLAB software. The second part is the programming modeling and simulation. For this semester, the programming will cover some of JPEG-2000 architecture such as wavelet transform, quantization and entropy. Data collection is the last step in evaluating the performance of wavelet family. Literature will cover 5 areas such as basic image processing, wavelets theory, wavelet image compression and image properties.

Table 1 : FYP Semester 1 Gantt Chart

Activity/Task	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1. Introduction	X														
2. Image Processing															
Image Representation		X													
Sampling			X												
Quantization				X											
3. Wavelets															
MRA					X										
1-D DWT						X									
2-D DWT							X								
Wavelet Filter Bank								X	X						
4. Programming										X	X	X			
5. Simulate Programming													X	X	X

Table 2 : FYP Semester 2 Gantt Chart

Activity/Task	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1. Wavelet Properties															
Orthogonal	X														
Bi-orthogonal	X	X													
Vanishing Moment		X	X												
Filter Length			X	X											
Compact Support				X											
Support Width				X											
2. Image Compression															
Information Theory				X	X										
Run Length Coding						X									
Huffman Coding							X								
Arithmetic coding								X							
Scalar Quantization									X						
3. Image Properties															
Texture									X						
4. Programming															
Level 1	X	X	X	X	X										
Level 2						X	X	X	X	X					
Quantization											X	X	X		
5. Data Comparison															
MSE													X	X	
PSNR													X	X	
6. Compile															X

CHAPTER 2

LITERATURE REVIEW AND THEORIES

2.1 Image Processing Overview

Digital images are created from continuous image. The process of converting the continuous image to digital form is done by the analog to digital converter (ADC). ADC process consists of sampling and quantization. Sampling is digitizing the pixel of image in coordinate form. Quantizing is the process of rounding off values to nearest integer. The result of sampling and quantizing are shown in figure 1. Figure 1 shows that coordinate of pixel values are arranged in matrix. The image is taken from Digital Image Processing 2nd edition text book [8].

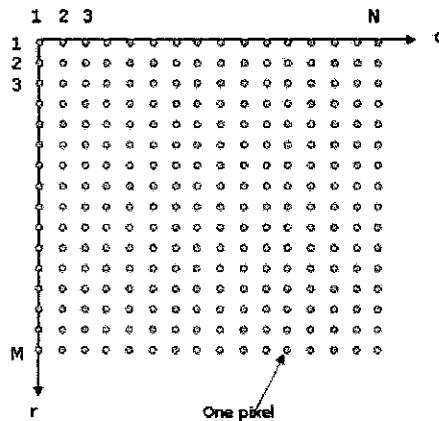


Figure 1 : Image arranged in matrix.

The digital image is a representation of 2-D array of continuous image in discrete sample. The values of the array are the function of intensity or brightness. The intensity ranges from 0 to 255 for 8-bit quantization. 0 is black and 255 is white [8].

Digital image processing is simple. As image is a matrix of pixel values, the image processing is simply done by matrix manipulation and matrix calculation. MATLAB is software that uses matrix approach to solve image manipulation. In MATLAB, 4 types of images are supported including index images, intensity images, binary images and RGB images. MATLAB basically stores digital images in 2- D matrix form. Each element in the matrix consists of intensity or grey levels values. Grey levels values ranges from 0 to 255 for 8-bit quantization. Quantization levels are commonly referred to as grey levels. 8 bit quantization has 256 possible grey levels .2 bit quantization is a binary image [1].

2.2 JPEG2000 Architecture

JPEG-2000 is a standard used to compress digital images using wavelet transformation. JPEG-2000 process consist of 2 main part called encoder and decoder as shown in figure 2. The block diagram is taken from wavelet-based image compression paper [3]. For encoder part, the process starts with the source images are partitioned into rectangular non-overlapping region. Then the tiled component is decomposed into several sub bands. The decomposition process is done using the forward discrete wavelet transform (DWT). After that, all decomposition coefficients are quantized to reduce the number of grey levels to represent the images. Lastly, the quantized coefficients are reduced in bits to represent the image. The reduction process is called entropy coding. For the decoder part, the process is simply the reverse of the encoder process.

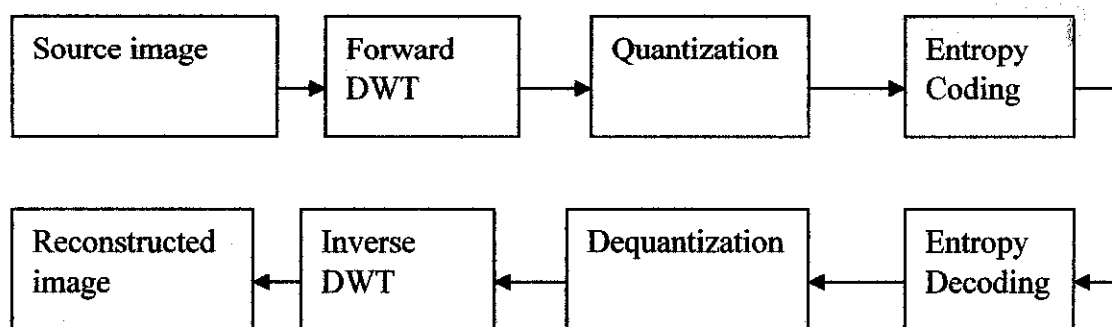


Figure 2 : JPEG2000 Block diagram

2.3 Multi-resolution analysis (MRA) properties

Wavelet decomposition can be expressed as a function of the wavelet basis $\{\psi_{j,k}(t)\}$.

The wavelet basis $\{\psi_{j,k}(t)\}$ decompose the function space of L^2 into a direct sum of orthogonal subspaces W_j and V_j [3, 4]. For a resolution level j , a wavelet transform performs a decomposition of the space V_j into a direct sum of orthogonal subspaces [3, 4]. More detail on mathematical definition of MRA can be found on Appendix D.

For orthogonal the decomposition

$$V_{j+1} = V_j \oplus W_j \quad V_1 = V_0 \oplus W_0 \quad V_2 = V_0 \oplus W_0 \oplus W_1 \quad L^2 = V_0 \oplus W_0 \oplus W_1$$

For bi-orthogonal the decomposition

$$V_{j+1} = W_j \oplus V_j \quad V_j = \text{span}\{\phi_{j,k}\} \quad W_j = \text{span}\{\psi_{j,k}\} \quad \tilde{V}_j = \text{span}\{\tilde{\phi}_{j,k}\}$$

$$\tilde{W}_j = \text{span}\{\tilde{\psi}_{j,k}\}$$

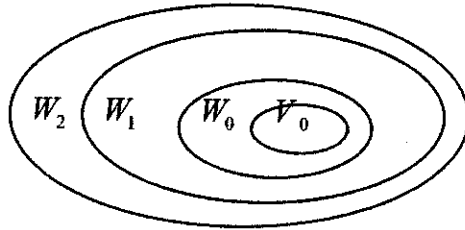


Figure 3 : MRA space and subspace[3,4]

Requirement for Multi Resolution Analysis:[4]

- a. Nestedness $V_2 \subset V_1 \subset V_0 \subset V_{-1} \subset V_{-2}$
- b) Density $\bigcap_{j \in \mathbb{Z}} V_j = \{0\}$, and $\overline{\bigcup_{j \in \mathbb{Z}} V_j} = L^2(\mathbb{R})$
- c) Scaling $f(x) \in V_j \Leftrightarrow f(x-k) \in V_0$ for all integer k
- d) Orthonormal Basis $\exists \phi \in V_0$ such that $\{\phi(x-k)\}_{k \in \mathbb{Z}}$ constitute basis for V_0
- e) $\{\phi(x-k)\}_{k \in \mathbb{Z}}$ is a riesz basis

2.4 One dimensional Discrete Wavelet Transforms (orthogonal)

Discrete wavelet transform are constructed by cascading two channel filter bank with analysis filter and synthesis filter as shown in figure 4. The filter bank image is taken from MATLAB wavelet toolbox. The decomposition is performed by analysis filter and reconstruction is performed by synthesis filter. Analysis filters simply the time reversal of synthesis filters for orthogonal [1].

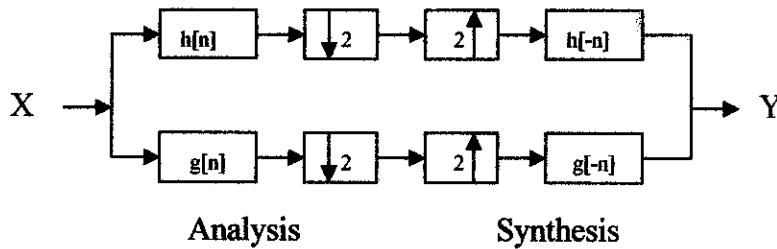


Figure 4 : 1-D Orthogonal DWT filter bank

2.5 One dimensional Discrete Wavelet Transforms (Bi-orthogonal)

Discrete wavelet transform are constructed by cascading two channel filter bank with analysis filter and synthesis filter as shown in figure 5. The image of bi-orthogonal filter bank is taken from MATLAB wavelet toolbox. The decomposition is performed by analysis filter and reconstruction is performed by synthesis filter. Analysis filters simply inverse of synthesis filters for bi-orthogonal [1].

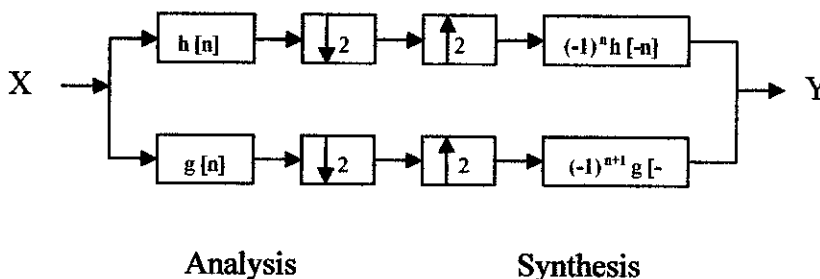


Figure 5 : 1-D Bi-orthogonal DWT filter bank

2.6 Perfect reconstruction condition for wavelet Filter Bank

i) Condition for Biorthogonal

$$\text{No distortion } h(-z)h(z) + -g(-z)g(z) = 2z^{-d}$$

$$\text{No aliasing } h(-z)h(z) + -g(-z)g(-z) = 0$$

ii) Condition for Orthogonal

$$\text{No distortion } h(z^{-1})h(z) + g(z^{-1})g(z) = 2z^{-D}$$

$$\text{No aliasing } h(z)h(z^{-1}) + g(z^{-1})g(z) = 0$$

2.7 Two dimensional Discrete Wavelet Transforms (Orthogonal)

Wavelet basis decompose image using low pass filter (scaling wavelet) and high pass filter (dilation wavelet). Wavelet basis decompose $L^2(\mathbb{R}^2)$ space into several sub space. $L^2(\mathbb{R}^2)$ space is created from the tensor product of scaling function φ and wavelet function ψ . Decomposition process is the product of scaling function φ and wavelet function ψ . For orthogonal decomposition, the image will be decomposed into 4 sub bands called Approximation Coefficients (LL), Vertical coefficients (HL), Horizontal Coefficients (LH) and Diagonal Coefficients (HH). Similarly, orthogonal decomposition can be written as:

$$\begin{aligned} \text{LH} &= \psi^1(x, y) = \varphi(x)\psi(y), & \text{HL} &= \psi^2(x, y) = \psi(x)\varphi(y) \text{ and} \\ \text{HH} &= \psi^3(x, y) = \psi(x)\psi(y) \end{aligned} \quad (1)$$

For bi-orthogonal decomposition, the difference is the scaling function and wavelet function has dual pair namely $\varphi, \psi, \tilde{\varphi}$ and $\tilde{\psi}$. Similarly the dual wavelets for a bi-orthogonal decomposition can be written as:

$$\begin{aligned} \text{LH} &= \tilde{\psi}^1(x, y) = \tilde{\varphi}(x)\tilde{\psi}(y), & \text{HL} &= \tilde{\psi}^2(x, y) = \tilde{\psi}(x)\tilde{\varphi}(y) \text{ and} \\ \text{HH} &= \tilde{\psi}^3(x, y) = \tilde{\psi}(x)\tilde{\psi}(y) \end{aligned} \quad (2)$$

For bi-orthogonal 2-D DWT , the image is decomposed using scaling and wavelet function same way as 1-D DWT. 2-D DWT filter bank can simply constructed by cascading to channel filter bank consist of analysis filter, synthesis filter, down sampling and up sampling operation.

The structure of the filter bank is simply cascading two series of one dimensional filter bank to each other and down sampling by factor of two or up sampling after each filter as figure 6. For decomposition, analysis filter will be followed by down sampling by a factor of 2. For reconstruction, the synthesis filter is followed by up sampling by a factor of two.

After transformation using 2 channel filter bank, the image then decomposed into 4 sub bank as before. LL, HL, LH and HH will results as the result of the images passed through filter bank .The image of 2-D DWT filter bank is taken from MATLAB Wavelets Toolbox [1].

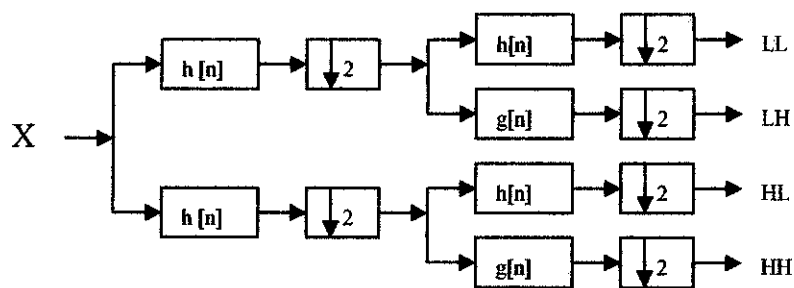


Figure 6 : 2-D Orthogonal Analysis Filter level 1

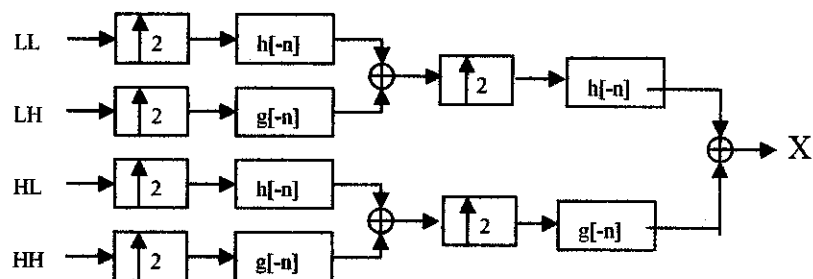


Figure 7 : 2-D Orthogonal Synthesis filter level 1

After the decomposition, the image $f(x, y)$ is decomposed into 4 sub bands called approximation coefficient a_j , vertical detail d_j^x , horizontal detail d_j^y , and a diagonal detail d_j^{xy} . The approximation and detail can be written as below:

$$\begin{aligned} a_j(x, y) &= \langle f(x, y), \phi_j(x)\phi_j(y) \rangle & d_j^x(x, y) &= \langle f(x, y), \psi_j(x)\phi_j(y) \rangle \\ d_j^y(x, y) &= \langle f(x, y), \phi_j(x)\psi_j(y) \rangle & d_j^{xy}(x, y) &= \langle f(x, y), \psi_j(x)\psi_j(y) \rangle \end{aligned} \quad (3)$$

The image $f(x, y)$ becomes half of original size after the down sampling operation as shown in figure 8. The image is taken from Wavelet and Signal Processing Course Lecture Note from University of Tehran. One is low-pass filtered in the x-direction and high-pass filtered in the y direction $\psi^1(x, y) = \phi(x)\psi(y)$. One is low-pass filtered in the y direction and high-pass filtered in the x direction $\psi^2(x, y) = \psi(x)\phi(y)$. Finally, one is high-pass filtered in both x and y directions $\psi^3(x, y) = \psi(x)\psi(y)$. Figure 8 shows us 4 orientations including approximation (LL), horizontal (LH), vertical (HL) and diagonal (HH) [5].

LL $a_j(x, y)$	HL $d_j^x(x, y)$
LH $d_j^y(x, y)$	HH $d_j^{xy}(x, y)$

Figure 8 : Decomposition Level 1

2.8 Performance Measurement

1. Signal-to-Noise-Ratio (SNR) measures are estimates of the quality of a reconstructed image compared with the original image.

$$SNR = 10 \log_{10} \left(\frac{\sigma_x^2}{\sigma_q^2} \right) db \quad (4)$$

2. Mean square error (MSE) is the cumulative squared error between original and compressed image. Lower value of MSE means lesser error, and these values give higher peak signal-to-noise ratio (PSNR).

$$\sigma_q^2 = \sum (f_{original}(i, j) - f_{compressed}(i, j))^2 \quad (5)$$

3. Peak signal-to-noise ratio (PSNR) is another qualitative measure based on the root-mean-square-error of the reconstructed image. [8]

$$PSNR = 10 \log_{10} \left[\frac{255 \times MN}{\sum_{xy} |g(x, y) - f(x, y)|} \right] \quad (6)$$

$$PSNR = 10 \log_{10} \left(\frac{255^2}{\sigma_q^2} \right) db \quad (7)$$

2.9 Image Quantization

Quantization is the process of converting some values of discrete set of grey level to an integer between 0 and 255 for 8-bit images. 0 is black and 255 is white. The quantization process is done by the analog to digital converter. 8-bit quantization has 255 levels. A quantizer with m bits defines $M = 2^m$ levels [6].

Most quantizers use 8-bit having $m=8$. So the image will have 256 different gray levels. 6-bits is often sufficient for human visual system. 50 levels of quantization are satisfactory for human to view images because human eye-brain is limited to brightness resolution of an image.

The step size b can also be called resolution. It can be defined as the difference in brightness between adjacent gray levels. In uniform quantization, the step size of quantized levels is fixed.

$$b = \frac{f_{\max} - f_{\min}}{2^m - 1} \quad (8)$$

Where f_{\max} and f_{\min} are the minimum and maximum of pixel values and m is the number of quantizer bits. [6]

Image can be quantized using equation below

$$f_q[x, y] = Q \left\{ \frac{f[x, y] - f_{\min}}{b} \right\} \quad (9)$$

where $Q\{ \}$ represents the nonlinear truncation or rounding operation. The form Q determines the location of the decision level [6].

Quantized images are then reconstructed by assigning all the pixel values to its corresponding decision levels. Factor $b/2$ are added for reconstruction levels.

$$\hat{E}[x, y] = \left(f_q[x, y] \cdot \frac{E_{\max} - E_{\min}}{2^m - 1} \right) + E_{\min} + \frac{b}{2} \quad (10)$$

Usually the reconstructed values are not equal to original value, $\hat{E}[x, y] \neq E[x, y]$ due to the quantization. The error results are called quantization error [6].

2.10 Information Theory

Data compression is the process of reducing the amount of data to represent the information. The compression simply removes the redundancy information. The information content of data is determined by its Entropy [6, 7].

$$H(x) = -\log_2 p(x) \quad (11)$$

where x is the symbol and $p(x)$ is its probability. Entropy will decrease when probability increase. Higher the probability, lower the entropy thus the shorter codeword should be assigned to the symbol. The entropy $H(x)$ gives the number of bits required to code the symbol in an average.

An image is the discrete set of probabilities. Probabilities can be found from histogram counts of grey levels of an image. Shannon has derive entropy equation which can be used to determine the quantity of information [6,7].

$$H(f) = -\sum_{x=1}^k p(x) \cdot \log_2 p(x) \quad (12)$$

where k is the number of symbols in the alphabet. Entropy is measured in bits per pixel/symbols.

The redundancy R can be defined as the ratio of non-redundant data to the total data:

$$R = 1 - \frac{\text{nonredundant data}}{\text{total data}} \quad \text{or} \quad R(f) = 1 - \frac{H(f)}{I_{\max}} \quad (13)$$

where $H(f)$ is the entropy of image f and I_{\max} is the number of quantization bits.

2.11 Image Entropy

To estimate the information content or entropy of an image, the source model based on the relative frequency of occurrences of grey levels is constructed. Image entropy $H(f)$ can be calculated using Shannon entropy equation [8]:

$$H(f) = -\sum_{x=1}^k p(x) \cdot \log_2 p(x) \quad \text{where } p(x) = \frac{h(k)}{MN}$$

where $h(k)$ is the frequency of grey level and M and N is image size. Suppose we have 8X4 image as below and we need to find the entropy.

Table 3 : 8X4 image with corresponding Gray level

21	21	21	95	169	243	243	243
21	21	21	95	169	243	243	243
21	21	21	95	169	243	243	243
21	21	21	95	169	243	243	243

Table 4 : Frequency and probability of gray level 8X4 image

Gray level	count	probability
21	12	12/32
95	4	4/32
169	4	4/32
243	12	12/32

The entropy of the image then calculated using equation below

$$H[f] = -\sum_{i=0}^k p(x) \log_2 p(x)$$

$$H[f] = -\frac{12}{32} \log_2 \frac{12}{32} - \frac{4}{32} \log_2 \frac{4}{32} - \frac{4}{32} \log_2 \frac{4}{32} - \frac{12}{32} \log_2 \frac{12}{32} = 1.81 \text{ bits / pixel}$$

2.12 Run-length encoding

Run length encoding inserts a special token each time a chain of more than two equal input tokens are found. This special input advises the decoder to insert the following token n times into his output stream. For example, consider the sequence of 3-bit data:

7 7 7 2 6 6 6 6 6 2 2 5 5 5 5 5 5 5 5

The encoded RLE image would be: 7 3 2 1 6 5 2 2 5 9

where the first digit in the encoded image is the gray value of the first pixel, the second digit is the number of occurrences. The sequence is reduced from 20 digits to 10 digits.

2.13 Huffman Code

A procedure for obtaining a code from the histogram was specified by David A. Huffman in 1951. The most common gray levels are assigned to a code word that requires few bits to transmit, while levels which occur infrequently are encoded with many bits. To derive the Huffman code, arrange the occupied levels in descending order of probability. One bit is used to distinguish the least likely pair of levels. The binary digit 1 is assigned to the more probable level and 0 to the less probable. The schematic of the entire sequence is shown in the figure 9. Image is taken from Basic Imaging Science Lecture Note [6].

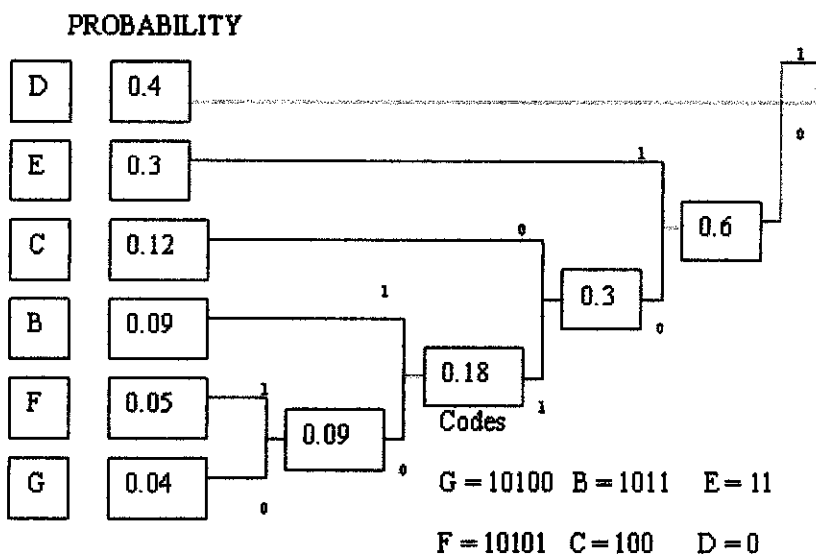


Figure 9 : Calculation of Huffman code for 3-bit image [6].

2.14 Arithmetic Coding

In arithmetic coding, a message is represented by an interval of real numbers between 0 and 1. As the message becomes longer, the interval needed to represent it becomes smaller and the number of bits needed to specify that interval grows. Before anything is transmitted, the range for the message is the entire interval $[0, 1)$, denoting the half-open interval $0 \leq x < 1$. As each symbol is processed, the range is narrowed to that portion of it allocated to the symbol. Proceeding like this, the decoder can identify the whole message [9].

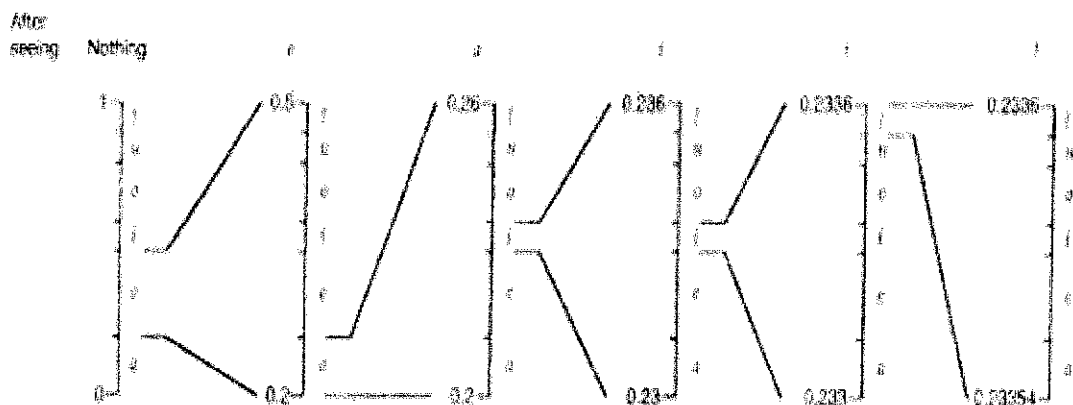


Figure 10 : Representation of interval scale up of Arithmetic coding

More detail explanations about entropy coding on run length encoding, Huffman code and arithmetic coding can be found in appendix E.

CHAPTER 3

METHODOLOGY / PROJECT WORK

3.1 Methodology

Figure 11 below shows the methodology involved in finishing the whole project.

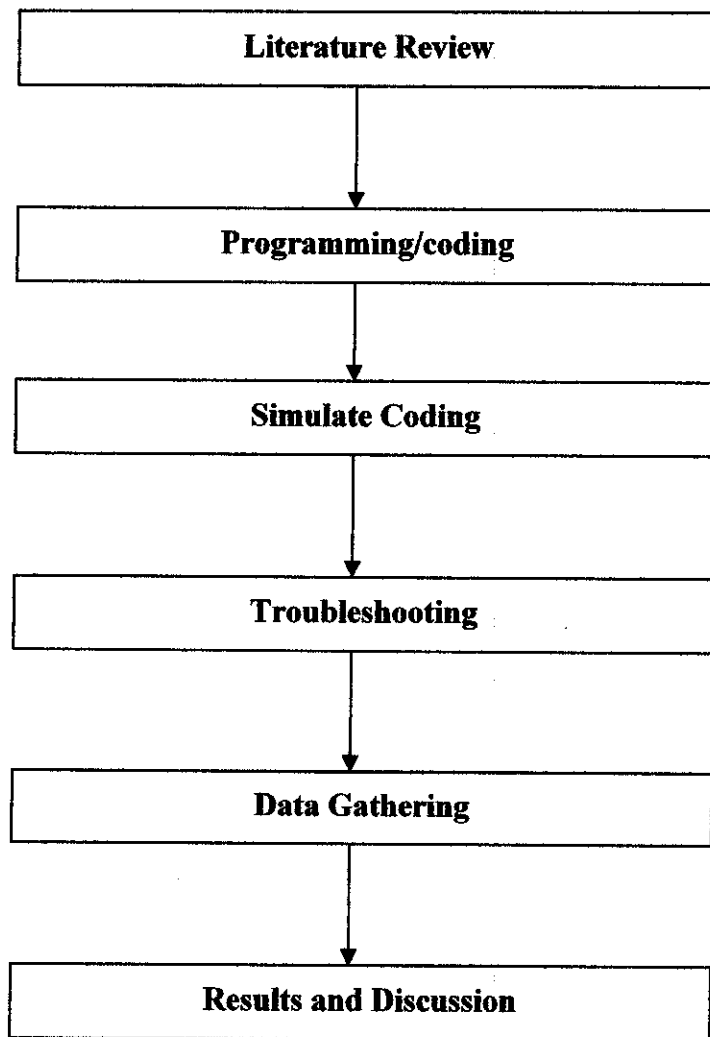


Figure 11 : Methodology

1. Literature Review

- **Image Processing** -Study digital image processing concept such as image sampling, image quantization and image representation in MATLAB. Learn to manipulate image in MATLAB.
- **Wavelet** - conduct research and learn the wavelet theory in image compression, multi resolution analysis, properties of wavelet basis, mathematical background on wavelets, wavelet filter bank, 1-D DWT, 2-D DWT and orthogonal/bi-orthogonal wavelets families.
- **Image Compression** – study basic of data compression such as information theory, entropy, Run length coding, Huffman coding, Arithmetic coding and scalar quantization.

2. Programming/coding

- **Choose wavelet basis** - Select either orthogonal basis or biorthogonal basis to use in the coding.
- **Implement 2-D DWT filter** using MATLAB.
- **Perform the decomposition of image** using Analysis filter
- **Quantizing the wavelet coefficient** using Scalar Quantization
- **Perform the reconstruction** using the Synthesis Filter.
- **Learn MATLAB** by referring Image Processing Toolbox and Wavelet Toolbox user guide.

3. Simulate Coding

- **Compile the program**
- **Check error**

4. Troubleshooting

- **Eliminate error** in the program
- **Recompile the program** or make new program

5. Data Gathering

- Collect Mean Square error , Signal To Noise Ratio , Peak Signal To Noise Ratio and Compression Ratio

6. Data Analysis

- Compare with each type of wavelet family using MSE, SNR, PSNR and CR. The calculation is done by MATLAB.

7. Results and Discussion

- Compare the performance of the image compression using wavelet on different type of wavelet family
- Manipulate data and use Microsoft Excel to obtain graph

3.2 MATLAB Implementation Methodology

3.2.1 Two Dimensional Discrete Wavelets Transform

Actually, the process to use in implementing the wavelet compression is by using to 2-D DWT filter bank as shown in figure below. Figure 12 shows how the filter bank is constructed.

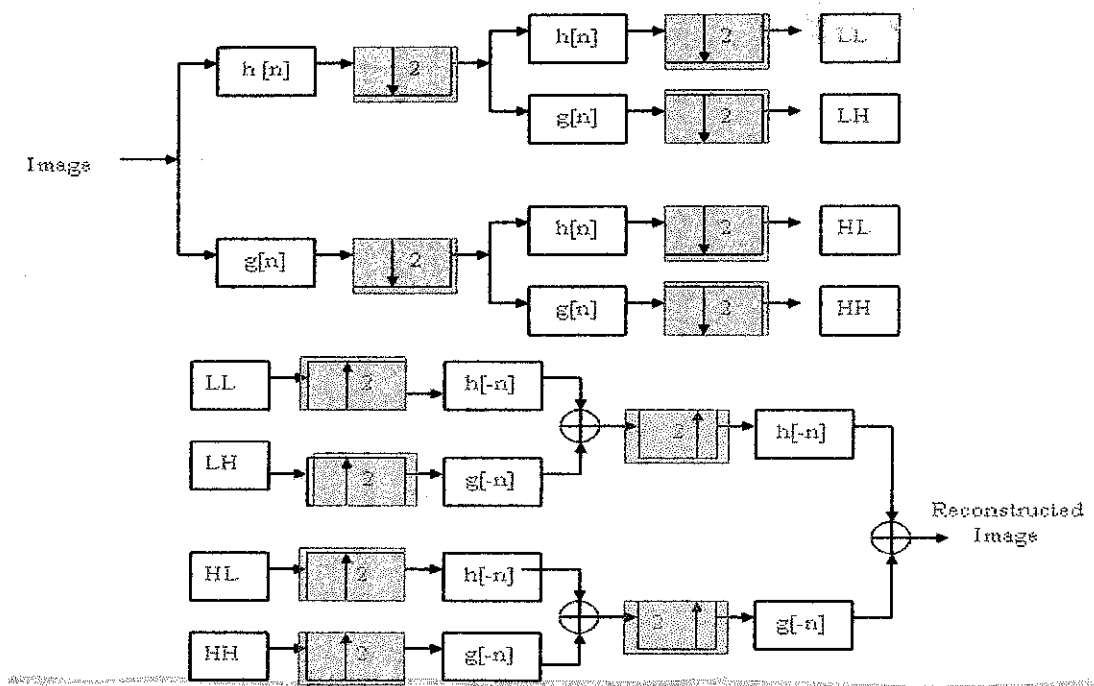


Figure 12 : 2-D DWT filter banks (analysis and reconstruction)

The step involve in implementing 2-D DWT are as follows:

1. Loading the RGB Image (Lena_std.tif)
2. Convert RGB image to gray scale
3. Finding the filter coefficient for the filter bank using wfilters function
4. Perform decomposition on the y - row
 - by convolving low pass filter with image
 - by convolving high pass filter with image

5. Down sample along the y direction

- downsample the value of convolution of low pass
- downsample the value of convolution of high pass

6. After down sample , perform convolution again

- by convolving second low pass filter with down sampled value of low pass
- by convolving with second high pass filter with down sampled value of low pass
- by convolving second low pass filter with down sampled value of high pass
- by convolving with second high pass filter with down sampled value of high pass

7. Down sample for second time along x – column

- downsample the value of convolution of second low pass - low pass
- downsample the value of convolution of second high pass - low pass
- downsample the value of convolution of second low pass - high pass
- downsample the value of convolution of second high pass - high pass

8. Arrange the decomposition coefficient in matrix.

9. Quantizing the wavelet decomposition matrix

- parameters (step size 64/uniform quantizer)

10. Now the reconstruction Part

11. Finding the reconstruction filter coefficients

12. Upsample the coefficient of second time along x – column

- upsample the downsample value of convolution of second low pass - low pass
- upsample the downsample value of convolution of second high pass - low pass
- upsample the downsample value of convolution of second low pass - high pass
- upsample the downsample value of convolution of second high pass - high pass

13. After upsample , perform convolution

- by convolving second low pass filter with up sampled value of low pass
- by convolving with second high pass filter with up sampled value of low pass
- by convolving second low pass filter with up sampled value of high pass
- by convolving with second high pass filter with up sampled value of high pass

14. Sum the results of convolution of the

- up sampled value of low pass + up sampled value of low pass
- up sampled value of high pass + up sampled value of high pass

15. Up sample the value of summation along y - column

- summation of the up sampled value of low pass + up sampled value of low pass
- summation of the up sampled value of high pass + up sampled value of high pass

16. After up sample, perform convolution

- by convolving low pass filter with value of upsample of summation
- by convolving high pass filter with value of upsample of summation

17. Sum the results of convolution of the

- up sampled value of low pass
- up sampled value of high pass

18. The result of last summation is reconstructed image.

CHAPTER 4

RESULT AND DISCUSSION

4.1 JPEG2000 Process

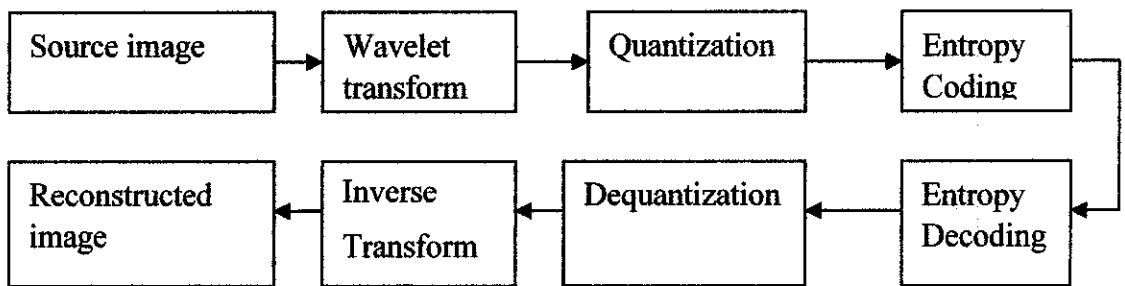


Figure 13 : Block Diagram of JPEG 2000

Before starting MATLAB programming, the properties of each type of wavelet family need to be studied. The properties required to study as it gives impact an image quality. These are two types of basis are orthogonal and biorthogonal. The simulation was done by according to the JPEG-2000 architecture. First the image needed to convert into grey scale first. Then the grey image is applied to forward discrete wavelet transform. If decomposition level one was chosen, the image will be divided into 4 sub band of frequency. The decomposition coefficients were then quantized to difference decision level according to how many bits the quantizer used.

4.2 Wavelet Basis Properties

The first step in wavelet transform is to choose appropriate wavelet basis. Currently, Haar wavelet family, Daubachies wavelet family, Coiflet wavelet family, Symlet wavelet family and Biorthogonal Spline wavelets family are chosen in implementing wavelet transform. The choice of which type of basis to use is depends on several properties of basis. The wavelet basis and scaling wavelet basis properties are studied first. The primary concern is to choose the wavelet basis.

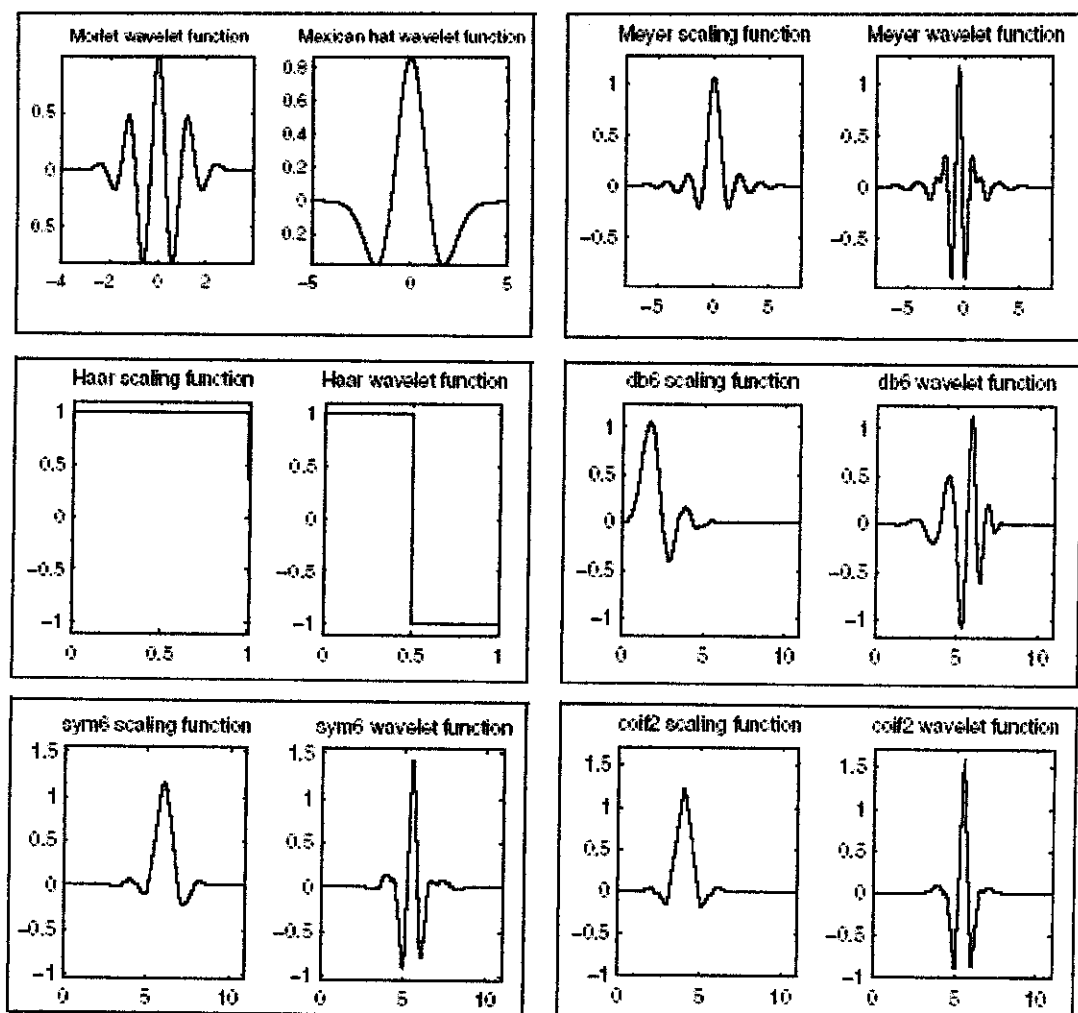


Figure 14 : Wavelet Families [1]

Below are the summary of common properties of wavelets and scaling.

1. Orthogonal - refers to MRA basis properties. Inverse transform get back the coefficient $g_i = \langle f(x), \psi(x) \rangle = \int f(x)\psi(x)dx$ [4]
2. Vanishing moment - refers to number of region which the integral of the region is zero. The basis is said to has p vanishing moment if $\int f^k \psi(x)dx = 0$ for $0 \leq k \leq p$. Haar has one vanishing moment and Db4 has 4 vanishing moment [4].
3. Filter Length – refers to the length of basis length[4]
4. Compact support – refers to the region which the function is nonzero. The compact support of Haar has width of 1 [4].

Type	Orthogonal /Biorthogonal	Vanishing Moment	Filter Length	Compact Support	Support Width
Haar	Yes, both	1	2	Yes	1
Coif2	Yes, both	4	12	Yes	11
Coif4	Yes, both	8	24	Yes	23
Db5	Yes, both	5	10	Yes	9
Db10	Yes, both	10	20	Yes	19
Sym4	Yes, both	4	8	Yes	7
Sym7	Yes, both	7	14	Yes	13

Table 5 : Wavelet Properties Used[1]

4.3 Filter Bank Structure

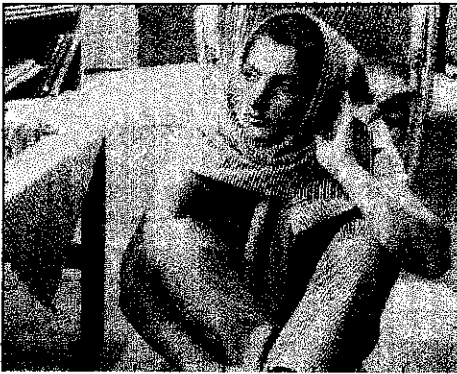
The second step is choosing filter bank structures. There are many types of filter structure such as separable, multi level and packet structures. Wavelet packet filter bank structure is chosen for implementation in MATLAB.

Choosing filter bank structure is required for implementing discreet wavelet transform. The filter bank consists of analysis filter and synthesis filter. For the orthogonal reconstruction, the synthesis filter is simply time reversal of the analysis side.

4.4 Selecting Decomposition Level

Selecting decomposition level is also important before starting the programming. Decomposition and reconstruction until level 4 are chosen. At the moment only level one is completed.

Actually two dimensional wavelet transform is formed by applying one-dimensional transforms in both the horizontal and vertical directions on the image. The image is decomposed into a set of low pass samples and a set of high-pass samples. Assume we have 4X4 matrix image. The pixel values are arranged as below.



$$\begin{bmatrix} A & B & C & D \\ E & F & G & H \\ I & J & K & L \\ M & N & O & P \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{bmatrix}$$

Figure 15 : 4X4 image

We begin by applying one dimensional discrete wavelet transform on both row and column of the image. First of all, the image will be divided into two parts as figure 16. These two parts perform on row of the image. The left side is low pass filter and the right side is high pass filter. On the low pass side we apply sum operation on the values between adjacent pixels. While for the high pass side we apply the difference between adjacent pixel values.



$$\frac{1}{\sqrt{2}} \begin{bmatrix} A+B & C+D & A-B & C-D \\ E+F & G+H & E-F & G-H \\ I+J & K+L & I-J & K-L \\ M+N & O+P & M-N & O-P \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1+2 & 3+4 & 1-2 & 3-4 \\ 5+6 & 7+8 & 5-6 & 7-8 \\ 9+10 & 11+12 & 9-10 & 11-12 \\ 13+14 & 15+16 & 13-14 & 15-16 \end{bmatrix}$$

Figure 16 : Row operation

After row operation finish, the image then divides once again. This time we apply column operation on the image. The above part of the column of image is low pass side and below part of column is high pass side. We then do sum operation on low pass part and then apply the difference on high pass side again.

$$\frac{1}{\sqrt{2}} \begin{bmatrix} (A+B)+(E+F) & (C+D)+(E+H) & (A-B)+(E-F) & (C-D)+(G-H) \\ (I+J)+(M+N) & (K+L)+(O+P) & (I-J)+(M-N) & (K-L)+(O-P) \\ (A+B)-(E+F) & (C+D)-(G+H) & (A-B)-(E-F) & (C-D)-(G+H) \\ (I+J)-(M+N) & (K+L)-(O+P) & (I-J)-(M-N) & (K-L)-(O-P) \end{bmatrix} =$$

$$\frac{1}{\sqrt{2}} \begin{bmatrix} (1+2)+(5+6) & (3+4)+(7+8) & (1-2)+(5-6) & (3-4)+(7-8) \\ (9+10)+(13+14) & (11+12)+(15+16) & (9-10)+(13-14) & (15-16)+(11-12) \\ (1+2)-(5+6) & (3+4)-(7+8) & (1-2)-(5-6) & (3-4)-(7+8) \\ (9+10)-(13+14) & (11+12)-(15+16) & (9-10)-(13-14) & (11-12)-(15-16) \end{bmatrix} =$$

As results of the sum and difference, the image then transformed into 4 sub band frequency as results of filtering. Analysis filter used to decompose image into its approximation coefficient (LL), vertical detail (LH), horizontal detail (HL) and diagonal detail (HH).



$$\frac{1}{2} \begin{bmatrix} 14 & 22 & -2 & -2 \\ 46 & 54 & -2 & -2 \\ -8 & -8 & 0 & 0 \\ -8 & -8 & 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 7 & 11 & -1 & -1 \\ 23 & 27 & -1 & -1 \\ -4 & -4 & 0 & 0 \\ -4 & -4 & 0 & 0 \end{bmatrix}$$

Figure 17 : Column operation

4.5 Selecting Quantization Method

The resulting coefficients of the forward wavelet transform are quantized using scalar quantization. Real JPEG-2000 architecture use vector quantization. In JPEG2000, vector quantization is used due to its performance better compared to scalar quantizer. Vector quantization quantizes value by array while scalar quantizer quantizes individual value.

In implementing the scalar quantization in MATLAB, Uniform Quantization is chosen in quantizing the wavelet transform coefficient. In the programming, scalar quantization is choosing in quantizing the wavelet coefficients. Uniform quantization is selected because it is simpler to implement in MATLAB programming.

The uniform quantizer is chosen due to it is easier to understand. Before doing the quantization, the concept of image quantization such as step size, reproduction level, and dynamic range are studied and revisited.

4.5.1 Uniform Quantization Calculations

Let $t_n \in \{0,1,\dots,256\}$ denotes the sequence of threshold ($n=0,1,\dots,P-1$)

Consider the P half open discrete intervals $R_n = [t_n, t_{n+1})$ ($t_0 = 0, t_p = 256$)

Let $r_n \in R_n$ be the production level of interval R_n

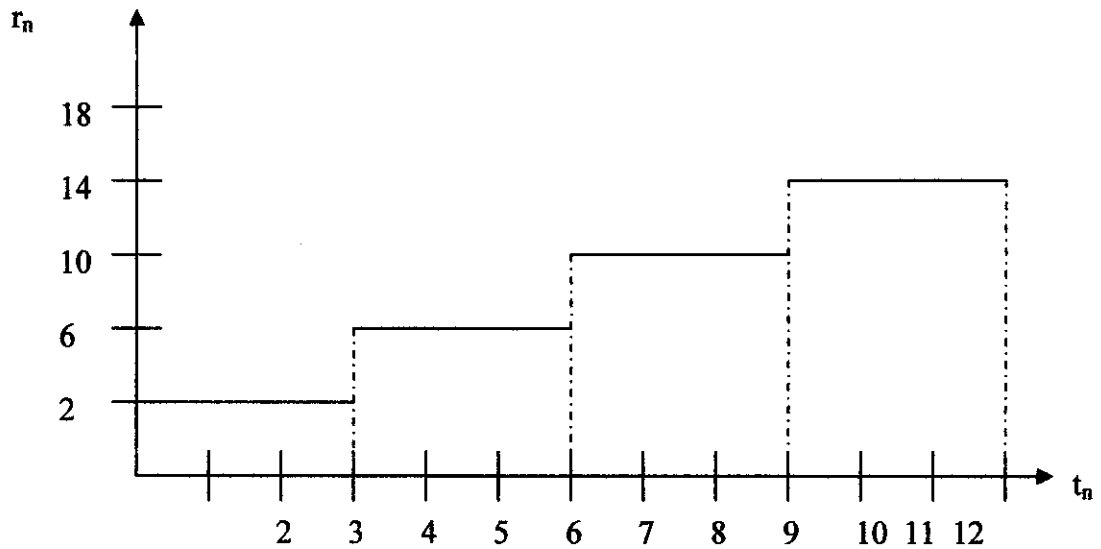


Figure 18 : Illustration of 6-bit uniform quantization with 64 levels

Reproduction level can be calculated using equation $r_n = n\Delta + \Delta/2$

Step size of uniform quantizer is fixed. Delta can be calculated using $\Delta = 256/P$

For example, we want to calculate reproduction level for $n = 0,1$ and 2. By using equation $r_n = n\Delta + \Delta/2$, we get $r_n = 2,6$ and 14

$$r_0 = 0(4) + 4/2 = 2$$

$$r_1 = 1(4) + 4/2 = 6$$

$$r_2 = 2(4) + 4/2 = 10 \quad r_3 = 3(4) + 4/2 = 14$$

So, uniform quantization is simply $Q(A) = \text{delta} * \text{floor}(A/\text{delta}) + \text{delta}/2$

For example, to quantize pixel value 7, $Q(7) = 4 * \text{floor}(7/4) + 4/2 = 10$

4.5.2 Quantization Comparison

Below are the comparison between un-quantized image and quantized image using 6-bit quantizer. The image pixel of quantized image is reduced after the quantization. By human vision the image is the same. To see clearer the pixel reduce we can see the histogram of the image. From figure 19 and figure 20 it is clear that histogram of original image is having more grey levels compared to histogram of the quantized image. This is because many of the pixel value are truncated or rounded to 64 levels corresponding reproduction levels.

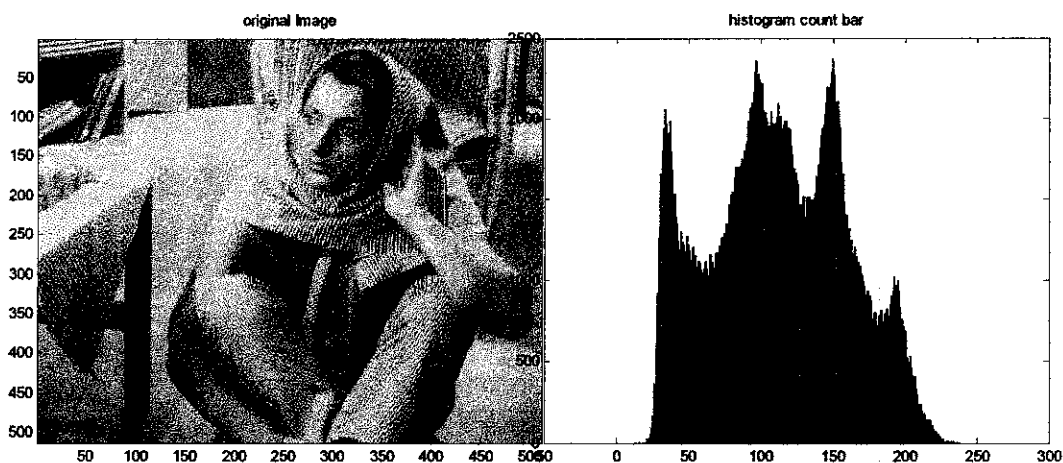


Figure 19 : Original Barbara image and its Histogram

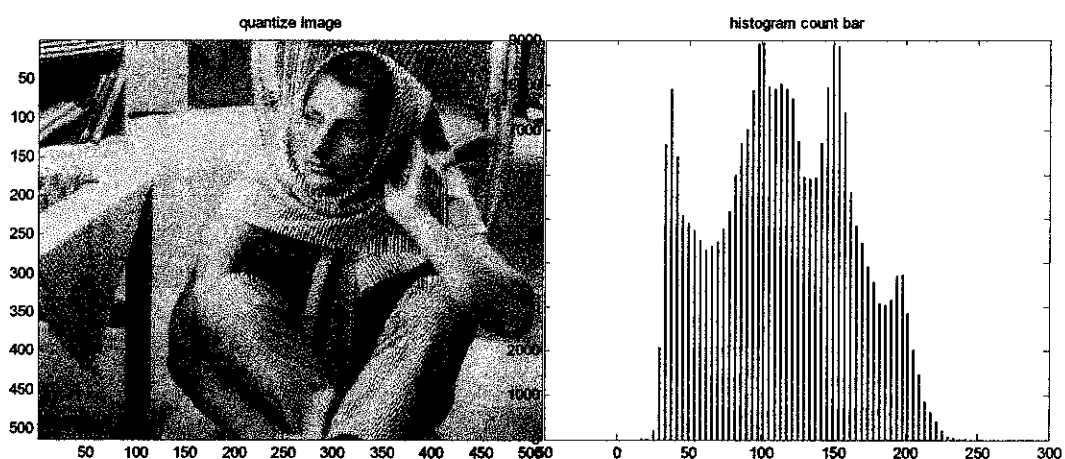


Figure 20 : Quantized Barbara Image using 6-bit quantizer and Histogram

4.6 Image Entropy

Entropy H can be calculated using equation $H = -\sum_{x=1}^k p(x) \cdot \log_2 p(x)$.

To find the probabilities, first we need to get the quantity of grey level from image histogram. The probabilities can be obtained by dividing the frequency of grey level to the total array of 512x512 images as shown in Table 5. For example, we select 5 pixel values and list down its frequency/count in the table 6. The count value is obtained from the histogram of Barbara image as shown in figure 21.

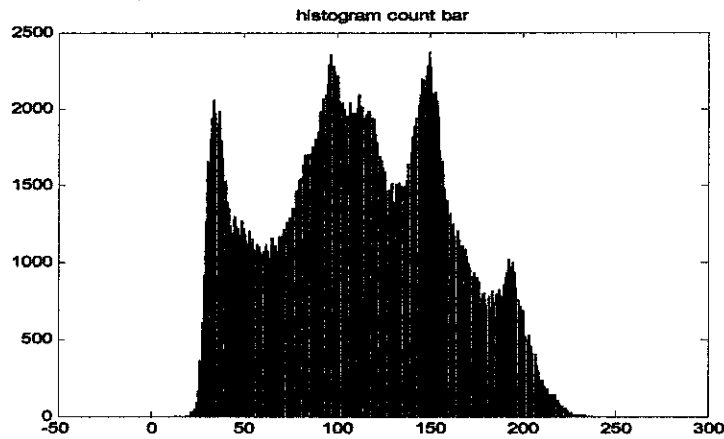


Figure 21 : Histogram of Original Barbara Image

Table 6 : Gray Level Frequency table

counts	Grey Level	Probability(x)	Log ₂ p(x)
2057	34	2057/(512*512)= 0.0078	-7.0023
1968	35	1968/(512*512)= 0.0075	-7.0589
1907	36	1907/(512*512)= 0.0073	-7.0979
1981	37	1981/(512*512)= 0.0076	-7.0398
1797	38	1797/(512*512)= 0.0069	-7.1792

Lastly, substitute the values into the formula and sum all the entropy.

$$\begin{aligned}
 H &= 0.0078(-7.0023) + 0.0075(-7.0589) + 0.0073(-7.0979) + 0.0076(-7.0398) \\
 &\quad + 0.0069(-7.1792) \\
 &= -(-0.26241332)
 \end{aligned}$$

4.7 Data Comparison

4.7.1 Comparing Between Same Families

From the MATLAB experiment, the MSE and PSNR are obtained after the wavelet transformation and quantization. A comparison between the results on 9 different images shows that the results obtained are quite similar and close to each others. In comparing wavelet, orthogonal wavelet families and bi-orthogonal wavelet families will be evaluated.

4.7.2 Orthogonal wavelet filter

i) Coiflet Wavelets

At level 1 decomposition, for Coiflet families, the effect on the images performance is the same for all images. MSE of Coiflet families tend to increase when filter length increase and higher order used. The higher order Coiflet filter used the greater error. Coif4 has the least MSE compared to Coif5 and Coif30. For level 1, MSE of Coiflet wavelet can be described by graph in Appendix F by referring to figure 30 and figure 31.

At level 2 decomposition, for Coiflet families, the effect goes different way. MSE of Coiflet families tend to decrease when filter length increase and higher order used. The same trend goes to F16 at 2nd level decomposition. Now, the MSE of Coiflet families tend to decrease when filter length increase and higher order used. For level 2, MSE of Coiflet wavelet can be described by graph in Appendix F by referring to figure 32 and figure 33.

ii) Symlet Wavelets

At level 1 decomposition, for the Lena image, Symlet filter that having the lowest order or least filter length resulting in the least MSE values. MSE of Symlet families tend to increase when higher filter length used. Symlet20 having the greatest MSE compared other Symlet filter of its families. For level 1, MSE of Symlet wavelet can be described by graph in Appendix F by referring to figure 34 and figure 35.

At level 2 decomposition, for the Lena image, different things happen. Symlet filter that having the greatest order or longest filter length resulting in the least MSE values. MSE of Symlet families tend to decrease when higher filter length used. Symlet20 having the least MSE compared other Symlet filter of its families. For level 2, MSE of Symlet wavelet can be described by graph in Appendix F by referring to figure 36 and figure 37.

For Boat image, Symlet filter that having the highest order or greatest filter length resulting in the least MSE values. MSE of Symlet families tend to decrease when higher filter length used. Symlet20 having the least MSE compared other Symlet filter of its families.

iii) Daubachies Wavelets

For Lena image, Daubachies results are quite the same as other orthogonal wavelet families. The MSE of the Daubachies families tend to increase when higher filter length used and higher order used. Daub 20 having the largest MSE compared to Daub4.

For Goldhill, at first level decomposition, different things occur. The MSE of the Daubachies families tend to decrease when higher filter length used and higher order used. Daub 20 having the least MSE compared to Daub4. For first level decomposition, MSE of Daubachies wavelet can be described by graph in Appendix F by referring to figure 38 and figure 39.

For Goldhill, at second level decomposition, same thing goes. The MSE of Daubachies decreases with increase in filter length. For second level decomposition, MSE of Daubachies wavelet can be described by graph in Appendix F by referring to figure 40 and figure 41.

4.7.3 Bi-orthogonal-Spline Wavelets

Bi-orthogonal-Spline wavelet exhibits the same trend for all test images used. As filter length and filter order increase the MSE values tend to reduced. Biorthogonal Spline 6.8 having lower MSE compared to Biorthogonal Spline 3.5. For level 1, MSE of Bi-orthogonal-Spline wavelet can be described by graph in Appendix F by referring to figure 42 and figure 43. For level 2, MSE of Bi-orthogonal-Spline wavelet can be described by graph in Appendix F by referring to figure 44 and figure 45.

4.7.4 Comparing Between Different Wavelets Families

As the values of MSE and PSNR results obtain from all 9 testing images used are close to each other and quite similar, the comparison will choose the image that having the minimum/ least MSE values and maximum/greatest MSE values. The image that has smallest MSE is Baboon. While the image that having the largest MSE is F16.

For Orthogonal families, the MSE results shows most of the wavelet filter families will exhibits higher MSE when the higher order filter are used. However, some of the images show that when the higher filter order or the higher filter length used the MSE will decrease.

For Bi-orthogonal Families, the trend is fixed. When the order of the filter increase or the length of filter increase, the MSE will reduce and the PSNR will increase.

For clear depiction of the data comparison, first consider the Baboon image that having the lowest MSE and PSNR among all 9 images. From the table below, we can conclude that Daubachies is the best filter to choose as it gives the lowest MSE.

For clear understanding, graph of MSE for all filters can be found in Appendix F. For level 1 MSE, we can refer to figure 46 and figure 47. For level 2 MSE, we can refer to figure 48 and figure 49.

Table 7 : Baboon MSE level 1

Wavelet Families	MSE Range	PSNR Range
Coiflet	364.3440 – 365.0913	22.5157 – 22.5068
Daubachies	360.0708 – 364.7039	22.5669 – 22.5114
Symlet	363.0828 – 364.6884	22.5307 – 22.5116
Bi orthogonal Spline	368.9388 – 486.0851	22.4613– 21.2637

Entropy_Original = 7.3579

Entropy_compressed = 1.0666

Table 8 : Baboon MSE level 2

Wavelet Families	MSE Range	PSNR Range
Coiflet	515.3079 - 515.6138	21.0101 - 21.0076
Daubachies	511.7137 - 516.1005	21.0405 - 21.0035
Symlet	514.2737 - 515.2665	21.0189 - 21.0105
Bi orthogonal Spline	503.3479 - 588.0076	21.1121 - 20.4370

Secondly, consider the F16 image that having the largest MSE among all the 9 testing images. From the table below, we can conclude that Daubachies is the best filter to choose as it give the highest PSNR and the least MSE.

Table 9 : F16 MSE level 1

Wavelet Families	MSE Range	PSNR Range
Coiflet	426.0049 – 430.1945	21.7942 – 21.8367
Daubachies	412.7994 – 423.6672	21.8378 – 21.9734
Symlet	426.0049 – 426.5423	21.8312 – 21.8599
Bi orthogonal Spline	446.8476 – 598.6834	20.3588 – 21.6879

Table 10 : F16 MSE level 2

Wavelet Families	MSE Range	PSNR Range
Coiflet	588.3153 - 592.6305	20.4347 - 20.4030
Daubachies	579.4243 - 587.7575	20.5008 - 20.4388
Symlet	587.5570 - 607.9090	20.4403 - 20.2924
Bi orthogonal Spline	586.7354 - 698.1819	20.4464 - 19.6911

Entropy Original = 6.7025

Entropy compressed = 0.8281

From both results obtain, we can make conclusion that for High Frequency Image, the suitable filter to use is the filter having the longest filter length as it gives smallest MSE. For Low Frequency Image, the suitable filter to use is the filter having the shortest length or lower order as it will give smallest MSE.

After comparison, the images now classified according to which suitable filter to use. The corresponding filters are match to images as summarized as table 10 below.

Table 11 : Suitable filter with its image to use

Level 1		Level 2	
Higher Order	Lower Order	Higher Order	Lower Order
Daubachies		Daubachies	
Lena	Baboon Boat Peppers Girl House F16 Barbara	Lena	Baboon Boat Peppers Girl House F16 Barbara
Coiflet		Coiflet	
	Baboon Boat Peppers Girl House F16 Barbara	Lena Baboon Boat Peppers Girl House F16 Barbara	
Symlet		Symlet	
Baboon Girls House F16	Lena Goldhill Boat Peppers Barbara	Lena Baboon Boat Peppers Girl House F16	Barbara

4.8 MATLAB RESULTS

In figure 22, original image is decomposed into 4 sub band comprised of LL, HL, LH and HH. The size of the image is reduced to half. Original size is 512x512 and the size of the image after decomposition is 256x256. Figure 23 shows the approximation coefficient (LL).

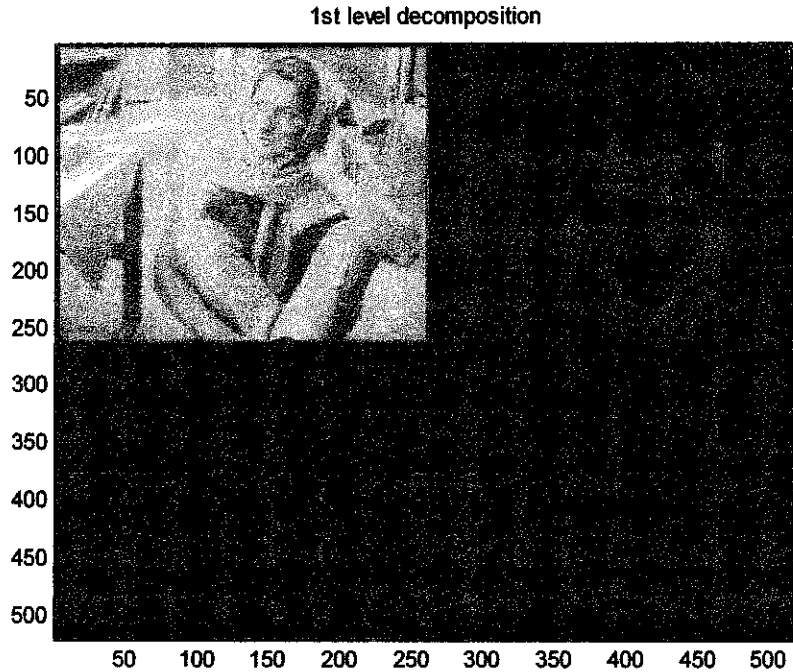


Figure 22 : Level 1 decomposition

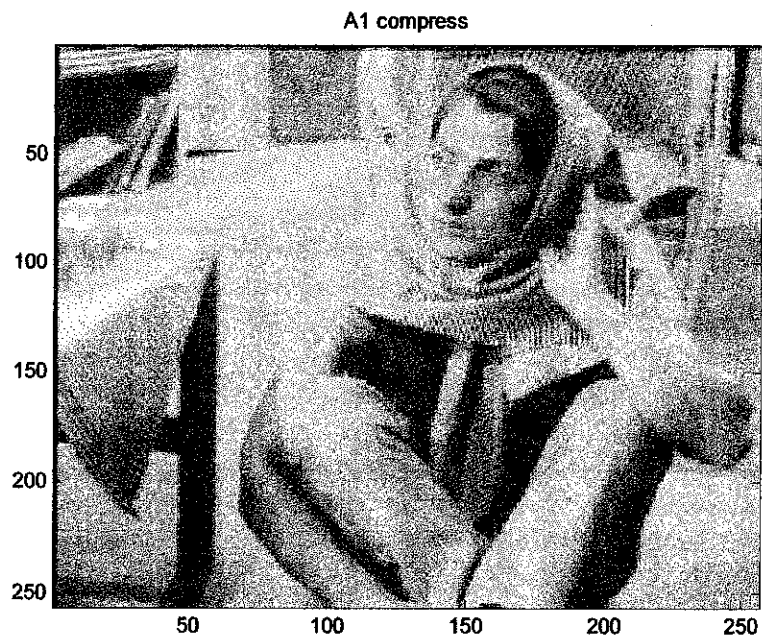


Figure 23 : Approximation Coefficients Level 1

From the histogram we can know the frequency of the grey levels. From both picture of figure 24 and figure 25, we can notice that the histogram shows that the number of grey level of original are more than the compressed one. The things happen due to many of the pixel values are round off to 8 different decisions levels.

In figure 26, although the number of grey level of the reconstruct image is more than the compress image, the value is still less than the original image. The quantization is irreversible process because the images after compression are reduced in the quantity of information.

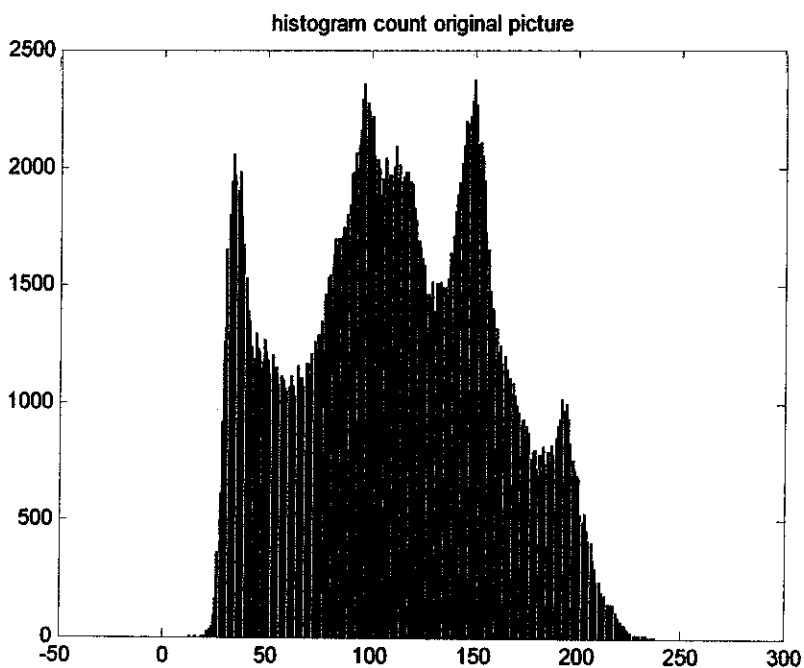


Figure 24 : Histogram of original picture

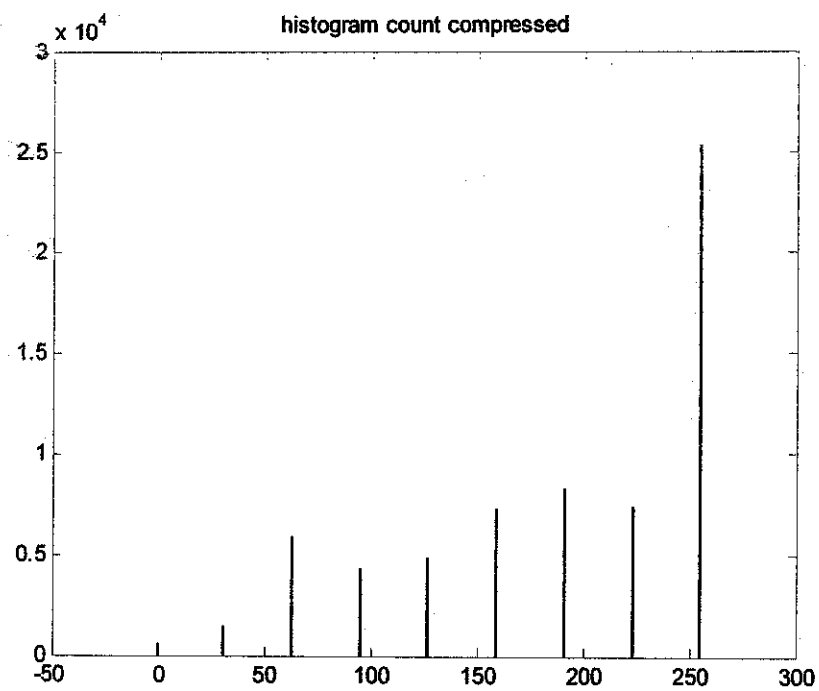


Figure 25 : Histogram of the compressed image

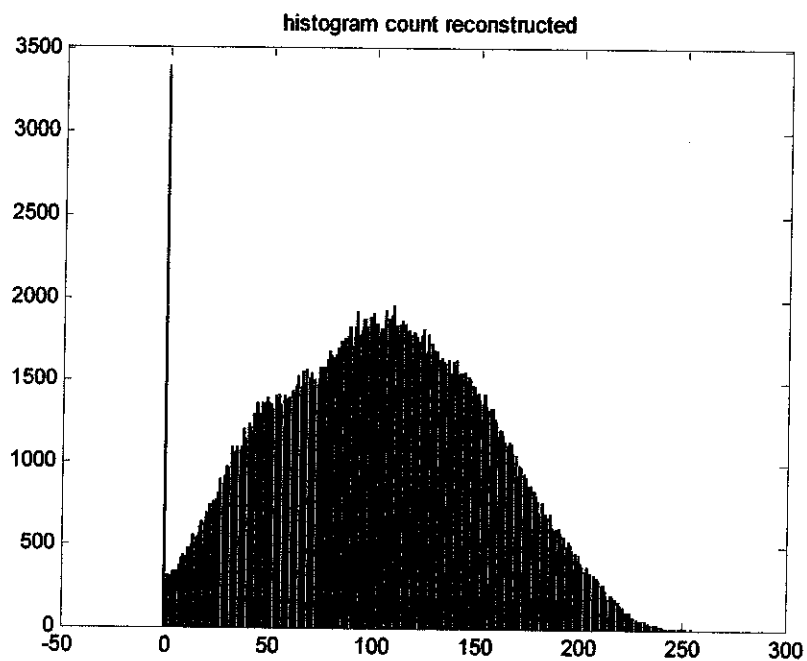


Figure 26 : Histogram of the reconstructed image

CHAPTER 5

CONCLUSION

As a conclusion overall objective is fully achieved. The main objective to make simulation on wavelet transform had been achieved. Currently, the simulation had covers 2/3 parts of JPEG-2000 architecture.

The two parts that had been complete was wavelet transform and quantization of the wavelet coefficients. The third part Entropy coding was not being implemented due to difficulties and time constraint. First part is the wavelet transform. The simulation for wavelet transform is fully completed. The simulation can now support various wavelet filter families such as Haar wavelet families, Coiflet wavelet families, Symlet wavelets families, Daubachies families and Biorthogonal Spline wavelets families.

Second part of programming is quantization parts. The simulation only implements uniform quantization. Currently, the programming on quantizing the wavelet decomposition coefficient was fully complete. Third part is entropy coding. No simulation was done. Only literature review was done. Many areas were covered including Run length Coding, Huffman Coding and Arithmetic Coding.

After the simulation complete, performance measurement can be made. At the moment, comparison has successfully made due to all the reconstructed value having some MSE due to quantization. As wavelet transform is using perfect reconstruction filters the MSE can only obtain after the quantization. 4-bit quantizer were chosen in quantizing the image.

In this paper, the performance comparisons made are evaluated according to quality of image after the wavelet compression. The performance is compared by analyzing MSE and PSNR results of different images after the compression.

The purpose of the project is to analyze and examine the effect of difference wavelet functions, filter order, filter length, decomposition level and image content and quantizer type in compressing image. After some comparison, we have seen that it is not easy to select the most suitable filter to use in compressing the image.

From the results obtain, we can conclude that quality of the compressed image depend on many factor. First conclusion is effect of decomposition level on quality after compression. As we moved on larger decomposition, quality will decrease. Values of MSE increase when decomposition level increases.

If we analyze effect of filter order or filter length, we see that higher filter order or higher filter length does not imply better image quality. From the experiment, most of the images indicate that lower order or shorter filter length will give higher performance.

We also can make conclusion that effect of wavelet filter depend on the properties of the images itself. For High Frequency Image, the suitable filter to use is the filter having the longest filter length as it gives smallest MSE. For Low Frequency Image, the suitable filter to use is the filter having the shortest length or lower order as it will give smallest MSE.

For all test images, the result shows that the most outstanding wavelets filter family is Daubachies wavelet family. Daubachies wavelet family is proven having the lowest MSE compared to other wavelet families used in this paper. Hence, the best filter to use in wavelet image compression is Daubachies because it has the best performance among all filter that have been used in the simulation. Hopefully in the future, the simulation coding can support vector quantization, arithmetic coding and Huffman coding.

REFERENCES

1. **Wavelet Toolbox User's Guide Version 2**
COPYRIGHT 1997 – 2002 by the MathWorks, Inc.
2. **Image Processing Toolbox User's Guide Version 2**
COPYRIGHT 1997 – 2002 by the MathWorks, Inc.
3. **Wavelet-based image Compression 18th November 2002**
Gang Jun Tu, Department of Computer Science
University of Aarhus, Denmark
4. **Wavelet Tour of Signal Processing Presentation**
Stephane Mallat, 2nd Edition, Academic Press
http://cas.ensmp.fr/%7Echaplais/Wavetour_presentation/Wavetour_presentation_US.html
5. **Wavelet And Signal Processing Lecture Note**
University of Tehran
6. **Basic Principal Of imaging Science II Lecture Note**
Roger Easton
7. **Image Compression Lecture Note**
Alexander Kolesnikov
University Of Joensuu
8. **Digital Image Processing 2nd edition**
Rafael C.Gonzalez
9. **Arithmetic Coding For Data Compression**
Ian H. Witten, Radford M. Neal, And John G. Cleary

APPENDIX

APPENDIX A

2-D DWT MATLAB CODE

Please refer to v6level2.m below

```
close all;
clc
clear all;
%-----%
image = menu('Choose an
image', 'Lena', 'Baboon', 'goldhill', 'tiffany', ...
    'peppers', 'boat', 'F16', 'Barbara', 'house');

if image == 1
    I_color = imread('Lena_std.tif');
    X = rgb2gray(I_color);
elseif image == 2
    X = imread('Baboon.tif');
elseif image == 3
    X = imread('goldhill.tif');
elseif image == 4
    I_color = imread('tiffany.tiff');
    X = rgb2gray(I_color);
elseif image == 5
    I_color = imread('peppers.tiff');
    X = rgb2gray(I_color);
elseif image == 6
    X = imread('boat.tif');
elseif image == 7
    I_color = imread('F16.tiff');
    X = rgb2gray(I_color);
elseif image == 8
    X = imread('Barbara.tif');
elseif image == 9
    I_color = imread('house.tiff');
    X = rgb2gray(I_color);
end
```

```

% finding filter coefficients

wname = menu('Choose a filter', 'Haar', 'Coif4', 'Coif5', 'db5', 'db7', ...
'sym4', 'sym8', 'Bior35', 'Bior68', 'CDF97', 'coif30', 'haar2', 'daub4', ...
'daub10', 'daub20', 'sym16', 'sym20', 'bior97')

if wname == 0
    disp('error')

elseif wname == 1
    [lo_dec, hi_dec, lo_rec, hi_rec] = wfilters('haar');

    %lo_dec = lo_dec
    %hi_dec = hi_dec
    %lo_rec = fliplr(lo_rec)
    %hi_rec = fliplr(hi_rec)

elseif wname == 2
    wname = 'coif4';
    [lo_dec, hi_dec, lo_rec, hi_rec] = wfilters(wname);

elseif wname == 3
    wname = 'coif5';
    [lo_dec, hi_dec, lo_rec, hi_rec] = wfilters(wname);

elseif wname == 4
    wname = 'db5';
    [lo_dec, hi_dec, lo_rec, hi_rec] = wfilters(wname)

elseif wname == 5
    wname = 'db7';
    [lo_dec, hi_dec, lo_rec, hi_rec] = wfilters(wname);

elseif wname == 6
    wname = 'sym4';
    [lo_dec, hi_dec, lo_rec, hi_rec] = wfilters(wname);

```

```

elseif wname == 7
    wname = 'sym8';
    [lo_dec,hi_dec,lo_rec,hi_rec] = wfilters(wname);

elseif wname == 8
    wname = 'Bior3.5';
    [lo_dec,hi_dec,lo_rec,hi_rec] = wfilters(wname);

elseif wname == 9
    wname = 'Bior6.8';
    [lo_dec,hi_dec,lo_rec,hi_rec] = wfilters(wname);

elseif wname == 10
    wname = 'CDF97';

    lo_dec = .....
[
    3.782845550699535e-02,
    -2.384946501937986e-02,
    -1.106244044184226e-01,
    3.774028556126536e-01,
    8.526986790094022e-01,
    3.774028556126537e-01,
    -1.106244044184226e-01,
    -2.384946501937986e-02,
    3.782845550699535e-02
];
    lo_rec = .....
[
    -6.453888262893856e-02,
    -4.068941760955867e-02,
    4.180922732222124e-01,
    7.884856164056651e-01,
    4.180922732222124e-01,
    -4.068941760955867e-02,
    -6.453888262893856e-02
];

```

```

hi_dec = .....
[
-6.453888262893856e-02,
4.068941760955867e-02,
4.180922732222124e-01,
-7.884856164056651e-01,
4.180922732222124e-01,
4.068941760955867e-02,
-6.453888262893856e-02 ];
hi_rec = .....
[
-3.782845550699535e-02,
-2.384946501937986e-02,
1.106244044184226e-01,
3.774028556126536e-01,
-8.526986790094022e-01,
3.774028556126537e-01,
1.106244044184226e-01,
-2.384946501937986e-02,
-3.782845550699535e-02
];
elseif wname == 11
wname = 'coif30';

lo_dec = .....
[
-2.12080840e-05,
3.58589688e-04,
2.178236358e-03,
-4.159358782e-03,
-1.0131117521e-02,
2.3408156788e-02,
2.8168028974e-02,
-9.1920010569e-02,
-5.2043163181e-02,
4.21566206733e-01,
7.74289603730e-01,
4.37991626216e-01,
-6.2035963969e-02,

```

```
-1.05574208714e-01,  
4.1289208754e-02,  
3.2683574270e-02,  
-1.9761778945e-02,  
-9.164231163e-03,  
6.764185449e-03,  
2.433373213e-03,  
-1.662863702e-03,  
-6.38131343e-04,  
3.02259582e-04,  
1.40541150e-04,  
-4.1340432e-05,  
-2.1315027e-05,  
3.734655e-06,  
2.063762e-06,  
1.67443e-07,  
9.5177e-08
```

```
];
```

```
hi_dec = ...
```

```
[
```

```
9.5177e-008,  
-1.6744e-007,  
2.0638e-006,  
-3.7347e-006,  
-2.1315e-005,  
4.134e-005,  
0.00014054,  
-0.00030226,  
-0.00063813,  
0.0016629,  
0.0024334,  
-0.0067642,  
-0.0091642,  
0.019762,  
0.032684,  
-0.041289,  
-0.10557,  
0.062036,  
0.43799,
```

```
-0.77429,  
0.42157,  
0.052043,  
-0.09192,  
-0.028168,  
0.023408,  
0.010131,  
-0.0041594,  
-0.0021782,  
0.00035859,  
2.1208e-005
```

```
];
```

```
hi_rec = ...
```

```
[
```

```
2.1208e-005,  
0.00035859,  
-0.0021782,  
-0.0041594,  
0.010131,  
0.023408,  
-0.028168,  
-0.09192,  
0.052043,  
0.42157,  
-0.77429,  
0.43799,  
0.062036,  
-0.10557,  
-0.041289,  
0.032684,  
0.019762,  
-0.0091642,  
-0.0067642,  
0.0024334,  
0.0016629,  
-0.00063813,  
-0.00030226,  
0.00014054,  
4.134e-005,
```

```
-2.1315e-005,  
-3.7347e-006,  
2.0638e-006,  
-1.6744e-007,  
9.5177e-008  
];
```

```
lo_rec = ...  
[  
9.5177e-008,  
1.6744e-007,  
2.0638e-006,  
3.7347e-006,  
-2.1315e-005,  
-4.134e-005,  
0.00014054,  
0.00030226,  
-0.00063813,  
-0.0016629,  
0.0024334,  
0.0067642,  
-0.0091642,  
-0.019762,  
0.032684,  
0.041289,  
-0.10557,  
-0.062036,  
0.43799,  
0.77429,  
0.42157,  
-0.052043,  
-0.09192,  
0.028168,  
0.023408,  
-0.010131,  
-0.0041594,  
0.0021782,  
0.00035859,  
-2.1208e-005  
];
```



```

elseif wname == 12
    wname = 'haar2';

    lo_dec = [0.7071    0.7071];
    hi_dec = [0.7071   -0.7071];
    lo_rec = [ 0.7071    0.7071];
    hi_rec = [ -0.7071    0.7071];

elseif wname == 13
    wname = 'daub4';

    hi_dec = [ -0.48296 0.83652    -0.22414    -0.12941];
    lo_dec = [ -0.12941 0.22414    0.83652    0.48296 ];
    hi_rec = [ -0.12941 -0.22414    0.83652    -0.48296];
    lo_rec = [ 0.48296 0.83652    0.22414    -0.12941];

elseif wname == 14
    wname = 'daub10';
    lo_dec = ...
[
    0.0033357,
    -0.012581,
    -0.0062415,
    0.077571,
    -0.032245
    -0.24229,
    0.13843,
    0.72431,
    0.60383,
    0.1601
];
    hi_dec = ...
[
    -0.1601,
    0.60383,
    -0.72431,
    0.13843,
    0.24229,
    -0.032245,
    -0.077571,

```

```
-0.0062415,  
0.012581,  
0.0033357
```

```
];
```

```
hi_rec = ...
```

```
[
```

```
0.0033357,  
0.012581,  
-0.0062415,  
-0.077571,  
-0.032245,  
0.24229,  
0.13843,  
-0.72431,  
0.60383,  
-0.1601
```

```
];
```

```
lo_rec = ...
```

```
[
```

```
0.1601,  
0.60383,  
0.72431,  
0.13843,  
-0.24229,  
-0.032245,  
0.077571,  
-0.0062415,  
-0.012581,  
0.0033357
```

```
];
```

```
elseif wname == 15
```

```
wname = 'daub20';
```

```
hi_dec = ...
```

```
[
```

```
-0.02667,  
0.18818,  
-0.5272,  
0.68846,
```

```
-0.28117,  
-0.24985,  
0.19595,  
0.12737,  
-0.093057,  
-0.071394,  
0.029458,  
0.033213,  
-0.0036066,  
-0.010733,  
-0.0013954,  
0.0019924,  
0.00068586,  
-0.00011647,  
-9.3589e-005,  
-1.3264e-005  
];  
lo_dec = ...  
[  
-1.3264e-005,  
9.3589e-005,  
-0.00011647,  
-0.00068586,  
0.0019924,  
0.0013954,  
-0.010733,  
0.0036066,  
0.033213,  
-0.029458,  
-0.071394,  
0.093057,  
0.12737,  
-0.19595,  
-0.24985,  
0.28117,  
0.68846,  
0.5272,  
0.18818,  
0.02667  
];  
hi_rec = ...
```

```
[  
  -1.3264e-005,  
  -9.3589e-005,  
  -0.00011647,  
  0.00068586,  
  0.0019924,  
  -0.0013954,  
  -0.010733,  
  -0.0036066,  
  0.033213,  
  0.029458,  
  -0.071394,  
  -0.093057,  
  0.12737,  
  0.19595,  
  -0.24985,  
  -0.28117,  
  0.68846,  
  -0.5272,  
  0.18818,  
  -0.02667
```

```
];
```

```
lo_rec = ....
```

```
[  
  0.02667,  
  0.18818,  
  0.5272,  
  0.68846,  
  0.28117,  
  -0.24985,  
  -0.19595,  
  0.12737,  
  0.093057,  
  -0.071394,  
  -0.029458,  
  0.033213,  
  0.0036066,  
  -0.010733,  
  0.0013954,  
  0.0019924,  
  -0.00068586,
```

```

-0.00011647,
9.3589e-005,
-1.3264e-005
];

elseif wname == 16
    wname = 'symlet16';
    lo_dec = ....
[
-0.0033824,
-0.00054213,
0.031695,
0.0076075,
-0.14329,
-0.061273,
0.48136,
0.77719,
0.36444,
-0.051946,
-0.027219,
0.049137,
0.0038088,
-0.014952,
-0.00030292,
0.00189
];
    hi_dec= ...
[
-0.00189,
-0.00030292,
0.014952,
0.0038088,
-0.049137,
-0.027219,
0.051946,
0.36444,
-0.77719,
0.48136,
0.061273,
-0.14329,

```

```
-0.0076075,  
0.031695,  
0.00054213,  
-0.0033824  
];  
hi_rec = ...  
[  
-0.0033824,  
0.00054213,  
0.031695,  
-0.0076075,  
-0.14329,  
0.061273,  
0.48136,  
-0.77719,  
0.36444,  
0.051946,  
-0.027219,  
-0.049137,  
0.0038088,  
0.014952,  
-0.00030292,  
-0.00189  
];  
lo_rec = ...  
[  
0.00189,  
-0.00030292,  
-0.014952,  
0.0038088,  
0.049137,  
-0.027219,  
-0.051946,  
0.36444,  
0.77719,  
0.48136,  
-0.061273,  
-0.14329,  
0.0076075,  
0.031695,  
-0.00054213,
```

```

-0.0033824
];

elseif wname == 17
    wname = 'symlet20';
    hi_dec = ...
[
-0.00077016,
9.5633e-005,
0.0086413,
-0.0014654,
-0.045927,
0.01161,
0.15949,
-0.070881,
-0.47169,
0.76951,
-0.38383,
-0.035537,
0.03199,
0.049995,
-0.0057649,
-0.020355,
0.00080436,
0.0045932,
-5.7036e-005,
-0.00045933
];
lo_dec = ...
[
-0.00045933,
5.7036e-005,
0.0045932,
-0.00080436,
-0.020355,
0.0057649,
0.049995,
-0.03199,
-0.035537,
0.38383,
0.76951,

```

```
0.47169,  
-0.070881,  
-0.15949,  
0.01161,  
0.045927,  
-0.0014654,  
-0.0086413,  
9.5633e-005,  
0.00077016  
];  
hi_rec = ...  
[  
-0.00045933,  
-5.7036e-005,  
0.0045932,  
0.00080436,  
-0.020355,  
-0.0057649,  
0.049995,  
0.03199,  
-0.035537,  
-0.38383,  
0.76951,  
-0.47169,  
-0.070881,  
0.15949,  
0.01161,  
-0.045927,  
-0.0014654,  
0.0086413,  
9.5633e-005,  
-0.00077016  
];  
lo_rec = ...  
[  
0.00077016,  
9.5633e-005,  
-0.0086413,  
-0.0014654,  
0.045927,  
0.01161,
```



```

-0.15949,
-0.070881,
0.47169,
0.76951,
0.38383,
-0.035537,
-0.03199,
0.049995,
0.0057649,
-0.020355,
-0.00080436,
0.0045932,
5.7036e-005,
-0.00045933
];

elseif wname == 18
    wname = 'bior97';
    lo_dec = ....
[
    0.026749,
    -0.016864,
    -0.078223,
    0.266864,
    0.602949,
    0.266864,
    -0.078223,
    -0.016864,
    0.026749
];
    lo_rec = ....
[
    -0.045636,
    -0.028772,
    0.295636,
    0.557543,
    0.295636,
    -0.028772,
    -0.045636
];
    hi_dec = ....

```

```

[
    -0.045636,
    0.028772,
    0.295636,
    -0.557543,
    0.295636,
    0.028772,
    -0.045636
];
hi_rec = ...
[
    -0.026749,
    -0.016864,
    0.078223,
    0.266864,
    -0.602949,
    0.266864,
    0.078223,
    -0.016864,
    -0.026749
];

end

% choosing quantizer bit
numbit = menu('Choose quantizer
bit', 'one', 'two', 'three', 'four', 'five', ...
'six', 'seven', 'eight');

if numbit == 1
    quantizerbit = 1;
elseif numbit == 2
    quantizerbit = 2;
elseif numbit == 3
    quantizerbit = 3;
elseif numbit == 4
    quantizerbit = 4;
elseif numbit == 5
    quantizerbit = 5;
elseif numbit == 6
    quantizerbit = 6;

```

```

elseif numbit == 7
    quantizerbit = 7;
elseif numbit == 8
    quantizerbit = 8;
end

lo_pass_decomposition = lo_dec(:)';
hi_pass_decomposition = hi_dec(:)';

%-----Decomposition Level 1-----%

% apply lo_dec and hi_dec on rows of image

row_lo = conv2(double(X),lo_dec(:)');
row_hi = conv2(double(X),hi_dec(:)');

% downsampling along y-axis

down_row_lo = dyaddown(row_lo,'c') ;
down_row_hi = dyaddown(row_hi,'c');

% apply lo_dec and hi_dec on columns of image

col_row_lo_lo = conv2(down_row_lo,lo_dec(:)) ;
col_row_lo_hi = conv2(down_row_lo,hi_dec(:));
col_row_hi_lo = conv2(down_row_hi,lo_dec(:));
col_row_hi_hi = conv2(down_row_hi,hi_dec(:));

% downsampling along x-axis

A1 = dyaddown(col_row_lo_lo,'r');
H1 = dyaddown(col_row_lo_hi,'r');
V1 = dyaddown(col_row_hi_lo,'r');
D1 = dyaddown(col_row_hi_hi,'r');

dec1 = [A1 V1; H1 D1];

%figure; imagesc(dec1);colormap(gray);
title('1st level decomposition')

```

```

%-----Uniform Quantization-----%

level=2^quantizerbit
delta=512/level

A1 = delta*floor(A1/delta);
H1 = delta*floor(H1/delta);
V1 = delta*floor(V1/delta);
D1 = delta*floor(D1/delta);

%-----Decomposition Level 2-----%

% apply lo_dec and hi_dec on rows of image

row_lo2 = conv2(A1,lo_dec(:)');
row_hi2 = conv2(A1,hi_dec(:)');

% downsampling along y-axis

down_row_lo2 = dyaddown(row_lo2,'c') ;
down_row_hi2 = dyaddown(row_hi2,'c');

% apply lo_dec and hi_dec on columns of image

col_row_lo_lo2 = conv2(down_row_lo2,lo_dec(:)) ;
col_row_lo_hi2 = conv2(down_row_lo2,hi_dec(:));
col_row_hi_lo2 = conv2(down_row_hi2,lo_dec(:));
col_row_hi_hi2 = conv2(down_row_hi2,hi_dec(:)) ;

% downsampling along x-axis

A2 = dyaddown(col_row_lo_lo2,'r');
H2 = dyaddown(col_row_lo_hi2,'r');
V2 = dyaddown(col_row_hi_lo2,'r');
D2 = dyaddown(col_row_hi_hi2,'r');

dec2 = [A2 V2; H2 D2];

%figure;
imagesc(dec2);colormap(gray); title('2nd level decomposition');

```

```

%-----Uniform Quantization-----%

level=2^quantizerbit
delta=512/level

A2 = delta*floor(A2/delta);
H2 = delta*floor(H2/delta);
V2 = delta*floor(V2/delta);
D2 = delta*floor(D2/delta);

map=gray(256); imwrite(A2,map,'semuacompressLev2.bmp');

%-----reconstruction level 2-----%

lo_pass_reconstruction = lo_rec(:);
hi_pass_reconstruction = hi_rec(:);

% upsampling along x-axis

up_row_A2 = dyadup(A2,0,'r');
up_row_H2 = dyadup(H2,0,'r');
up_row_V2 = dyadup(V2,0,'r');
up_row_D2 = dyadup(D2,0,'r');

% apply lo_dec and hi_dec on columns of image

row_A2 = conv2(up_row_A2,lo_rec(:));
row_H2 = conv2(up_row_H2,hi_rec(:));
row_lo2 = row_A2 + row_H2;

row_V2 = conv2(up_row_V2,lo_rec(:));
row_D2 = conv2(up_row_D2,hi_rec(:));
row_hi2 = row_V2 + row_D2;

% upsampling along y-axis

up_row_lo2 = dyadup(row_lo2,0,'c');
up_row_hi2 = dyadup(row_hi2,0,'c');

% apply lo_dec and hi_dec on rows of image

```

```

row_lo2 = conv2(up_row_lo2,lo_rec(:)');
row_hi2 = conv2(up_row_hi2,hi_rec(:)');
A2_rec = row_lo2 + row_hi2;
A2_rec = wkeep (A2_rec,[size(H1)]);

Rec_2 = [A2_rec V1; H1 D1 ];
%figure; imagesc(Rec_2);colormap(gray); title('2nd level
reconstruction');
%figure; imagesc(A2_rec);colormap(gray); title('reconstruction level
2')

%-----Reconstruction level 1 -----%

lo_pass_reconstruction = lo_rec(:);
hi_pass_reconstruction = hi_rec(:);

% upsampling along x-axis

up_row_A2_rec = dyadup(A2_rec,0,'r');
up_row_H1 = dyadup(H1,0,'r');
up_row_V1 = dyadup(V1,0,'r');
up_row_D1 = dyadup(D1,0,'r');

% apply lo_dec and hi_dec on columns of image

row_A1 = conv2(up_row_A2_rec,lo_rec(:));
row_H1 = conv2(up_row_H1,hi_rec(:));
row_lo1 = row_A1 + row_H1;

row_V1 = conv2(up_row_V1,lo_rec(:)) ;
row_D1 = conv2(up_row_D1,hi_rec(:)) ;
row_hi1 = row_V1 + row_D1;

% upsampling along y-axis

up_row_lo1 = dyadup(row_lo1,0,'c') ;
up_row_hi1 = dyadup(row_hi1,0,'c');

% apply lo_dec and hi_dec on rows of image

row_lo1 = conv2(up_row_lo1,lo_rec(:)');

```

```

row_hil = conv2(up_row_hil,hi_rec(:)');
A1_rec = row_lo1 + row_hil;
A1_rec = wkeep (A1_rec,[size(X)]);

figure; imagesc(A1_rec);colormap(gray); title('reconstruction level
1')

%-----calculation-----%

map=gray(256); imwrite(A1_rec,map,'semuarecLev2.bmp');

original = double(X);
rec_lev2 = A1_rec;
[m n]=size(original);
mse_lev2 = (sum(sum((original-rec_lev2).*(original-
rec_lev2))))/(m*n)
psnr_lev2 = 10*log10(255*255/mse_lev2)

[counts1,x] = imhist(X);
%figure; bar(x,count1);title('histogram count original picture')

C = imread('semuacompressLev2.bmp');
[counts2,x2] = imhist(C);
%figure; bar(x2,count2);title('histogram count compressed')
F = imread('semuarecLev2.bmp');
[counts3,x3] = imhist(F);
%figure; bar(x3,count3);title('histogram count reconstructed')

probability1=counts1/(m*n);
probability1=nonzeros(probability1);
Entropy_Original = -sum(sum(probability1.*log2(probability1)))
probability2=counts2/(m*n);
probability2=nonzeros(probability2);
Entropy_compressed = -sum(sum(probability2.*log2(probability2)))
probability3=counts3/(m*n);
probability3=nonzeros(probability3);
Entropy_Reconstr = -sum(sum(probability3.*log2(probability3)))
compressed = m*n*Entropy_compressed;
uncompressed = m*n*quantizerbit;
CompressionRatio = uncompressed/compressed;

```

APPENDIX B
MSE LEVEL 1

	haar	coif4	coif5	coif30	daub4	db5	db7	daub10
Lena	608.5788	415.3291	415.5765	415.7742	412.8303	415.3723	416.0099	415.3548
Baboon	357.4346	364.3440	365.0913	364.4467	360.0708	364.7383	364.6274	364.7039
Goldhill	386.4696	394.2196	394.9798	403.1135	394.2638	394.6198	393.4548	394.6001
Boat	387.2880	413.6643	414.2016	417.0723	406.3997	413.0037	412.9353	412.9959
Peppers	384.9653	405.2759	405.1402	406.0188	401.7266	406.6845	403.9207	406.6716
Girl	387.8996	418.2654	418.3394	422.5201	409.3010	419.4740	417.0659	419.4398
House	368.2152	415.9302	416.3916	419.5846	404.4077	413.4774	413.6790	413.4749
F16	378.0084	426.0049	427.0512	430.1945	412.7994	425.4181	425.4181	425.8964
Barbara	381.4774	408.8559	409.2628	411.8712	395.0097	405.7754	406.3764	405.7631
	daub20	sym4	sym8	sym16	sym20	CDF97	bior3.5	bior6.8
Lena	417.9533	412.9798	415.7671	415.7647	417.5718		556.2930	423.9918
Baboon	363.5584	363.0828	364.6884	364.6444	363.2548		486.0851	368.9398
Goldhill	392.7798	391.9932	393.8523	393.8203	394.1263		541.6001	403.0342
Boat	411.0181	410.6569	413.0982	413.1001	409.9260		567.7523	422.5845
Peppers	404.9515	404.2399	405.3151	405.3165	406.1791		488.0898	410.5963
Girl	415.5472	418.4780	418.5020	418.4923	418.0788		557.4385	427.6345
House	412.0304	413.9382	415.0382	415.0458	412.4633		603.2399	428.8920
F16	423.6672	426.0049	426.5416	426.5423	423.7323		598.6834	440.8476
Barbara	406.7749	404.4115	408.6395	408.6543	407.2395		549.5327	415.6809

APPENDIX C
MSE LEVEL 2

	haar	coif4	coif5	coif30	daub4	db5	db7	daub10
Lena	563.1559	577.2059	577.9052	574.6009	573.6643	577.7693	577.8787	577.2316
Baboon	541.1963	515.3079	515.6138	513.7682	511.7137	516.1005	515.8425	515.9915
Goldhill	564.4649	550.7877	551.2532	558.9036	549.9595	548.5709	546.2660	548.3186
Boat	571.2285	576.2076	576.0867	576.0284	569.5649	571.4300	568.8501	570.8539
Peppers	560.0931	567.8543	566.9604	566.6266	561.6838	569.6015	566.7090	569.2991
Girl	568.8997	583.1307	582.6067	582.1817	573.2146	584.6903	579.2760	584.1944
House	555.5368	574.3934	577.0238	574.2237	568.0360	567.6826	557.9613	567.3063
F16	563.1171	592.6305	591.5196	588.3153	579.4243	587.7575	583.2296	587.0877
Barbara	562.4634	568.0053	567.6768	567.9834	552.6620	564.3266	563.0537	564.1344
	daub20	sym4	sym8	sym16	sym20	CDF97	bior3.5	bior6.8
Lena	578.8478	584.0489	579.0461	578.6550	580.2025		658.2878	568.4106
Baboon	513.9462	514.4952	515.2665	515.1262	514.2737		588.0076	503.3479
Goldhill	545.9720	550.9216	549.7887	549.6244	550.0778		643.0636	542.5924
Boat	572.6363	586.7665	574.5199	573.6559	571.7908		667.4295	567.1892
Peppers	566.8278	570.4422	567.8140	567.5802	567.3661		591.7493	556.2242
Girl	578.3707	593.7023	582.5037	581.9032	583.0239		660.2371	573.2740
House	573.9231	593.8123	571.6319	570.5915	571.1718		698.8325	567.9380
F16	588.8559	607.9090	594.8321	587.5570	586.8620		698.1819	586.7354
Barbara	563.5115	565.9519	566.4257	566.2644	566.8165		651.9331	557.7870

APPENDIX D

MRA DEFINITION

The sub space V_j is span of energy L^2 . When scale and translate replace $x \rightarrow 2^j x - k$. When j increase the subspace span increase. To perform reconstruction, it is simply inverse of decomposition. The basis functions $\phi_{j,k}(x)$ of V_1 , $\phi(t-x)$ and wavelet basis $\psi(t-x)$ constitute an orthonormal basis for V_0 and W_0 respectively and $V_1 = V_0 \oplus W_0$, where

$$V_0 = \text{scale function} \Rightarrow \phi(x) = \sum h_k \sqrt{2}(2x - k) \text{ for } k \in Z$$

$$W_0 = \text{mother wavelet} \Rightarrow \psi(x) = \sum g_k \sqrt{2}(2x - k)$$

We can get back the coefficient by dot products with $f(x)$ and wavelets or scaling function [3, 4].

$$g_i = \langle f(x), \psi(x) \rangle = \int f(x) \psi(x) dx$$

$$h_i = \langle f(x), \phi(x) \rangle = \int f(x) \phi(x) dx$$

Table 12 : Orthogonal and Biorthogonal Basis Properties[3,4]

<i>Orthogonal</i>	<i>Biorthogonal</i>
$W_k \perp V_k$ $\langle \psi_{j,k}, \phi_{j,k} \rangle = \int \psi(x)\phi(x)dx = 0$ $\langle \phi_{j,k}, \phi_{j',k'} \rangle = \delta_{k,k'} = \int \phi(x)\phi(x)dx = 1$ $\langle \psi_{j,k}, \psi_{j',k'} \rangle = \delta_{k,k'} = \int \psi(x)\psi(x)dx = 1$ $\phi(x) = \sqrt{2}h_k\phi(2x-k)$ $\psi(x) = \sqrt{2}g_k\phi(2x-k)$	$W_j \perp \tilde{V}_j$ and $\tilde{W} \perp V_j$ $\tilde{W}_j \perp W_j$ for $j \neq j'$ $\langle \tilde{\psi}_{j,k}, \phi_{j,k} \rangle = \int \tilde{\psi}(x)\phi(x) = 0$ $\langle \tilde{\phi}_{j,k}, \psi_{j,k} \rangle = \int \tilde{\phi}(x)\psi(x) = 0$ $\langle \psi_{j,k}, \tilde{\psi}_{j',k'} \rangle = \delta_{j,j'} \cdot \delta_{k,k'} = \int \psi(x)\tilde{\psi}(x)dx = 1$ $\langle \phi_{j,k}, \tilde{\phi}_{j,k'} \rangle = \delta_{k,k'} = \int \phi(x)\tilde{\phi}(x)dx = 1$ $\phi(x) = \sqrt{2} \sum h_k \phi(2x-k)$ $\tilde{\phi}(x) = \sqrt{2} \sum \tilde{h}_k \tilde{\phi}(2x-k)$ $\psi(x) = \sqrt{2} \sum g_k \phi(2x-k)$ $\tilde{\psi}(x) = \sqrt{2} \sum \tilde{g}_k \tilde{\phi}(2x-k)$

APPENDIX E

ENTROPY CODING

Huffman Code

The concept of such a code is quite simple. The most common gray levels are assigned to a code word that requires few bits to transmit, while levels which occur infrequently are encoded with many bits. The Huffman code assumes that the gray values are selected at random from some known probability distribution. The entropy of the Huffman-coded image always is within 1 bit per pixel of the information content defined by Shannon. The Huffman code removes bits from the message by discarding objective redundancy [6].

The Huffman code is best described by example. The pixels in an image with 8 gray levels are usually defined specified by a binary code that requires 3 bits per pixel. For clarity, the gray levels will be indicated by alphabetic characters:

Decimal = Binary \rightarrow Alphabetic

- 0. = $000_2 \rightarrow A$
- 1. = $001_2 \rightarrow B$
- 2. = $010_2 \rightarrow C$
- 3. = $011_2 \rightarrow D$
- 4. = $100_2 \rightarrow E$
- 5. = $101_2 \rightarrow F$
- 6. = $110_2 \rightarrow G$
- 7. = $111_2 \rightarrow H$

Consider a 3-bit 100-pixel image with levels distributed as in the following histogram:

$$H [A, B, C, D, E, F, G, H] = [0, 9, 12, 40, 30, 5, 4, 0]$$

The probability of each gray level therefore is:

$$p[A] = 0, p[B] = .09, p[C] = 12, p[D] = .40, p[E] = 0.30, p[F] = 0.05, p[G] = 0.04, p[H] = 0$$

The sequence continues until all bits have been decoded [6]:

100	00	11	11	1011	11	0	10101	10100	100	0	11	100
-----	----	----	----	------	----	---	-------	-------	-----	---	----	-----

CDDEEBEDFGCDEC

Arithmetic Coding

In arithmetic coding, a message is represented by an interval of real numbers between 0 and 1. As the message becomes longer, the interval needed to represent it becomes smaller, and the number of bits needed to specify that interval grows. Successive symbols of the message reduce the size of the interval in accordance with the symbol probabilities generated by the model. The more likely symbols reduce the range by less than the unlikely symbols and hence add fewer bits to the message. Before anything is transmitted, the range for the message is the entire interval $[0, 1)$, denoting the half-open interval $0 \leq x < 1$. As each symbol is processed, the range is narrowed to that portion of it allocated to the symbol. For example, suppose the alphabet is a, e, i, o, u, ! and a fixed model is used with probabilities shown in Table 6 [9].

Table 14 : Probabilities table for arithmetic coding

Symbol	Probability	Range
a	0.2	$[0,0.2)$
e	0.3	$[0.2,0.5)$
i	0.1	$[0.5,0.6)$
o	0.2	$[0.6,0.8)$
u	0.1	$[0.8,0.9)$
!	0.1	$[0.9,1.0)$

Imagine transmitting the message eaii!. Initially, both encoder and decoder know that the range is $[0, 1)$. After seeing the first symbol, e, the encoder narrows it to $[0.2, 0.5)$. The range the model allocates to this symbol. The second symbol, a, will narrow this new range to the first one-fifth of it, since a has been allocated $[0, 0.2)$. This produces $[0.2, 0.26)$, since the previous range was 0.3 units long and one-fifth of that is 0.06. The next symbol, i, is allocated $[0.5, 0.6)$, range $[0.23, 0.236)$. Proceeding in this way, the encoded message builds up as follows [9]:

Initially	$[0, 0.2)$
after seeing e	$[0.2, 0.5)$
a	$[0.2, 0.26)$
i	$[0.23, 0.236)$
i	$[0.233, 0.2336)$
!	$[0.23354, 0.2336)$

Figure 28 shows another representation of the encoding process. The vertical bars with ticks represent the symbol probabilities stipulated by the model. After the first symbol has been processed, the model is scaled into the range $[0.2, 0.5)$, as shown below.

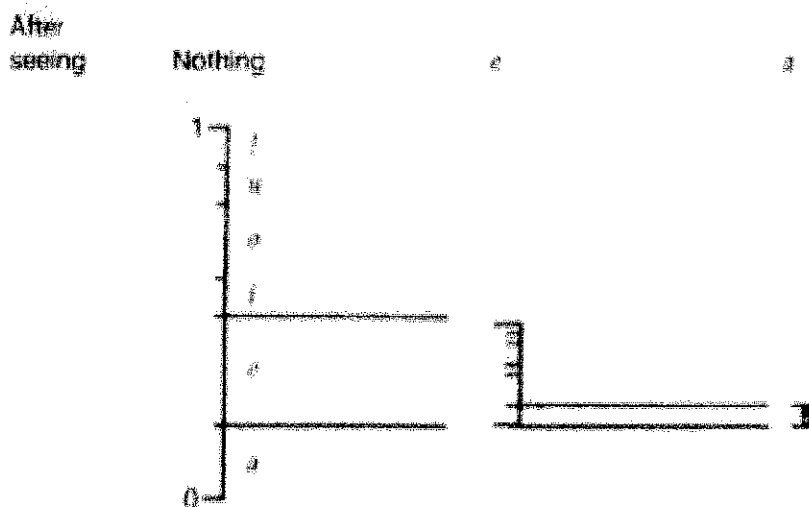


Figure 28 : The second symbol scales it again into the range $[0.2, 0.26)$

The second symbol scales it again into the range [0.2, 0.26). But the picture cannot be continued in this way without a magnifying glass! Consequently, Figure 29 shows the ranges expanded to full height at every stage and marked with a scale that gives the endpoints as numbers.

Suppose all the decoder knows about the message is the final range, [0.23354, 0.2336). It can immediately deduce that the first character was e, since the range lies entirely within the space the model of Table 14 allocates for e. Now it can simulate the operation of the encoder:

Initially [0, 1]

After seeing e [0.2, 0.5]

This makes it clear that the second character is a, since this will produce the range

After seeing a [0.2, 0.26), which entirely encloses the given range [0.23354, 0.2336).

Proceeding like this, the decoder can identify the whole message [9].

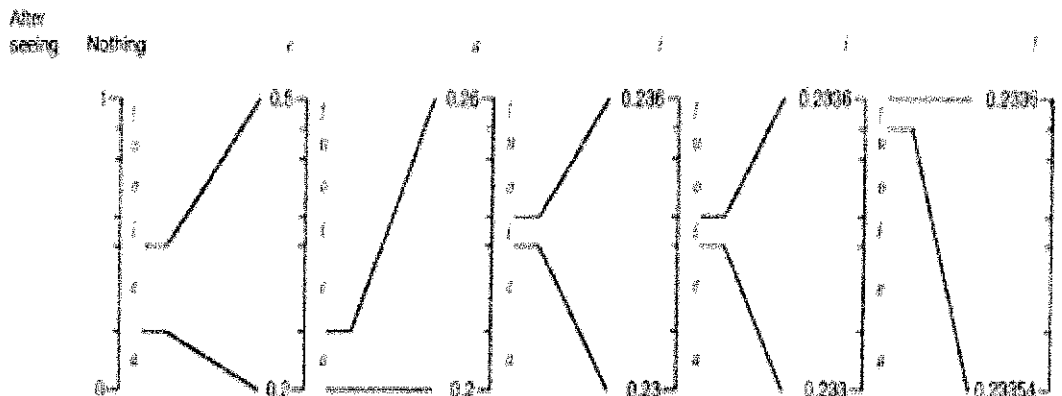


Figure 29 : Representation of interval scale up of Arithmetic coding

For example, we transmit eaii!

i) After seeing e, range = 0.3 from interval [0.2, 0.5)

ii) After seeing a

Range = previousHigh-previousLow

Newlowinterval = prevLow+range*(currentLow)=0.2+0.3(0) =0.2

Newhighinterval = prevLow+range*(currentHigh)=0.2+0.3(0.2)=0.26

iii) After seeing i

$$\text{Range} = \text{previousHigh} - \text{previousLow} = 0.26 - 0.2 = 0.06$$

$$\text{Newlowinterval} = \text{prevLow} + \text{range} * (\text{currentLow}) = 0.2 + 0.06(0.5) = 0.230$$

$$\text{Newhighinterval} = \text{prevLow} + \text{range} * (\text{currentHigh}) = 0.2 + 0.06(0.6) = 0.236$$

iv) After seeing i

$$\text{Range} = \text{previousHigh} - \text{previousLow} = 0.236 - 0.230 = 0.0006$$

$$\text{Newlowinterval} = \text{prevLow} + \text{range} * (\text{currentLow}) = 0.23 + 0.006(0.5) = 0.233$$

$$\text{Newhighinterval} = \text{prevLow} + \text{range} * (\text{currentHigh}) = 0.23 + 0.006(0.6) = 0.2336$$

v) After seeing!

$$\text{Range} = \text{previousHigh} - \text{previousLow} = 0.2336 - 0.233 = 0.00006$$

$$\text{Newlowinterval} = \text{prevLow} + \text{range} * (\text{currentLow}) = 0.233 + 0.00006(0.9) = 0.23336$$

$$\text{Newhighinterval} = \text{prevLow} + \text{range} * (\text{currentHigh}) = 0.233 + 0.00006(1.0) = 0.23354$$

APPENDIX F
DATA AND EXCEL GRAPH

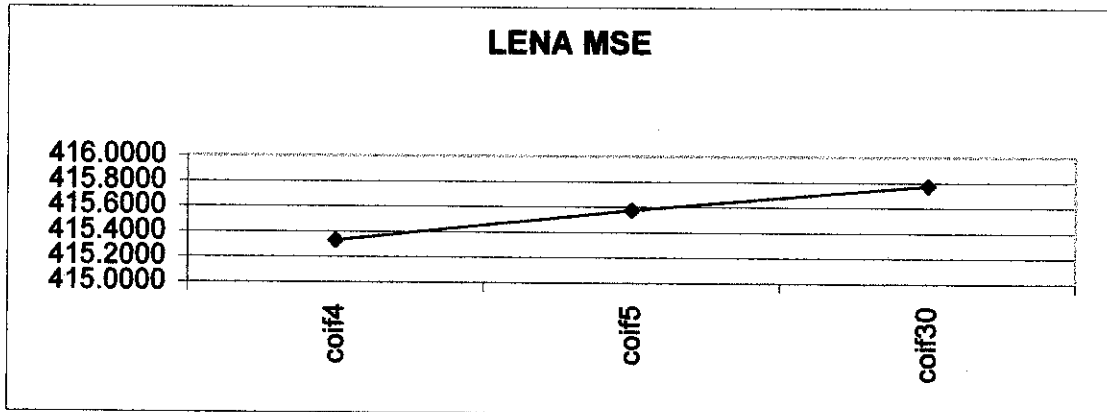


Figure 30 : Lena Coiflet MSE level 1

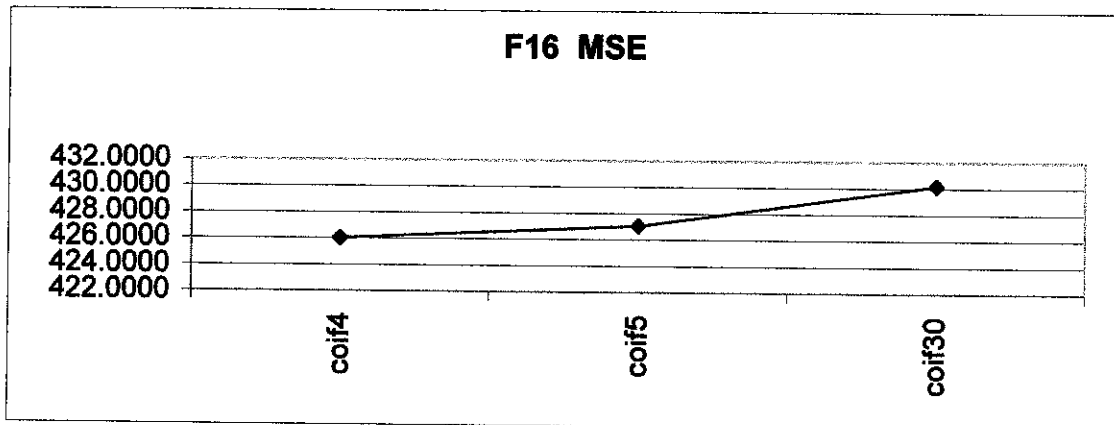


Figure 31 : F16 Coiflet MSE level 1

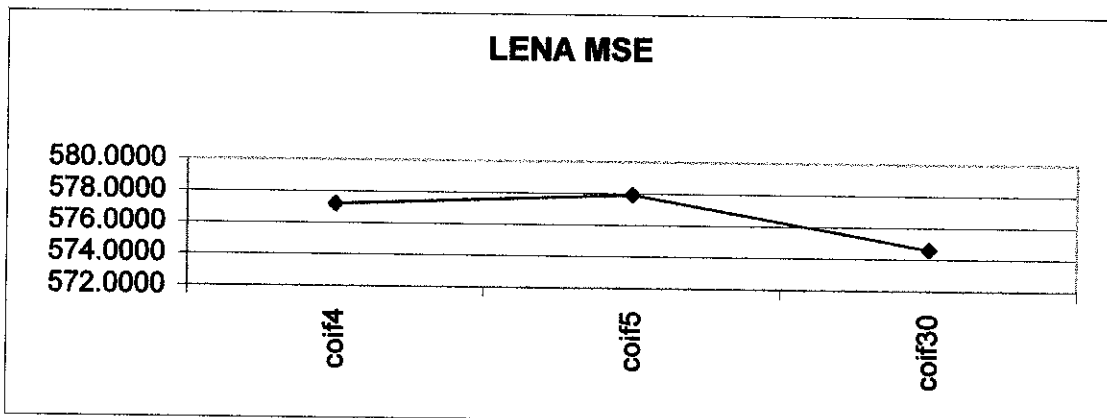


Figure 32 : Lena Coiflet MSE level 2

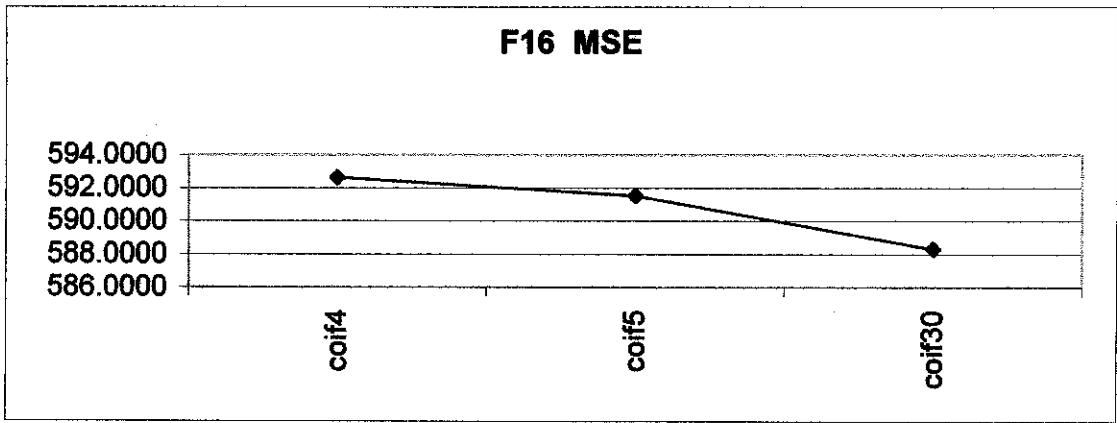


Figure 33 : F16 Coiflet MSE level 2

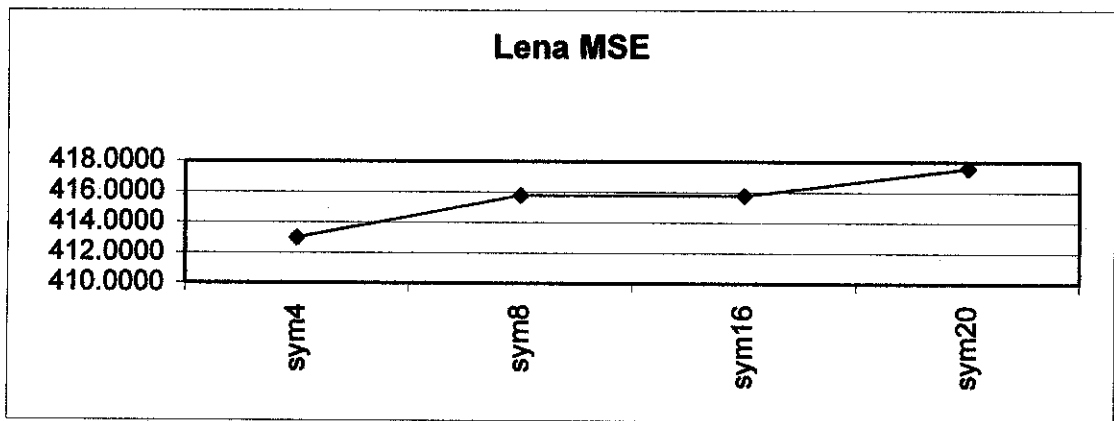


Figure 34 : Lena Symlet MSE level 1

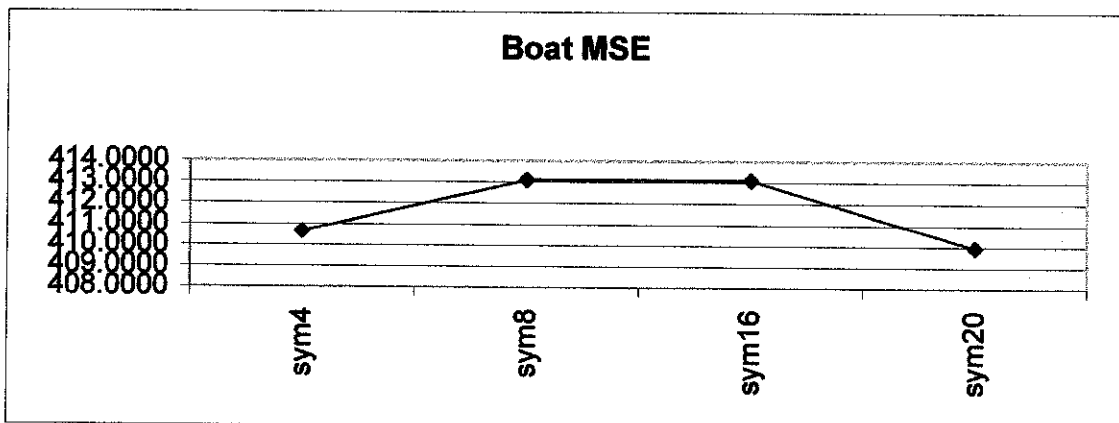


Figure 35 : Boat Symlet MSE Level 1

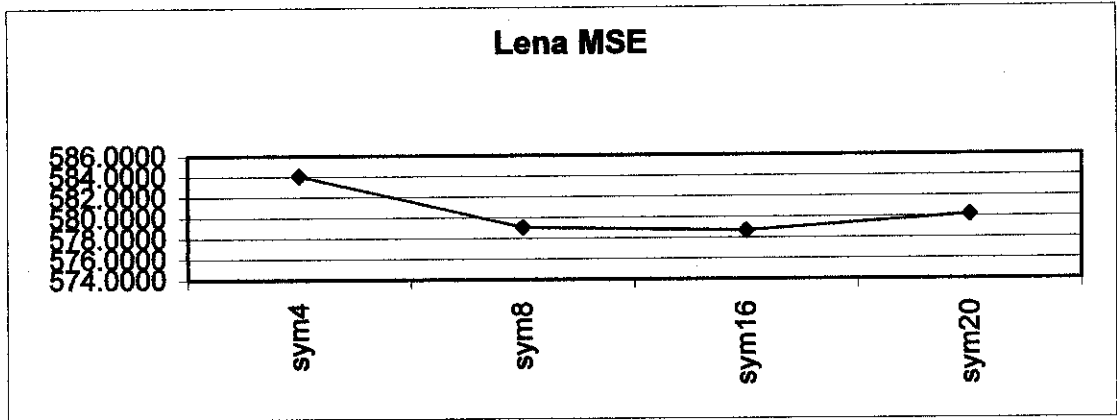


Figure 36 : Lena Symlet MSE level 2

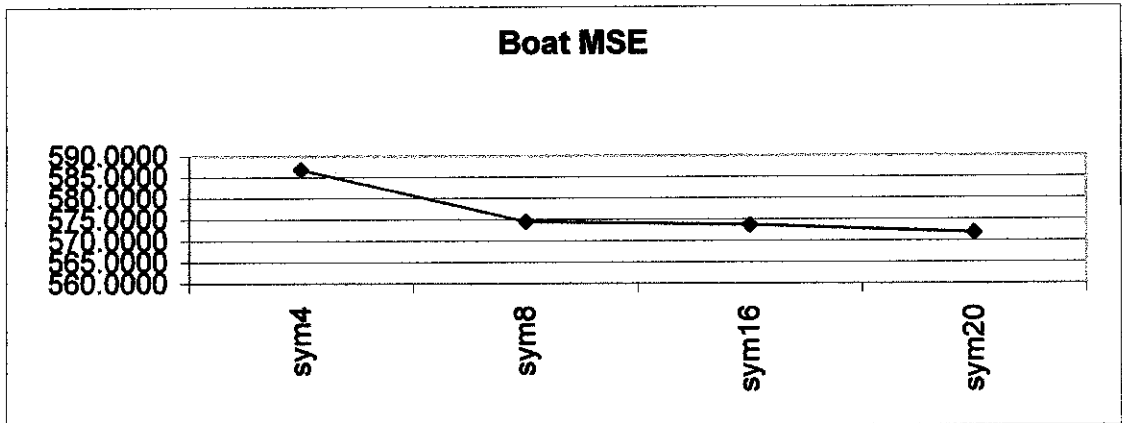


Figure 37 : Boat Symlet MSE Level 2

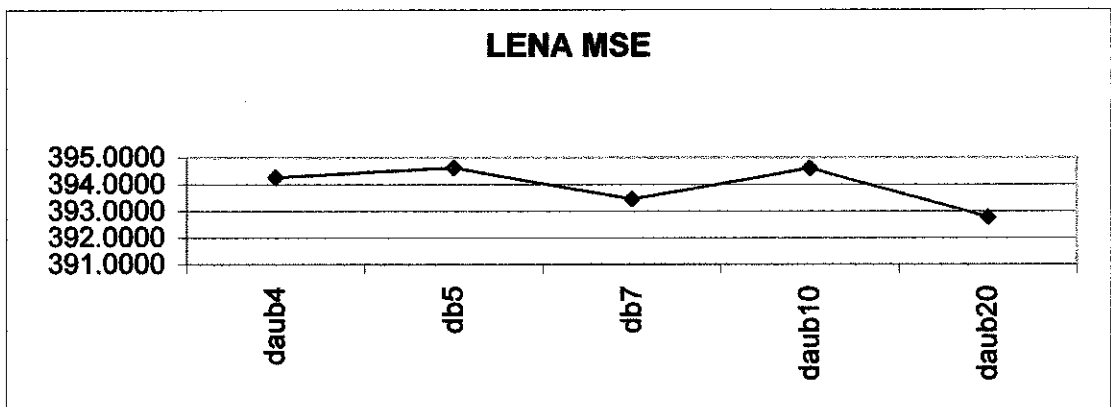


Figure 38 : Lena Daubachies MSE Level 1

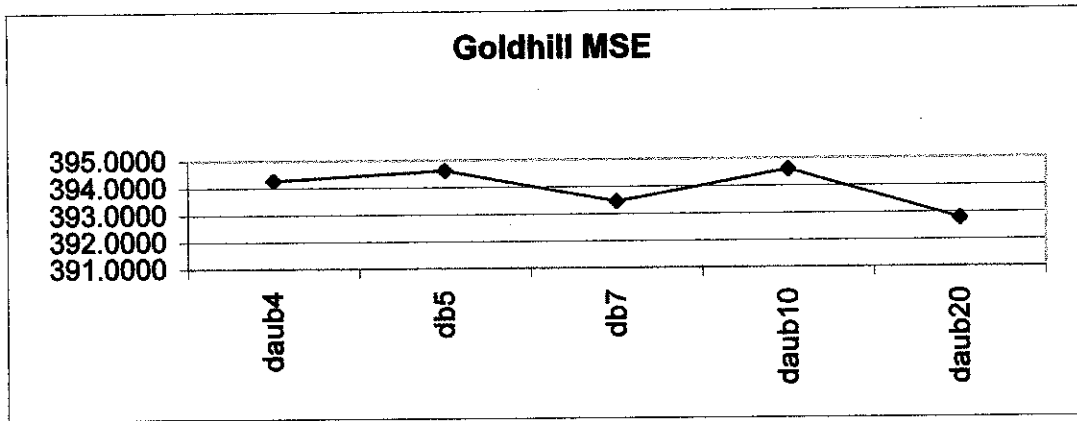


Figure 39 : Goldhill Daubachies MSE Level 1

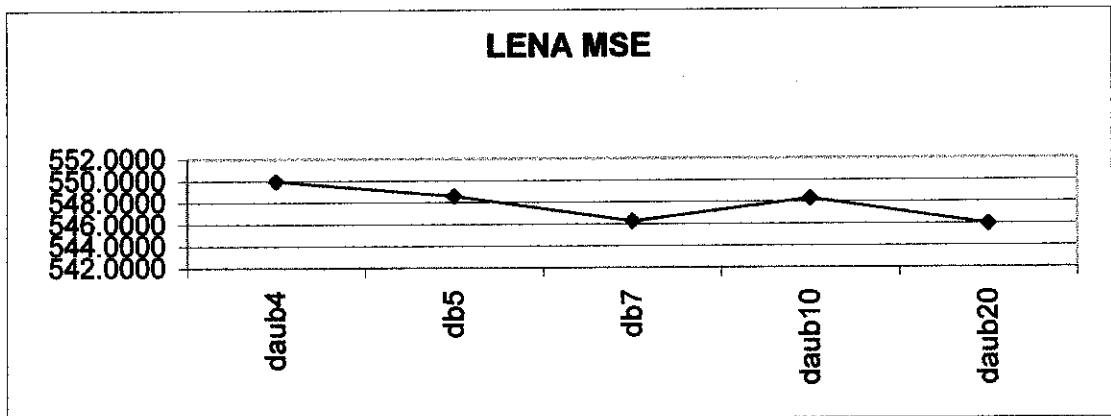


Figure 40 : Lena Daubachies MSE Level 2

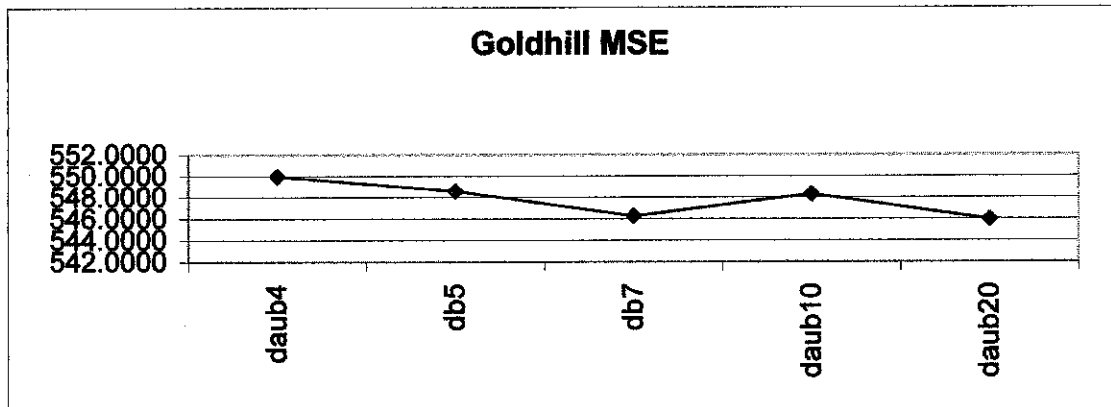


Figure 41 : Goldhill Daubachies MSE Level 2

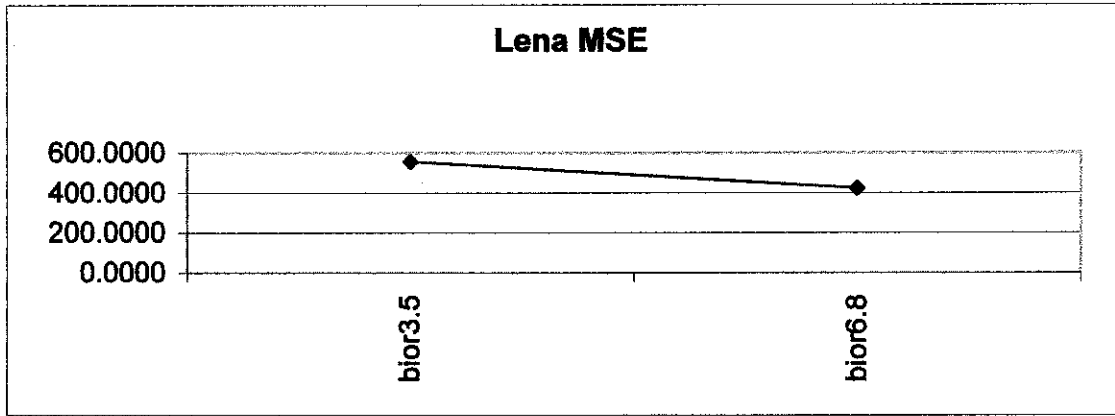


Figure 42 : Bior Spline MSE level 1

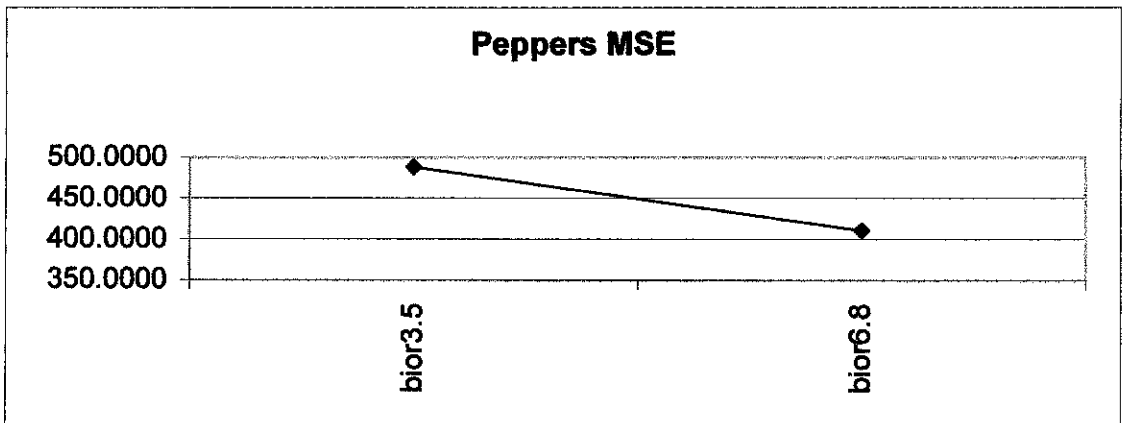


Figure 43 : Peppers Bior Spline MSE level 1

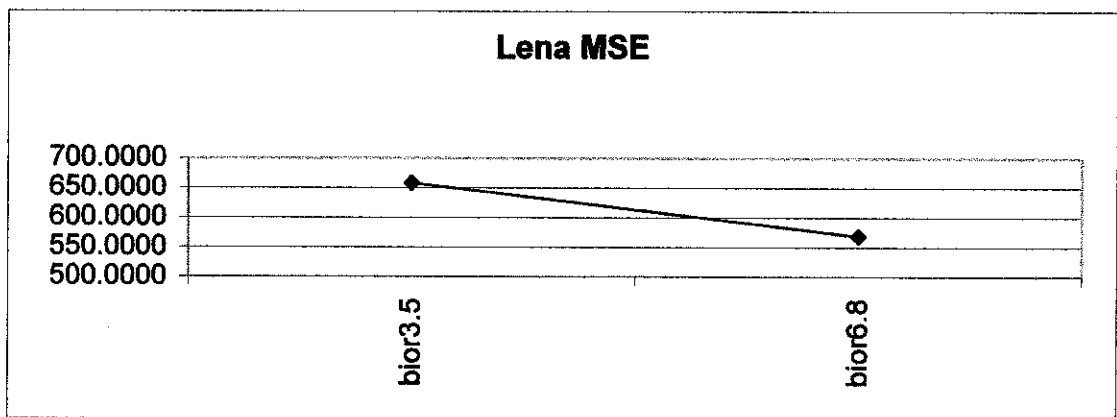


Figure 44 : Bior Spline MSE level 2

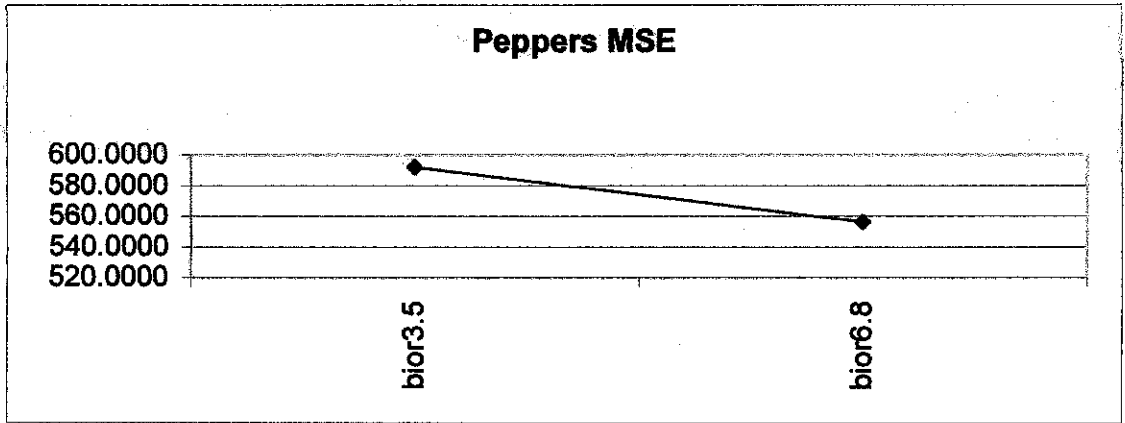


Figure 45 : Peppers Bior Spline MSE level 2

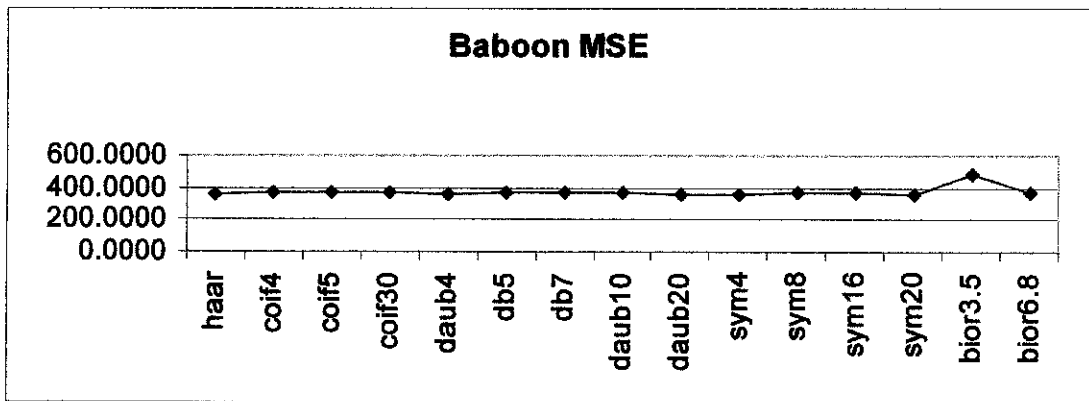


Figure 46 : Baboon MSE level 1

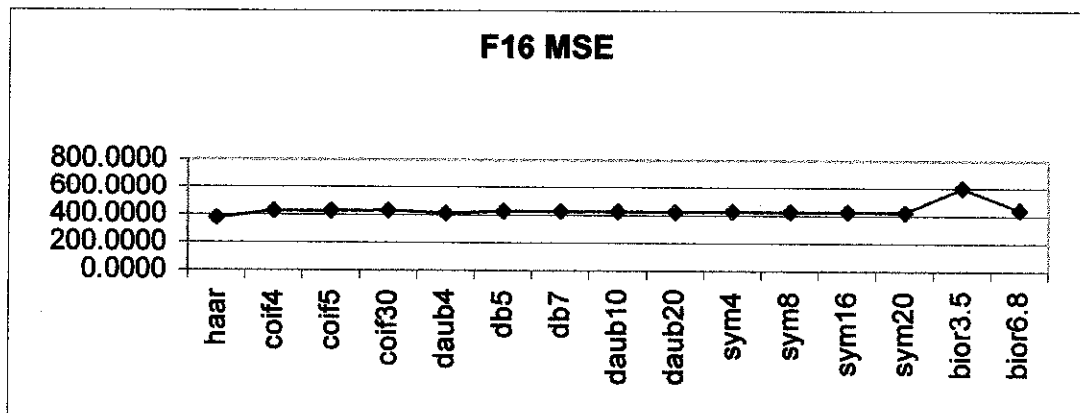


Figure 47 : F16 MSE level 1

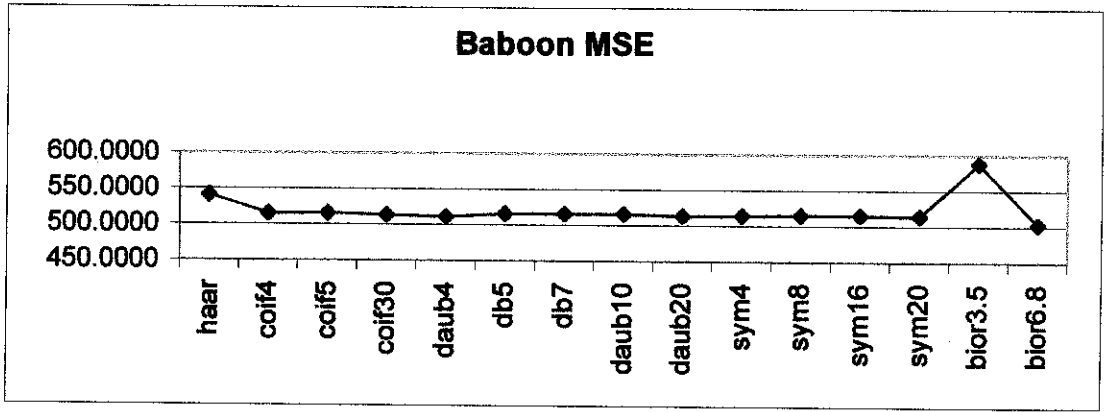


Figure 48 : Baboon MSE Level 2

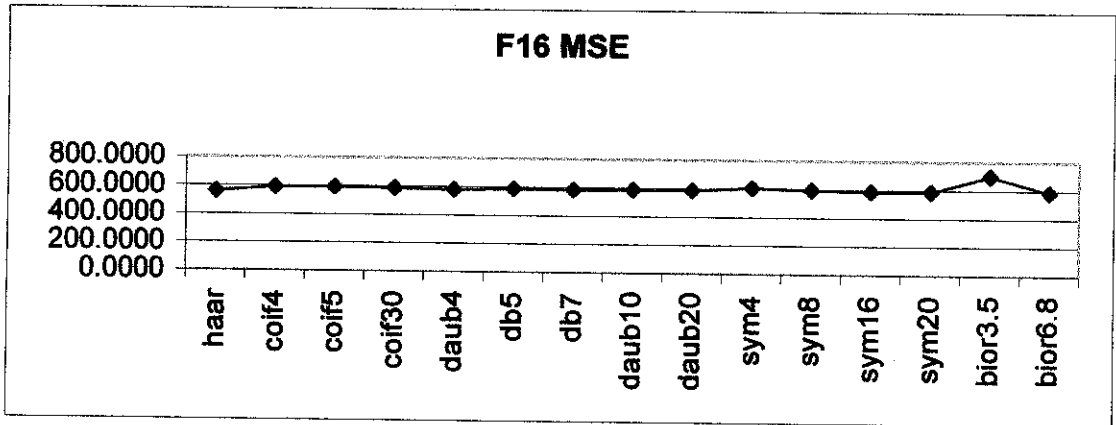


Figure 49 : F16 MSE Level 2

APPENDIX G
TEST IMAGES USED



Figure 50 : House Image



Figure 51 : Lena Image

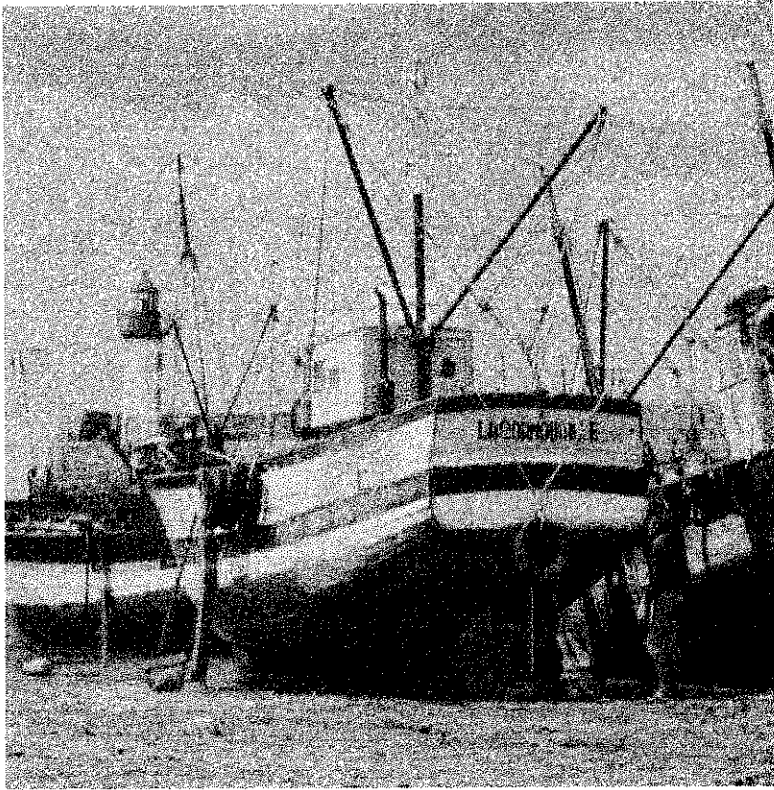


Figure 52 : Boat Image



Figure 53 : F16 Image



Figure 54 : Girl Image

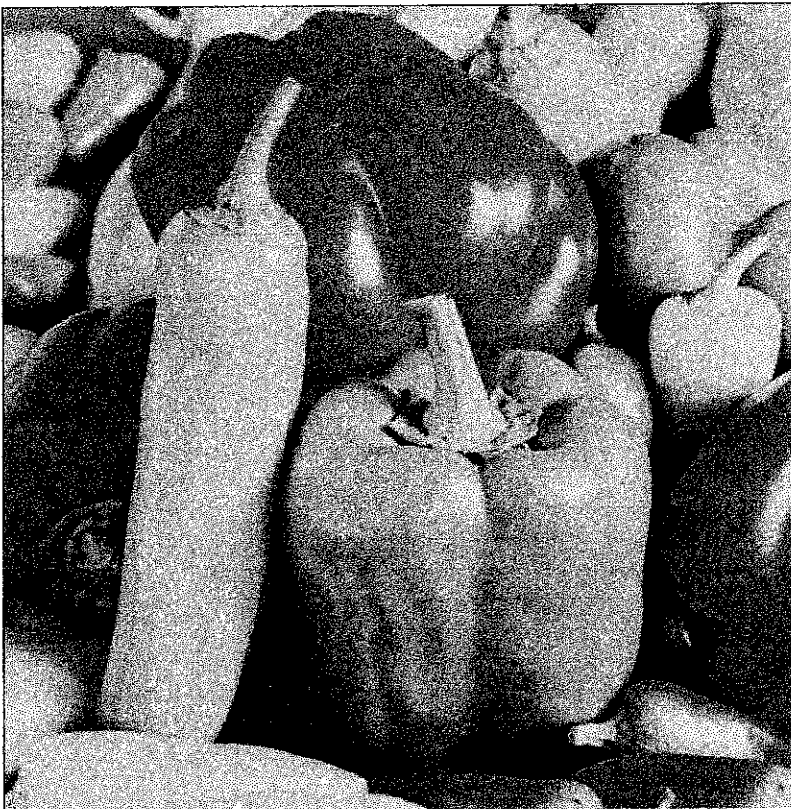


Figure 55 : Peppers Image

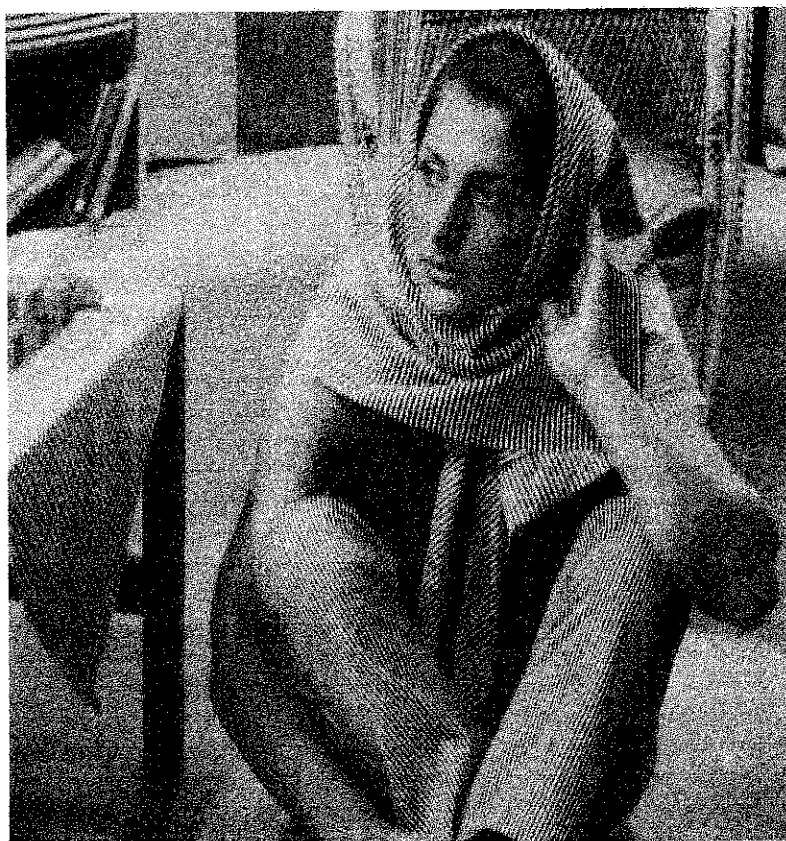


Figure 56 : Barbara Image



Figure 57 : Goldhill Image

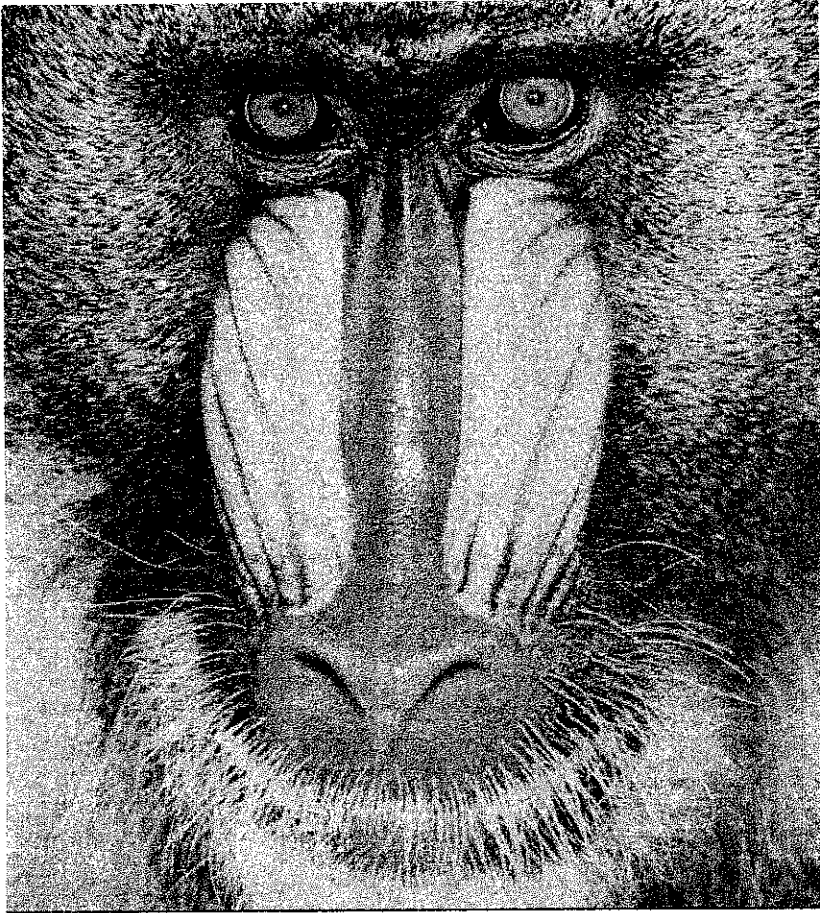


Figure 58 : Baboon Image